

You are here: Welcome to Rules Palette > Disclaimer

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Rules Palette Online Help

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

License Restrictions Warranty/Consequential Damages Disclaimer

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

Warranty Disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S.

Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and

license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Hazardous Applications Notice

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Third Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.



You are here: Introduction

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Welcome to the Rules Palette

The Rules Palette is Oracle's insurance configuration solution for both individual and group insurance products. It is a standalone application that can be used in conjunction with select Oracle Insurance applications, and provides straightforward functionality that assists with configuration tasks.

For the most recent and up-to-date version of our help documentation, refer to the Rules Palette help published on the Oracle Technology Network.

Navigate to <http://www-content.oracle.com/technetwork/documentation/insurance-097481.html>, click on the "Policy Administration" link at the top of the page, then click "View Library" for the Oracle Insurance Policy Administration library E59346-01. The Rules Palette help link is located at the bottom of the library in the Rules Palette section.

Help System Topics

This help system is divided into books that contain related topics. The left navigation directory tree structure provides access to the books and pages. Expand each book to view the associated help pages. Use the Index to look up specific terms and find related information. [View more...](#)

What's New for This Release

There are two ways to access information on the new functionality in this release. The [What's New](#) page provides an explanation of all the new functionality, while the **Prototype** book in the Table of Contents also provides direct links to new prototype samples as well as all past prototype samples.

Quick Start Guide

This guide is designed to jump start your configuration tasks. It provides a type of shortcut menu to the steps involved in each major area of OIPA configuration. [View Quick Start Guide...](#)

Additional Documentation

The Oracle Technology Network (OTN) contains documentation libraries for every release. These libraries hold links to all of the following documents, specific to each release:

- Rules Palette help system
- OIPA help system
- XML Configuration Guide
- Activity Processing document
- Extensibility document
- Cycle document
- OIPA Properties document
- Web Services document
- Release Notes for Rules Palette and OIPA
- Installation documents

Access the 10.1.1.0 Documentation Library at <http://www.oracle.com/technetwork/documentation/insurance-097481.html>. Scroll to the bottom of the page and select the E59346-01 OIPA Documentation Library for the 10.1.2.0 release.

Oracle Customer Support

If you have any questions about the use of our products, please visit the My Oracle Support website: <https://support.oracle.com>, or call (800) 223-1711.

What's New

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

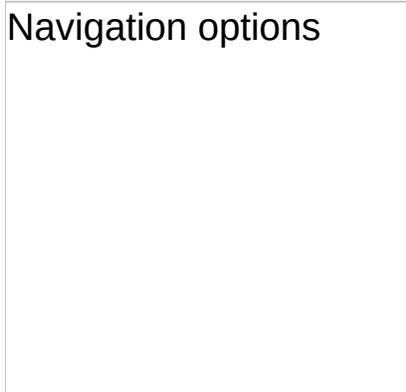
You are here: Welcome to Rules Palette > How To Use This Help

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

How To Use This Help

Use the directory tree along the left side of the screen to expand nodes and find links to the various help pages. Use the index to look up specific topics and access related information.

Navigation options



Navigation Tree Structure in Help and Index Link in Help Menu

Prototype Company

A Prototype Company is available in the Main Explorer of the Rules Palette. This company contains configuration samples, which demonstrate enhanced functionality specific to this release. There are many pieces of functionality that are demonstrated in the Prototype Company. Refer to the [Prototype](#) section of this help for an explanation of all new functionality.



Prototype Company Node in Main Explorer

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Welcome to Rules Palette > Quick Start Guide

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Quick Start Guide

The following links are designed to accelerate your learning curve in the Rules Palette. Specific configuration tasks are listed below, along with the main steps needed to complete the configuration task. Additional information on the configuration topic can be found in the help Table of Contents.

Click a link below to access the Quick Start pages.

Set Up a New Company

- [Create a new company](#)
- [Add company security](#)
- [Associate OIPA user with new company](#)
- [Create plans](#)
- [Add plan security](#)

More details on this topic can be found in the **Main Explorer | Companies** node of the TOC.

Create and Configure Transactions

- [Create a new transaction](#)
- [Configure a transaction](#)
- [Add security to a transaction](#)
- [Add translation to a transaction](#)
- [Add transaction to EligibleTransactionByPolicyStatus rule.](#)

More details on this topic can be found in the **Configuration | Transactions** node of the TOC.

Configure Valuation

- [Determine the type of valuation for the plan](#)
 - If PointInTime is used, configure the PointInTimeValuation rule.
 - If PointInTime is used, configure the WriteValuationElements rule.
- [Create funds](#)
 - [Configure InterestRateCalculation rule for fixed funds](#)
 - [Upload interest rates that will be used by InterestRateCalculation rule](#)
 - [Configure PrecisionValues rule for variable funds](#)
 - [Verify that the AsNetAssetValue table has unit values for funds to use](#)
- Configure valuation business rules
 - [Configure the ValuationDetails rule](#)
 - [Configure the PolicyValues rule](#)
 - [Configure the GainLossCalculation rule](#)
- [Configure the transaction that will run valuation](#)
- [Configure scheduled valuation for a group of policies.](#)

More details on this topic can be found in the **Configuration | Valuation** node of the TOC.

You are here: [What's New in 10.1.2.0](#) > What's New

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

What's New in 10.1.2

The release of OIPA 10.1.2.0 contains the below enhancements in the Rules Palette and OIPA application:

[Ability to Link between Policy and Plan Segment Entities](#)

This feature provides the ability to link the Policy segment entity with Plan segment entity. In group insurance set up, this enables the policy segment to capture the plan segment data.

[Ability to Transfer Indexed Funds from one Bucket to another Bucket](#)

This feature provides an ability to specify a “From” and “To” Bucket for transferring between Indexed funds.

[Market Value Adjustment \(MVA\) Funds Support](#)

This enhancement implements support for MVA processing. MVA funds are defined by a fund type and interest calculation. This calculation is invoked during any valuation that includes MVA funds.

[Batch Processing](#)

This feature invokes a defined business process for a group of specified policies from one screen (Batch Screen) and avoids the need to enter activities on each individual policy. This functionality can be used to invoke activities on other entities in addition to policies.

[Premium Tracking](#)

This feature provides the ability to track, apply, and remove premiums/payments based on age of premium. In addition to tracking premium/payment deposits, bonus or interest payments can be tracked. Tracking the money transactions in the life cycle of a policy can be used to calculate remaining premium, charges, age of deposit, etc.

Inbound Document Attachment

The ability to attach and retrieve a file through the OIPA UI has been added to the Comments functionality. Files are now extensions of comments and at any place a comment can be inserted, so may a file without any additional configuration. All file types are supported and the system allows one file upload per comment but any number of comments may be added to an entity. Entities that support user comments are: Policy, Activity, Client, Requirement, Suspense, and Segment.

Execute Valuation in Functions

This enhancement expands the capability of executing valuation within a transaction's resolved Math to the Math section of a User Defined Function.

[Client Search Screen Security by Client Type](#)

This feature allows limiting the type of Clients that a user/security group can search/view.

GetFundPosition Enhancements

The MathStatement activity function, GetFundPositions, has been modified that allows the exclusion of funds by Fund GUIDs or an array of Fund GUIDs. Previously, fund could only be excluded by Fund Type.

Consistent Policy Summary Sections for all Policy Screens

A consistent Policy Summary presentation has been introduced for all the Policy related screens, except the Policy Screen. The Policy Summary section can be configured or a default display will be used. The Policy Screen itself retains the current fixed field and dynamic field presentations without the Summary. Also, the Policy Summary section has been added to the Policy Overview Screen to make it consistent with all other Policy related screens.

Collections in Assignment Types

The ApplyByFund and RemoveByFund assignment types currently

require explicitly configured fund GUIDs and amounts. This feature will allow these assignments to accept a dynamic collection of fund GUIDs and amounts thereby greatly enhancing the functionality of the assignment types.

Multifield in ScreenMath Action & Event

All screens that support MultiFields will allow an Action/Event configured in a Screen's Business Rules ScreenMath to trigger the Action/Event configured in a MultiFields rule.

Transaction Times Business Rule Enhancement

Configuration of this rule will allow a <Prohibit> element that works the opposite of <Allow>. This supports conditions that are easier to restrict rather than permit. Additionally, the <Allow> and <Prohibit> elements are repeatable to support a 'best match' scenario to find the first configured set of criteria that are all true.

Inquiry Screen – Total

A new attribute 'Total' is added to the output column in the InquiryScreen. This displays the sum of the results of specific numeric data columns at the bottom of the grid.

Multifield Enhancements to Capture and Update data via Transactions

This enhancement allows updates to Multifield data into an entity like Policy or Segment or Role, etc. through a transaction and allows copying data back to Multifields from transactions.

Suspense Accounting based on a Dynamic Suspense Field

This feature allows accounting records to be created using the amount of a dynamic field instead of the Suspense 'Amount' value.

Usability Enhancements

Role Entry

This enhancement improves the user experience and

eliminates confusion while entering 'Role' information, using the 'Find Client' functionality or performing an action using the right click menu.

Client Address Entry

Improvements have been made to correct formatting and eliminate confusion when entering/viewing Client Address information.

History

Updates to a Client's Default Address will be reflected in History. The deletion of a Client's address will be reflected the Address History summary table as a 'Delete' action versus an 'Update' action. Additional information has been added to the History summary table for easier identification of which entity/role has been modified.

Labeling

Warning labels throughout the system (for messages that do not require any action) have been renamed 'Alert' to be in line with standard User Interface naming conventions.

You are here: [Getting Started](#) > [Environments](#) > [Environments Overview](#)

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Environments

An environment connection is needed in order for the Rules Palette to communicate with the database that stores the application configuration.

Creating an environment connection in the Rules Palette involves two steps. First, establish a name for the environment. Next, confirm the database credentials. The actual database properties are pre-populated by the Web Application Utility, which is a tool used by the Build Manager. Confirm the database information by entering the user name and password.

There are two database sections: an application database and an Internal Versioning System (IVS) database. Connecting to an IVS database allows versions of the configuration to be stored so that previous versions can be reverted to if necessary.

The Build Manager will send the information needed to [set-up an environment connection](#).

Changing Environment Properties

Currently Environment properties are stored locally in the Rules Palette's "asenv.properties" file. If a user attempts to change the environment properties manually in the "asenv.properties" file (outside of the Web Application Utility), then a warning message will appear when the user tries to log into the environment. This occurs because the Rules Palette checks the properties settings saved in the Web Application Utility each time a user logs on. If the properties in the asenv.properties file do not match what is stored in the Web Application Utility, then the following error message will appear:

Environment Properties do not match. Logon will continue and properties will be reset.

All Rules Palette properties must be updated through the Web Application Utility.

Environment Right-Click Options

When a user is logged into an environment, there are three right-click options available.

1. **Logout:** selecting this right-click option allows the user to log out of the environment. This action does not close the Rules Palette application.
2. **Refresh:** selecting this right-click option will close all editors and refresh all folders with the most recent updates.
3. **Properties:** selecting this right-click option allows the user to view all environment properties. These properties are set by the build manager during the initial installation. These values are read only and cannot be changed.

When a user is logged out of an environment, there are three right-click options available.

1. **Log on:** selecting this option opens the Log on screen, where a user can enter a Client ID and password to access the Rules Palette.
2. **Delete Environment:** selecting this right-click option allows the user to [delete the environment](#). Environments can only be deleted when the user is logged out.
3. **Properties:** selecting this right-click option allows the user to view all environment properties. These properties are set by the build manager during the initial installation. These values are read only and cannot be changed.

You are here: Getting Started > Environments > Create an Environment

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create an Environment

An environment connection can be established in the Rules Palette once the connection information is received from the Build Manager. The Build Manager will send the following information:

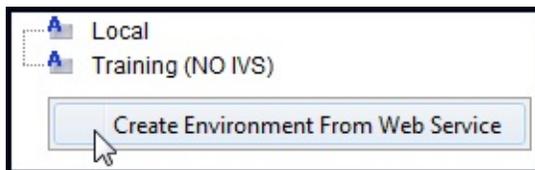
- configuration URL
- Rules Palette log-in name and password for step one of the wizard
- application database name and password for step two of the wizard
- IVS database name and password for step two of the wizard



To download the Rules Palette application, use the **User Installation Instructions** provided with the release.

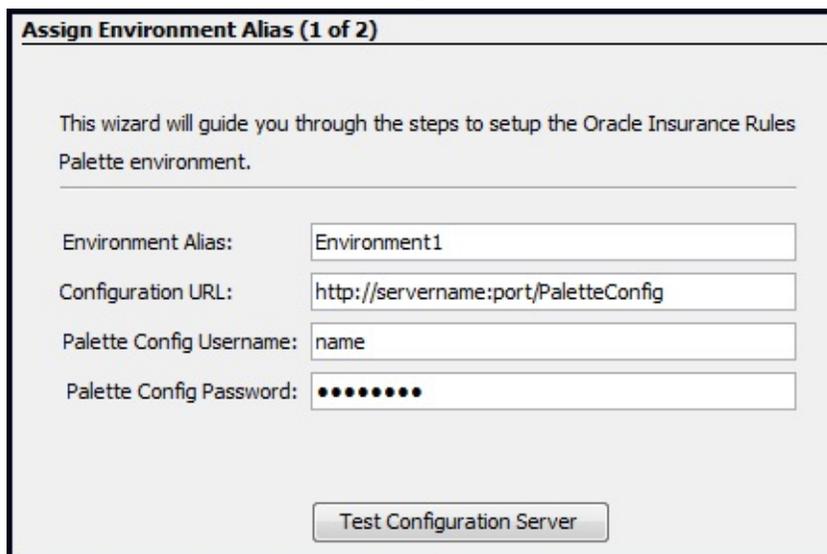
Steps for Rules Palette Users to Create a New Environment

1. Log in to the Rules Palette and select either the **Global Rules Explorer** tab or the **Main Explorer** tab.
2. Right-click anywhere in the tab and select **Create Environment from Web Service**.



Right-click Option to Create Environment

3. Enter an **Environment Alias** that distinguishes one environment from another. Only use alphabet characters in the alias. No numbers or other characters are supported at this time.
4. Enter the **Configuration URL** provided by the Build Manager.
5. Enter the Palette Config username provided by the Build Manager.
6. Enter the Rules Config password provided by the Build Manager.
7. Click **Test Configuration Server** to test the connection to the Web Service that will auto-populate the database properties. If the connection is successful, click **Next**.

A screenshot of a wizard window titled 'Assign Environment Alias (1 of 2)'. The window contains a text box with the instruction: 'This wizard will guide you through the steps to setup the Oracle Insurance Rules Palette environment.' Below this are four input fields: 'Environment Alias' with the value 'Environment1', 'Configuration URL' with the value 'http://servername:port/PaletteConfig', 'Palette Config Username' with the value 'name', and 'Palette Config Password' with a masked password of seven dots. At the bottom of the form is a button labeled 'Test Configuration Server'.

Step One of Environment Creation Wizard

8. When using SQL Server or DB2 databases, identify the location of the driver files. Browse to the location of the jdbc driver files. (SQL Server uses the jtds.jar and DB2 uses two jar files beginning with db2_***.jar. The file location can be obtained from the Build Manager.) This step is only performed the first time an environment is set-up. If additional environments are created, then this Browse field will not display.



Browse Button on Step Two of Environment Creation Wizard

9. Enter the user ID and password for the application database. The database host should be grayed out and listed in the field directly above the user ID and password.
10. Click **Test Connection** to test the database connection. If the connection is not successful, contact the Build Manager.
11. If an IVS database is used, enter the user ID and password for the IVS database. The IVS database host should be grayed out and listed in the field directly above the user ID and password.
12. Click **Test Connection** to test the IVS database connection. Once both connections are successful, click **Finish**.

A node for the environment that was just created will display on the Main Explorer and Global Rules Explorer tabs. If the automatic login box was not clicked, then log in now. Right-click on the node and select [Log on](#) to access the Rules Palette application. Enter the Rules Palette login name and password used in step one of the environment creation wizard.



The Admin Explorer tab cannot be used to log in or out of an environment.



Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Getting Started > Environments > Build Manager Environment Creation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Build Manager Environment Creation

The Build Manager will use the Web Application Utility to configure the Rules Palette environment properties and the remote debugging Web Service. When Rules Palette users create a new environment, they will need the server name, port number, and three sets of user IDs and passwords. The IDs and passwords needed are:

- Palette Config user ID & password: contact the build manager.
- database ID & password: contact the person who set-up the database.
- IVS database ID & password: contact the person who set-up the IVS database.

Web Application Utility

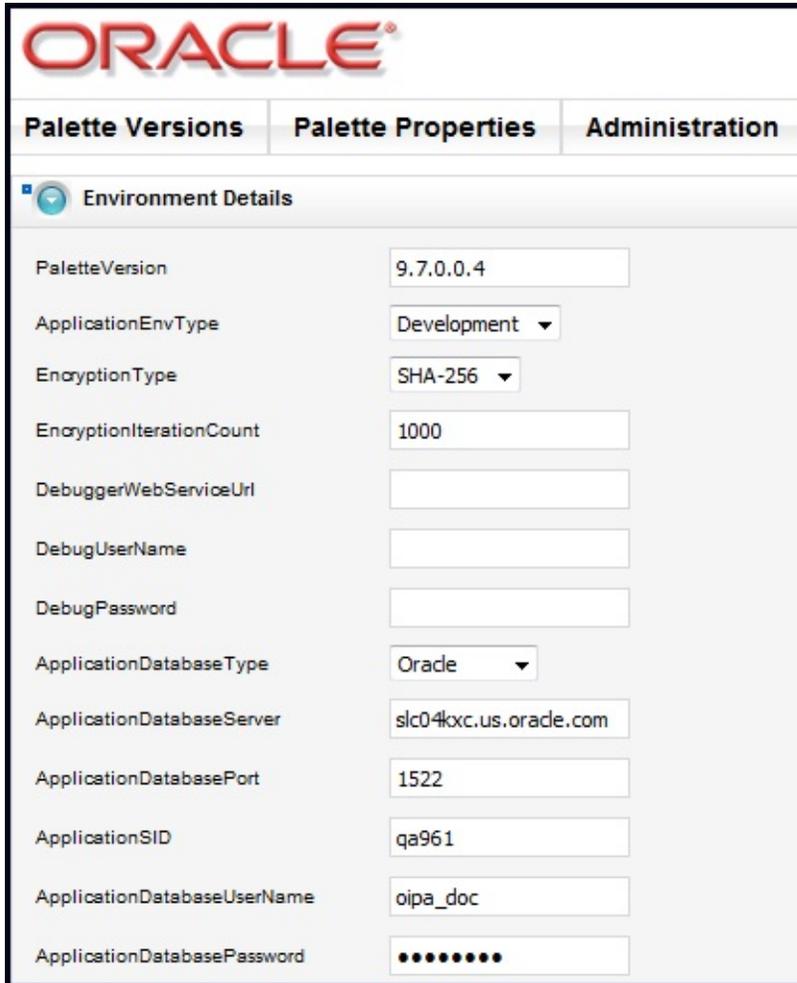
Log in to the Web Application Utility using the following URL:
[http://**host:port**/PaletteConfig/](http://host:port/PaletteConfig/). The host (server name) and port should be the same as the host and port of the application (OIPA or OINBU) being used. The default user name and password are provided in the Web Application Utility installation guide.

Steps for Build Manager to Set-up Environment Properties

1. Log in to the Web Application Utility using the URL:
`http://host:port/PaletteConfig/`.
2. Click **Environment Options** and then **Edit**.
3. Enter the information for the environment:
 - a. **PaletteVersion**: Enter the version of the Rules Palette that will be used.
 - b. **ApplicationEnvType**: Either **Development** or **Production** for the type of environment.
 - c. **EncryptionType**: Select the type of encryption to be used for the environment's password.
 - d. **EncryptionIterationCount**: Select the iteration count to be used with the selected encryption type.
 - e. **DebuggerWebserviceUrl**: URL for the Web Service used to connect for remote debugging. The host and port information must match the host and port information for the OIPA or OINBU application you are using. **Ex**: `http://<host>:<port>/PASJava/service/DebuggerService?wsdl`
 - f. **DebugUserName**: Enter the debug user name. This is a created in the Application Security section of Rules Palette. Add a user and grant debugger security privileges.
 - g. **DebugPassword**: Enter the debug password.
 - h. **ApplicationDatabaseType**: Either `SqlServer2005`, `SqlServer`, `Oracle` or `DB2`.
 - i. **ApplicationDatabaseServer**: Enter the server where the database is located.
 - j. **ApplicationDatabasePort**: Enter the port for the database.
 - k. **ApplicationSID**: Enter the SID. Only needed for Oracle.
 - l. **Application Database Schema**: Enter the schema of the database. Only needed for DB2 and Oracle.
 - m. **ApplicationDatabaseUserName**: Enter the database user

name.

- n. **ApplicationDatabasePassword:** Enter the database password.



The screenshot shows the Oracle Administration console interface. At the top, the Oracle logo is displayed in red. Below it, there are three tabs: "Palette Versions", "Palette Properties", and "Administration". The "Administration" tab is selected. Underneath, there is a section titled "Environment Details" with a blue circular icon. This section contains a list of configuration parameters, each with a text input field or a dropdown menu. The parameters and their values are: PaletteVersion (9.7.0.0.4), ApplicationEnvType (Development), EncryptionType (SHA-256), EncryptionIterationCount (1000), DebuggerWebServiceUrl (empty), DebugUserName (empty), DebugPassword (empty), ApplicationDatabaseType (Oracle), ApplicationDatabaseServer (slc04kxc.us.oracle.com), ApplicationDatabasePort (1522), ApplicationSID (qa961), ApplicationDatabaseUserName (oipa_doc), and ApplicationDatabasePassword (represented by 10 black dots).

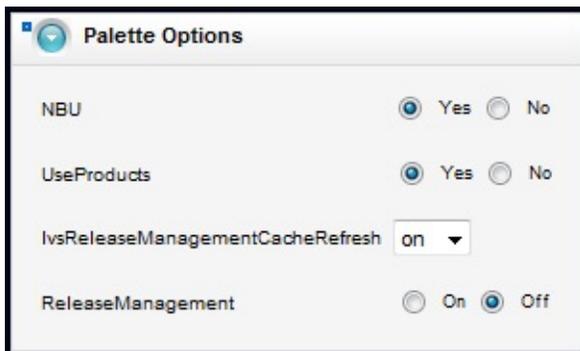
Parameter	Value
PaletteVersion	9.7.0.0.4
ApplicationEnvType	Development
EncryptionType	SHA-256
EncryptionIterationCount	1000
DebuggerWebServiceUrl	
DebugUserName	
DebugPassword	
ApplicationDatabaseType	Oracle
ApplicationDatabaseServer	slc04kxc.us.oracle.com
ApplicationDatabasePort	1522
ApplicationSID	qa961
ApplicationDatabaseUserName	oipa_doc
ApplicationDatabasePassword	••••••••••

4. Select the **Yes** radio button for IVS if using an IVS environment.
5. Enter the IVS environment information.
 - a. **IvsDatabaseType:** SqlServer2005, SqlServer, Oracle or DB2.
 - b. **IvsDatabaseServer:** Server where the database is located.
 - c. **IvsDatabasePort:** Port of the database.
 - d. **IvsSID:** The SID of the IVS database. Only needed for Oracle.
 - e. **IvsDatabaseSchema:** Schema of the database.
 - f. **IvsDatabaseUserName:** Enter the IVS database user name.

- g. **IvsDatabasePassword**: Enter the IVS database password.
- h. **IvsEnv**: Name of the IVS environment that will be used.
- i. **IvsTrackNumber**: Track number of the IVS environment that will be used.
- j. **IvsSequence**: Defines the position of the environment in the Release Management track. Release packages will need to be deployed to environments with an IVS Sequence of "2" before those with an IVS Sequence of "3" and so on.

The combination of **IvsEnv**, **IvsTrackNumber** and **IvsSequence** must be unique to each specific environment.

- 6. Select whether the environment should have New Business Underwriting functionality available.
- 7. Select whether the environment should use Products.
- 8. Select whether the environment should use Release Management.
- 9. If Release Management was turned on in the previous step, select whether the environment is a Release Management Entry Environment.



Palette Options fields on Web Application Utility

- 10. Select **Save**.
- 11. Send the host, port and password information to the Rules Palette users so that they can set-up the environment.



Send the location of the jtlds.jar file if users have a SQL Server database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Getting Started > Environments > Delete an Environment

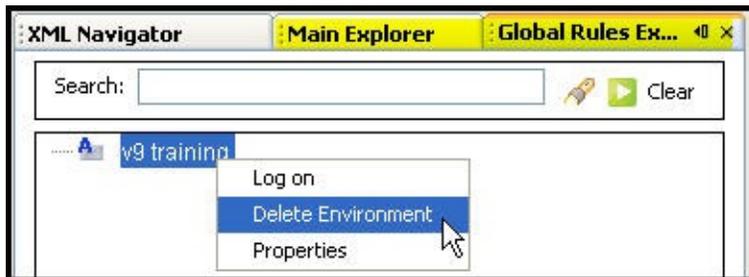
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Delete an Environment

When an environment connection is no longer needed the environment can be deleted. Log out of the environment in order to delete it. If an environment is logged in, then the delete option will not be available on the right-click menu.

Steps to Delete an Environment

1. Right-click on the environment that should be deleted. Make sure the environment is logged out.
2. Select the **Delete Environment** option.
3. Select the **Yes** button.
4. Confirm the delete. The environment connection will be removed from the Main Explorer and Global Rules Explorer tabs.



Right-click Option to Delete Environment

You are here: Getting Started > Log-On

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Log On to Rules Palette

After [creating an environment connection](#) users may log into the Rules Palette. To log on, right-click on the appropriate environment alias name in either the **Global Rules Explorer** or **Main Explorer** tab and select **Log on**. Then enter the Client ID and password. The log on information is case sensitive.

Currently Environment properties are stored locally in the Rules Palette's "asenv.properties" file. If a user attempts to change the environment properties manually in the "asenv.properties" file (outside of the Web Application Utility), then a warning message will appear when the user tries to log into the environment. This occurs because the Rules Palette checks the properties settings saved in the Web Application Utility each time a user logs on. If the properties in the asenv.properties file do not match what is stored in the Web Application Utility, then the following error message will appear:

Environment Properties do not match. Log on will continue and properties will be reset.



Right-click Option to Log On an Environment

After logging on, four tabs are available: Main Explorer, [Global Rules Explorer](#), [Admin Explorer](#) and the [XML Navigator](#). If the tabs are not visible, click **Window** on the menu bar and select the tab to open it.

Once all the rules are loaded into the Rules Palette, expand the Environment and access the folders in the navigation structure. The

Rules Palette will load rules on an as-needed basis. As folders are selected, additional information is loaded. This is done to reduce load-time.

Empty Folders

If a folder in the navigation tree appears to be empty, then check the bottom right corner of the screen to see if the rules have been fully loaded. Wait until the loading message is not visible in the bottom right corner and then try to access the folder. The global rules must finish loading before folders can be expanded.



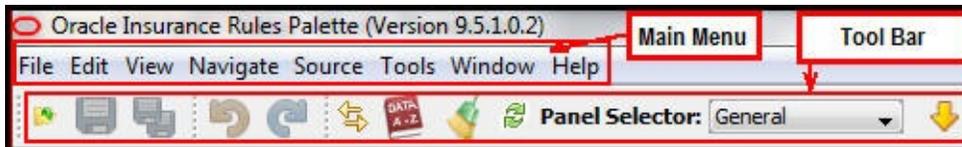
Loading Message for Global Rules

You are here: Getting Started > Navigation and Screen Display

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Navigation and Screen Display

There are several ways to navigate through the Rules Palette. The Main Menu at the top of the screen contains a Window option for opening and closing various windows. The [Tool Bar](#) provides access to short cut icons, which can also open functionality.

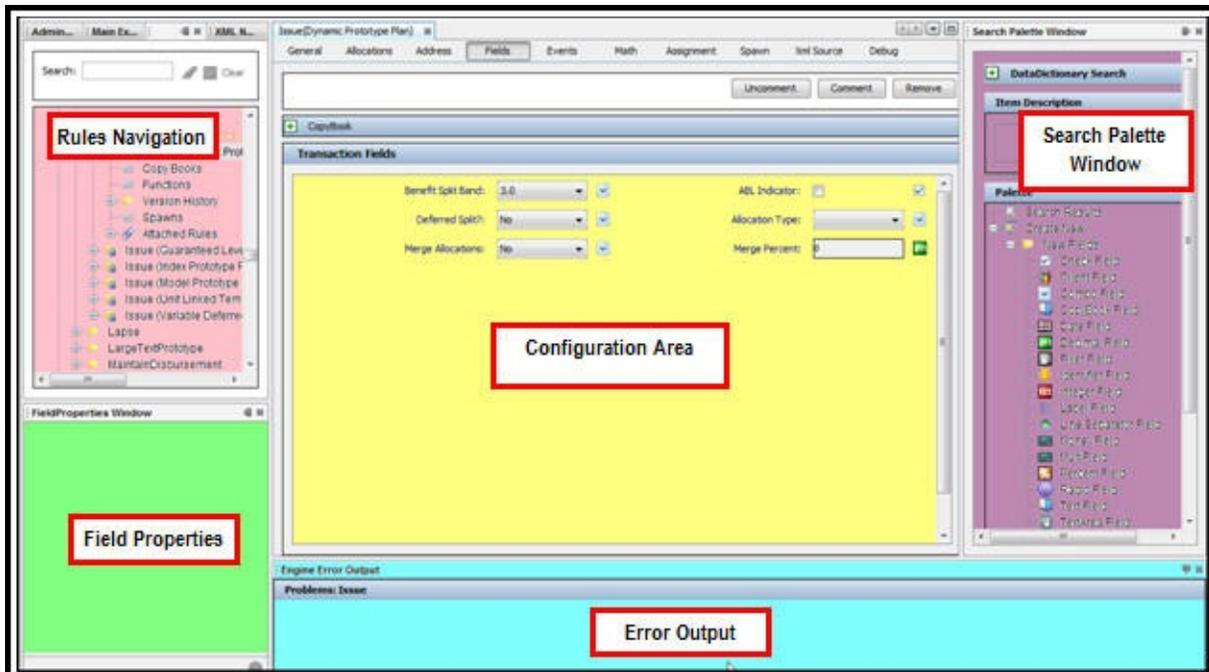


Rules Palette Main Menu and Tool Bar

There are five main sections of the screen:

1. **Rules Area:** contains four tabs. Click a tab to access the folder structure contained in that Explorer. If a tab is not visible, it can be accessed from the Window option on the Main Menu. An explanation of each Explorer is provided below.
 - Admin Explorer: organized by task. Allows the user to perform administrative tasks.
 - Main Explorer: organized by company and plan in a navigation tree structure. Allows the user to view transactions, segments and business rules from a company and plan perspective.
 - Global Rules Explorer: organized by type of global rule in a navigation tree structure. Allows the user to view all global transactions, segments and business rules used in the Rules Palette.
 - XML Navigator: helps the user navigate through the XML. Search for specific HTML tags using this tool.
2. [Configuration Area](#): where rules, transaction and segments are configured.
3. **Palette Area:** contains a list of all available fields and math variables. These are populated according to the pane open in the Configuration Area. If the Fields pane is open, then field elements will display. If the Math pane is open, then the math variable elements will display.

4. **Miscellaneous Area:** where various configuration options will display depending on the current configuration task being performed. For example, the [FieldProperties Window](#) will open in this area.
5. **Error Output Area:** where errors are displayed.
6. **Docking Area:** where docked windows are displayed and can be expanded.

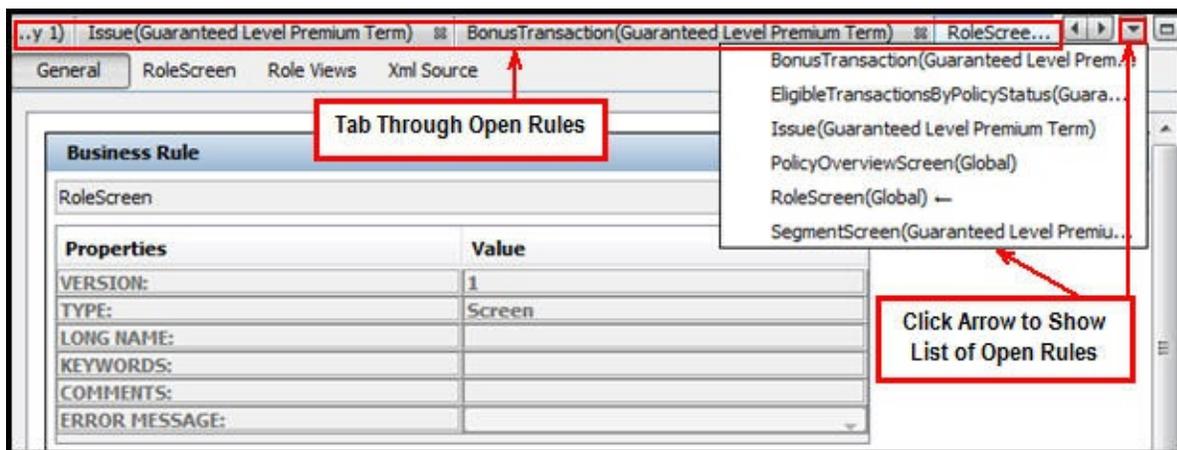


Five Main Sections of Rules Palette Highlighted in Various Colors

Navigating in Rules

Use the Explorer window to find specific business rules, segments or transactions. Double-click the XML file of a rule, segment or transaction to open the file in the Configuration Area. Multiple items may be open simultaneously. A tabbing system allows users to toggle between the open items. The downward-facing arrow button on the top right of the screen also allows users to toggle between rules.

Use the double arrow button  located in the upper right corner of the screen to open tabs if they are not visible on the pane. Move back and forth between tabs in either direction by clicking on the left/right arrows.



Tabbing System at the Top of the Configuration Area

You are here: Getting Started > Tool Bar

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Tool Bar

The Tool Bar is available directly under the Main Menu and provides quick access to commonly used functionality. Shortcut buttons are only available when they apply to what is currently being configured.

	Create New Environment	Create a new environment . Create an environment by right-clicking in the Main Explorer or Global Rules Explorer tab.
	Save	Saves changes to the configuration currently being worked on but does not save all open configuration changes. This does not write to the database. Select Check-in to write to the database.
	Save All	Saves and compiles all configuration changes. This does not write to the database. Select Check-in to write to the database. Important: Save All will save and compile any configuration that is open.
	Upload XML Schemas	Upload rule XML schemas into the database directly through the Rules Palette interface.
	Cut	Cut configuration.
	Copy	Copy configuration.
	Paste	Paste configuration.
	Undo	Undo configuration changes.
	Redo	Redo configuration changes.
	Find	Find term in configuration.
	Select in Explorer	Locates the rule's XML file in either the Main Explorer tab or Global Rules Explorer tab. A rule must be open and a pane in the Configuration Area must be selected in order for this feature to work.
	Data Dictionary	Opens Data Dictionary management.
	Clear Cache	Clears the cache, restores the settings to default, deletes all settings and closes the application.
	Refresh	Refreshes plan types and sub-folders.
Panel Selector: <input type="text" value="General"/>	Panel Selector	Lists all panes in the Configuration Area. Use this drop down box to open a pane that is not visible from the Configuration Area.

You are here: Getting Started > Upload XML Schemas

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Upload XML Schemas

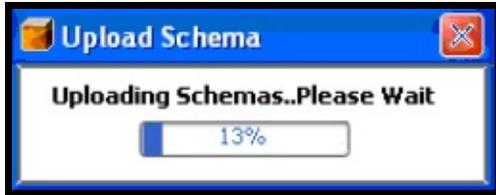
XML schemas are used by the Rules Palette to perform several functions. They determine the various categories as well as the rules that are placed into those categories. The XML schema also controls the dynamic editor used in the configuration of certain rules. Once a schema has been uploaded, the various categories become enabled and the appropriate icons appear for each of them.



Categories are Enabled and Icons are Present

After the initial Rules Palette installation and the [environment creation set-up](#), the default schemas should be uploaded and applied to the system. If schemas are not present in the system then the categories in the Explorer window will be disabled and no icons will be displayed.

New schemas are uploaded by clicking the **Upload Schemas** button  located in the Tool Bar. If schemas are not present in the system, the following pop-up window appears after the initial log-in when the Upload Schemas button is clicked:



Upload Schemas Progress Bar

This pop-up will only appear the first time a user logs into the newly set-up environment. Once the default schemas are uploaded they will remain in the database.

Overwrite Existing Schema

The Upload Schemas button  allows users to upload business rule XML schemas into the database directly through the Rules Palette interface. When this button is clicked, the default schema or other existing schemas in the database are overwritten. The Question pop-up window will appear if the environment already contains schemas and the Upload Schemas toolbar button is selected.

The system will confirm that the existing schema should be overwritten.

- **Yes** will overwrite the schema. This should be performed if new rules are added to the database, or if existing schemas that may have invalid XML need to be updated.
- **No** will cancel the request.



Any time new schemas are uploaded, the user must log out of the environment and log back in to ensure the schemas are uploaded correctly to the database. If a user attempts to upload schemas when not logged into an environment then a message will instruct the user to log into the environment.

You are here: Getting Started > Clear the Cache

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Clear the Cache

If an earlier version of the Rules Palette exists on the user's computer, then it is strongly advised that the cache be cleared after a new version is installed. To clear cached data, select the broom icon from the shortcut bar.



Clear Cache Button on Tool Bar

You are here: Getting Started > XML Navigator Tab

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

You are here: Getting Started > Movable Windows and Pane System

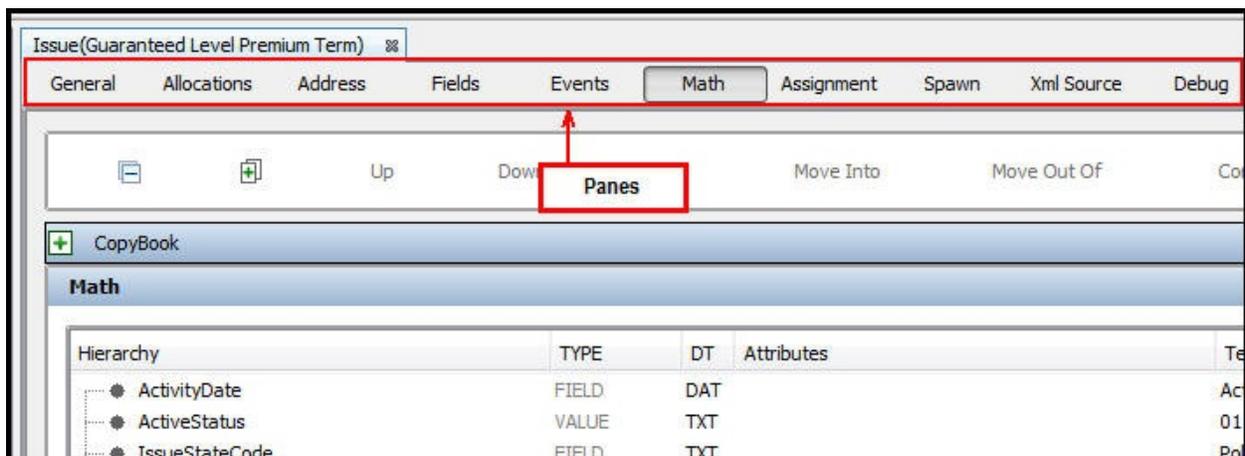
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Movable Windows and Panes

The window and pane layout in the Rules Palette can be manipulated to suit individual preferences. Actions that manipulate the windows and panes of the Rules Palette are listed below.

A window is a section of the screen that segregates groups of information. One example of a window is the [Search Palette](#) window. The Search Palette window contains DataDictionary search features and Palette visual configuration tools.

A pane is part of the [Configuration Area](#) that divides the XML Configuration into sections. Each pane provides access to a specific portion of the XML. One example of a pane is the Math pane, which provides access to the <Math> section of the XML and all visual editing tools relating to math variables.



Panes in Configuration Area

To change the size of a window:

Hover over the edge of a window and left-right arrows  or up-down arrows  will appear. Click and drag the window until it is the desired size.

To dock or undock a window:

Click the minimize button  and the window will collapse and appear along the side of your screen. Click the undock button  to move the window back onto the screen.

To close a window:

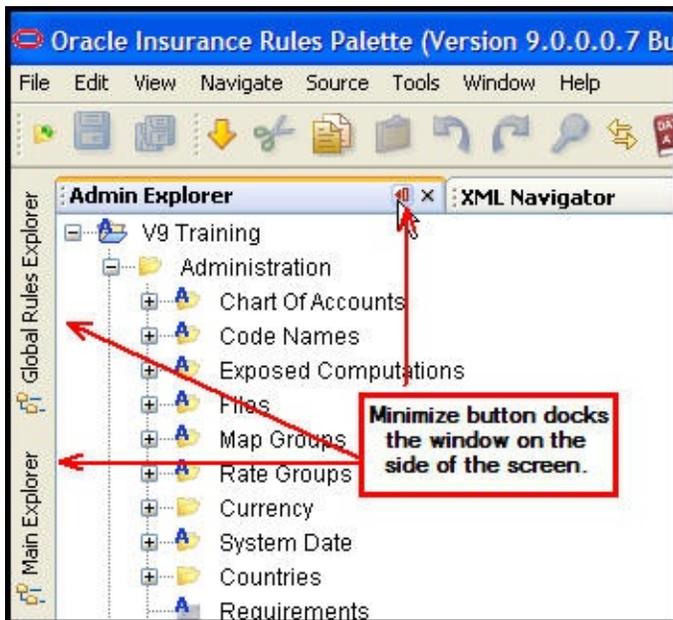
Click the close button  to close a window and remove it from the screen.

To open a window that has disappeared from the screen:

Click the **Window** option on the Tool bar. A drop down list will contain a list of all available windows. Click the name of the window to open.

To open items in the Configuration Area:

The Admin Explorer, Main Explorer and Global Rules Explorer tabs hold navigation trees containing all XML files that can be edited through the Rules Palette. Double-click an XML file in the navigation tree to open it in the [Configuration Area](#).



Windows Docked in the Rules Palette

You are here: Getting Started > Configuration Area

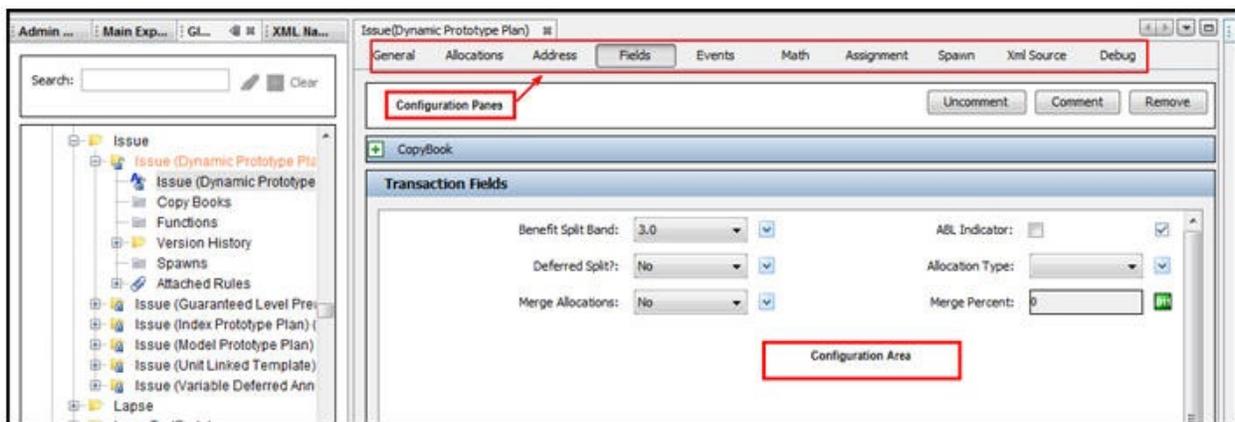
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Configuration Area

The Configuration Area is used to create and modify configurations. The contents of the area changes depending on the type of configuration being performed. The buttons across the top are called **panes** and correspond to each section of configuration for that rule. By selecting a pane, the Configuration Area displays configuration tools for that particular section.

Drag and drop features and selection options can be used to visually configure rules, transactions and segments. If a user prefers to work directly in the XML, then the XML Source pane should be used.

In order to edit in this pane, the rule must be [checked-out](#). If the rule is not checked-out, then the information is available for viewing but not for editing.



Panes Along the Top of the Configuration Area

You are here: Getting Started > XML Source Editing

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

XML Source Editing

Although the Rules Palette offers visual rule configuration, the option to configure in XML is still available through the XML Source tab. When a rule is opened, select the XML Source pane and configure as if using a standard XML editor.

The Line numbering can be turned on and off by selecting or deselecting **Show Line Numbers** and the Editor toolbar can be turned on or off by selecting or deselecting **Show Editor Toolbar** from **View** on the menu bar.



XML Source Pane with XML Editing Tools Highlighted

XML Editor Tools

The XML Editor tools are available to aid configuration. An explanation of each tool is provided below. The tools are explained in order from left to right.

- **Last Edit:** This moves the cursor back to the last edit that was performed.
- **Back:** This moves the cursor back one step.
- **Forward:** This moves the cursor forward from previous steps taken.
- **Find Selection:** This finds each selection of the occurrence of the highlighted text.
- **Find Previous Occurrence:** This finds the previous occurrence of the highlighted text.
- **Find Next Occurrence:** This finds the next occurrence of the highlighted text.
- **Toggle Highlight Search:** This turns on and off all highlighted instances of the searched text.
- **Shift Line Left:** This shifts highlighted text to the left.
- **Shift Line Right:** This shifts highlighted text to the right.

You are here: Getting Started > SearchPalette Window

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Search Palette Window

The **Search Palette** window is used to drag and drop fields and math variables into the [Configuration Area](#). You can search for fields and math variables that have been added to the Data Dictionary by using the [Data Dictionary Search](#) section or you can create new fields and math variables.

Results from a **Data Dictionary Search** will be viewable in the Palette section under Search Results. Use drag and drop functionality to move variables found in the Data Dictionary Search results into either the Fields or Math panes of the Configuration Area.

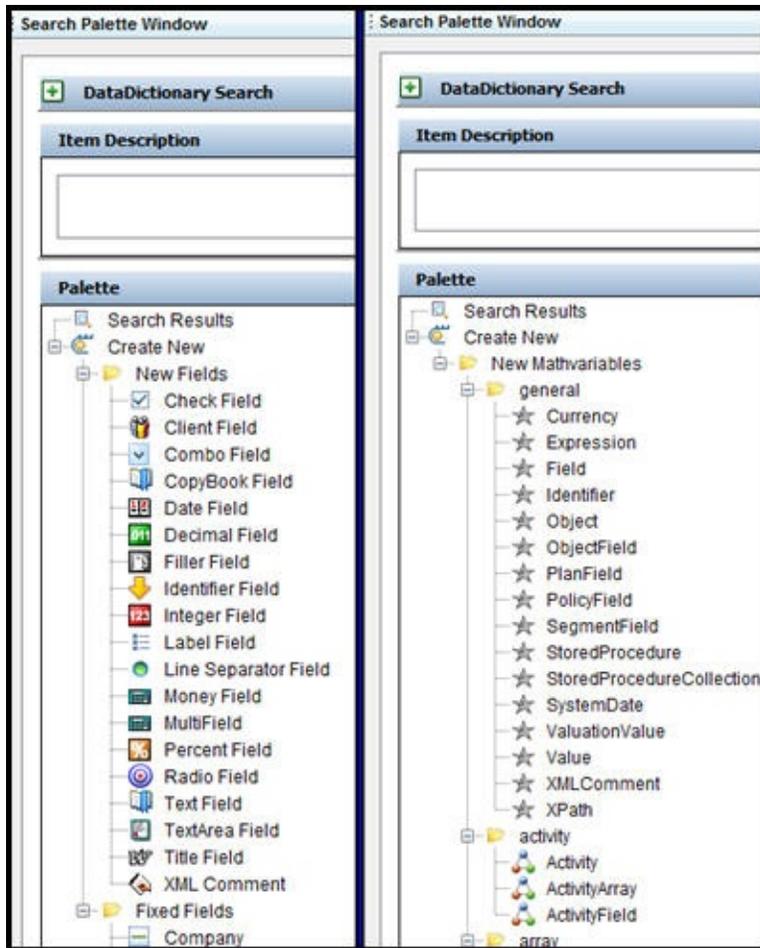


A search can only be performed through the Search Palette window if a transaction or rule is checked-out. The Fields pane in the Configuration Area also needs to be open.

When search results display, click the book  icon to open the Data Dictionary and view the details for that particular term.

Another way to create new fields and math variables is to use the **Palette** window. All available fields are listed in the Palette window when configuring the Field pane. When configuring math using the Palette window, the math variable types are divided into folders according to type. Expanded folders display the types available and can collapse to help in locating the appropriate folder. Once the appropriate math variable or field is selected, drag and drop it from the **Create New** folder onto the Configuration Area.

Search Palette Window for Fields and **Search Palette Window for Math**



Search Palette Fields and Search Palette Math Variables

You are here: Getting Started > Search Using Explorer Tabs

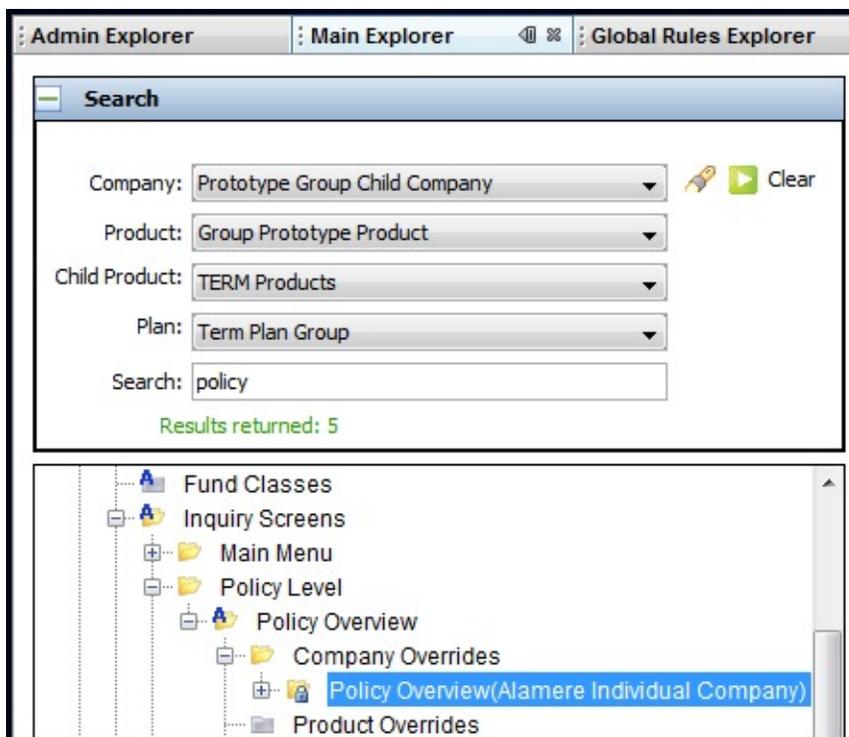
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Search Using Explorer Tabs

Search functionality is available in **Main Explorer**, **Global Rules Explorer** and **XML Navigator** tabs. The Search box at the top of the tab can be used to enter a specific search term. All locations where that term appears in that tab will be displayed. This is a much easier process than hunting through the tree directory structure for the term.

To perform a search, simply enter the search term in the Search field and click the Search button . If the search term is found, the search results will be highlighted in the expanded tree directory structure and a message will appear beneath the Search field displaying the number of results found.

Select the Next Result  button to the right of the Search button to automatically go to the next result in the list. Double-click on the name of a file to open it in the [Configuration Area](#).



Search Results

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Getting Started > Engine Error Output Window

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

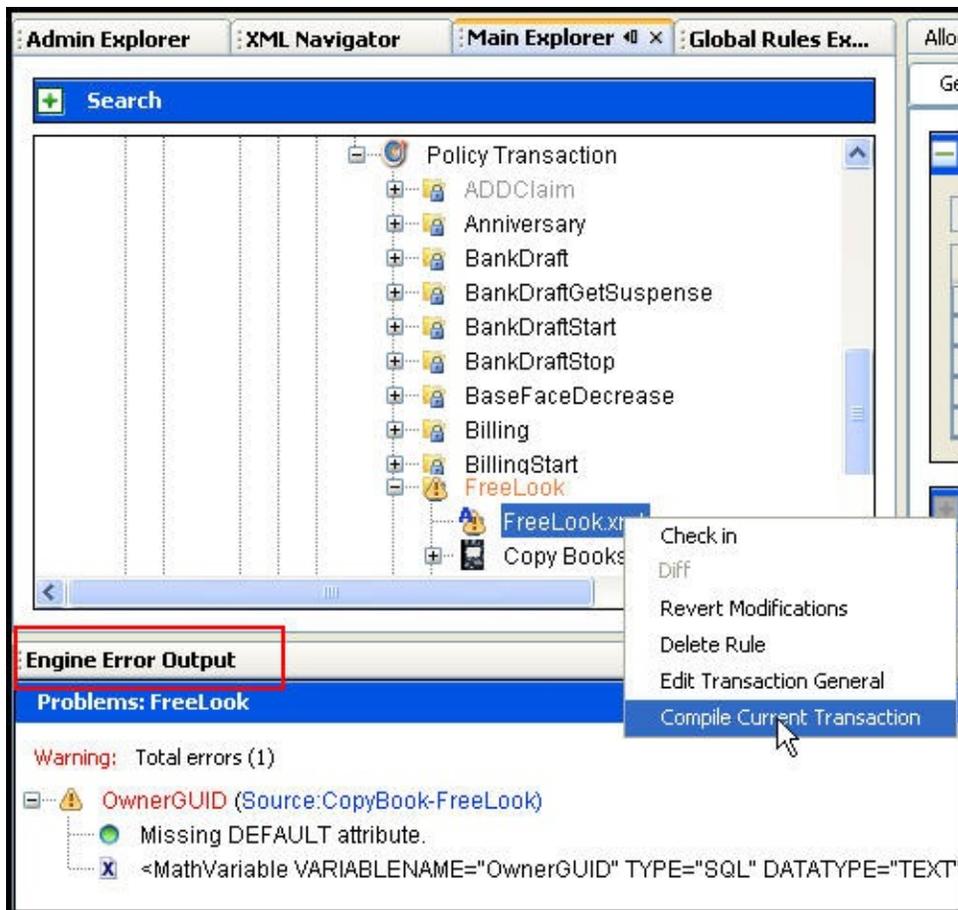
Engine Error Output Window

The Engine Error Output window displays errors that are found when a transaction, segment or calculate general rule is compiled. It also displays validation errors that occur when checking-in or saving information in the Admin Explorer tab. If this window is not visible, click **Window** on the Main Menu and select it.

There are four types of errors that are listed in the Engine Error Output window.

1. **Translation Error** is an error that was caused by a syntax error in the math section. All translation errors will be listed at one time.
2. **Compilation Error** is an error with the configuration in the math section. In this situation, the compile feature stops compiling the configuration of a transaction after it locates this type of error. Only one compilation error at a time is handled. After resolving the compilation error and selecting the **Save** button , the compiler runs again through the configuration and locates the next compilation error. This process repeats until all errors have been resolved.
3. **Runtime Error** is a system code error.
4. **Warning** doesn't keep transactions or segments from compiling. It simply provides an alert regarding missing information or information that is not critical for compilation.

Refer to [Compile a Transaction or Segment](#) for the steps involved in compiling.



Engine Error Output Window

You are here: Security > Security Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Security

The Rules Palette controls security for both the Rules Palette and for the Oracle Insurance Policy Administration (OIPA) system.

Palette Security

Security for the Rules Palette is configured at the user level, where an administrator can add users and associate them to roles. Roles are created by the administrator to specify privileges that allow access to different sections of the Rules Palette. These sections include the administration pane, rules, and data dictionary. [View more...](#)



The Rules Palette Security folder under Security in the Admin Explorer tab is not available in non-IVS environments. The Application Security folder will be the only folder visible.

Application Security

Configuration of OIPA security can limit access to fields, buttons, menu items, screens, plans and activities. Access to an entire screen can be denied or limited. Limited access means that access to individual buttons and fields can be granted or denied. Users can also be restricted from performing certain tasks at the activity level. Security in OIPA allows multiple levels of restriction for different users of the system, from agents to service representatives and department managers. Users may also belong to multiple levels of security as deemed necessary by the administrator. [View more...](#)

Field security is also available. When no selection is made, the default setting allows all fields and the values they hold to be visible and editable. Field security supports the following options, based on a user's role:

1. **Visible and Editable:** The user can view and update a field.
2. **Disable Field Value:** The user can view the field, but not update.
3. **Hide Field Value:** The user cannot see the field value. The value held in the field is hidden from view.



Palette Security and Application Security Folders

You are here: Security > Rules Palette Security

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Rules Palette Security

The Rules Palette controls all security for both the Rules Palette and for the Oracle Insurance Policy Administration (OIPA) system. The [Admin Explorer](#) contains a security folder, which holds security controls for the Rules Palette and OIPA. Only users with **Administration Palette Security Check-In/Check-Out** privileges will see the security folder.

Palette Security privileges are configured in two steps:

1. [User Security](#): contains a list of all users and the security roles assigned to them.
2. [Security Role](#): contains a list of all security roles and the privileges each role has. These roles are for users performing configuration tasks in the Rules Palette.

The security manager first creates specific security roles, such as BA, Configurator, Build Manager, etc. These roles grant users access to the parts of the Rules Palette that pertain to their specific job functions. Once the roles are defined, each user can be assigned a security role. [Security roles and privileges can be modified](#) at any time from the **User Security** or **Security Role** folder. Modifications can be made and saved by selecting the Save All  button. This will save the modifications to the database. In addition, any panes open for configuration with changes will be saved to the local drive.

Transaction and Company Security

[Security needs to be added to new transactions](#) and companies after they are created. The user who created the new transaction or company should notify a security manager when it is created. A security manager will then need to add the transaction or company to the security group for the appropriate users. If this step is not completed, then users will not be able to see the new transaction or company in OIPA.

Release Management Security

[Security roles need to be created for the release management process.](#)

Typically, BAs have access to the configuration package configuration. Then Build Managers and/or BAs with Administration privileges have access to create migration sets and release packages. Separate levels can also be set for a Release Manager so that he can deploy the release package. If a user is not given release management privileges, then he/she will not be able to see any of the release management functionality in the Rules Palette.

Security Role

Security Role is where the [security roles are created for users](#). There are pre-configured access privileges that can be combined to create security roles. The Rules Palette offers visual editing options when the **Security Role** option is double-clicked in the **Security** folder.

Available privileges within Rules Palette security are:

Available Privileges

Definition	
Administration - All Non-Security Administration - CheckIn/CheckOut	Check-in/Check-out all non-security items. The Administration folder will display in the Admin Explorer tab. Items in the folder can be modified.
Administration - All Non-Security View	View all non-security items. The Administration folder will display in the Admin Explorer tab. Items in the folder cannot be modified.
Administration - Palette Security- Check-In/Check-Out	Check-in/Check-out all Rules Palette security items. This privilege allows user to add security to transactions and apply security roles to users of the Rules Palette. The Security folder will display in the Admin Explorer tab.
Administration - OIPA Security - Check-In/Check-Out	Check-in/Check-out all OIPA security items. This privilege allows user to set security around items in OIPA. The Security folder will display in the Admin Explorer tab.
Business Rules - All Non-Administration Rules - Check-In/Check-Out	Check-in/Check-out all non-administration business rules.
Configuration Package - Add	Release Management security item. Allows user to add new configuration packages. The Release Management folder will display in the Admin Explorer tab.
Configuration Package - Put On Hold	Release Management security item. Allows user to put configuration packages on hold. The Release Management folder will display in the Admin Explorer tab.
Configuration Package - Ready to Migrate	Release Management security item. Allows user to migrate configuration packages. The Release

	Management folder will display in the Admin Explorer tab.
Configuration Package - Remove	Release Management security item. Allows user to remove a component from a configuration package. The Release Management folder will display in the Admin Explorer tab.
Configuration Package - Take Off Hold	Release Management security item. Allows user to take configuration packages off hold. The Release Management folder will display in the Admin Explorer tab.
Configuration Package - View All	Release Management security item. Allows user to view configuration packages created by all users. The Release Management folder will display in the Admin Explorer tab.
Configuration Package - ViewOwned	Release Management security item. Allows user to view only the configuration packages that he/she creates. The Release Management folder will display in the Admin Explorer tab.
DataDictionary - Check-in/Check-out	Check-in/Check-out DataDictionary terms.
DataDictionary - Manager	Manage the DataDictionary.
DataDictionary - View	View DataDictionary items.
Detached Migration - Deploy	Release Management security item. Allows a user to access the Detached Migration tool from the Rules Palette Tools menu.
	Release Management security item. Allows user to create migration sets.

Migration Set - Create	The Release Management folder will display in the Admin Explorer tab.
Migration Set - View All	Release Management security item. Allows user to view migration sets created by all users. The Release Management folder will display in the Admin Explorer tab.
Release Package-ApprovePromotion	Release Management security item. Controls whether the user has security rights to approve a release package for deployment in the Target environment.
Release Package - Build	Release Management security item. Allows user to build release packages. The Release Management folder will display in the Admin Explorer tab.
Release Package - Create	Release Management security item. Allows user to create release packages. The Release Management folder will display in the Admin Explorer tab.
Release Package - Deploy	Release Management security item. Allows user to deploy release packages to a new environment. The Release Management folder will display in the Admin Explorer tab.
Release Package - Set Ready for SCM	Release Management security item. Allows user to mark package ready for the Source Control Manager for approval and promotion. The Release Management folder will display in the Admin Explorer tab.
Release Package - Rollback	Release Management security item. Allows user to "rollback" a deployed release package to a non-entry environment in the

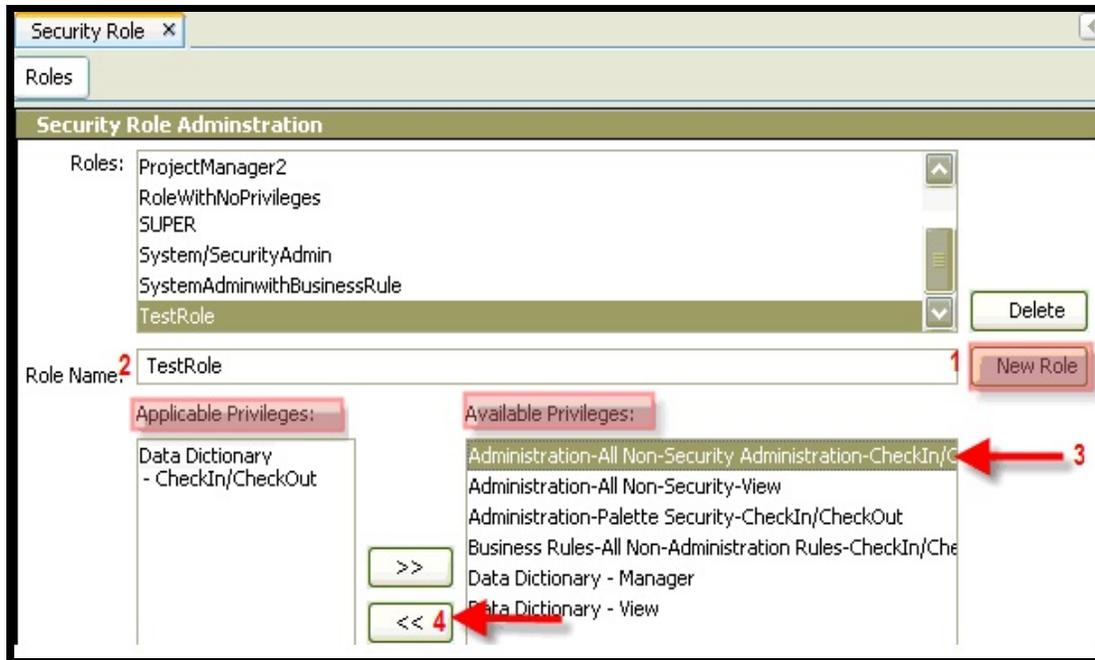
	event of an error or deployment to an incorrect environment.
Release Package - View All	Release Management security item. Allows user to see release packages created by all users. The Release Management folder will display in the Admin Explorer tab.
Utilities-PlanCopy	Plan Copy security item. Enables the Plan Copy option on the Tools menu. Only users with this privilege will be able to use the utility to copy Products and plans.



All users will have view access to all rules as a default.

Steps to Create a New Security Role

1. In **Admin Explorer**, double-click the **Security Role** option.
2. Select the **New Role** button.
3. Enter the name for the new role in the **Role Name** field. The word **Insert** will appear next to the role name to indicate that a new role is about to be added. It will disappear when the new role is saved.
4. Select the desired privileges to be granted to the new role from the list of **Available Privileges**. CTRL-click to select multiple options at a time. Selected items will become highlighted.
5. Use the left arrow  selection button to move selected privileges from the **Available Privileges** list to the **Applicable Privileges** list. The privileges placed in the **Applicable Privileges** list will be the ones that are applied to the selected role after clicking **Save All** . The right arrow  button can be used to remove items from the Applicable Privileges list and return them to the Available Privileges list.
6. Select the **Save All**  button on the Tool bar to save the changes. A window will appear asking for the save to be confirmed.
7. Select **Yes** to save or **No** to cancel



Security Role File

Steps to Delete a Security Role

1. In **Admin Explorer**, double-click the **Security Role** option.
2. Select the role from the **Roles** list that should be deleted.
3. Click the **Delete** button. The word **Delete** will appear next to the role selected for delete. This will disappear when the changes are saved.
4. Click the **Save All**  button on the Tool bar to save the changes.

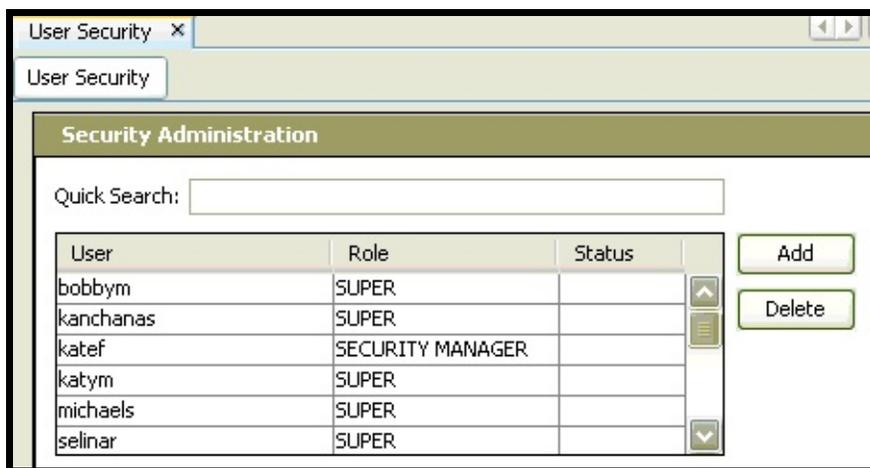


When a Security Role is deleted all users assigned to that role will have Read Only privileges the next time they log into the system.

User Security

The User Security window is the gateway for managing users and their respective security roles. New users can be added and existing users can be deleted using the Add or Delete buttons. Each user can only be assigned one security role.

Enter the first letter or letters of a user's name in the Quick Search option at the top of the window to find that user. Click on the column heading (User, Role or Status) to sort the selected column alphabetically.



User Security File

Steps to Create a New User

1. In the Admin Explorer open **Security | Palette Security | User Security**.
2. Select the **Add** button in the User Security window. Additional fields will appear at the bottom of the window.
3. Enter the new user's name in the UserName field and assign a password in the Password field.
4. Assign a role to the new user in the Role drop-down field. Depending on the role selected, the privileges field will be auto-populated by the system with the respective privileges. The privileges field is not editable.
5. After adding the new user, select the **Save All**  button on the shortcut tool bar. A window will appear to confirm that the changes should be saved to the database. When using the Save All button, changes to security will be saved to the database. In addition, any panes open for configuration with changes will be saved to the local drive.



Refer to [Username and Company Associations](#) for an explanation of how the user is associated to companies in OIPA.

Steps to Delete a User

1. Select the user from the list of available users in the **User Security** window.
2. Select the **Delete** button. A pop-up window will appear asking for confirmation of the delete.
3. Select **Yes** to delete the user or select **No** to cancel and return to the User Security window.

You are here: Security > Security at Product Level

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Security at Product Level

A Product is defined in Group as a Group Benefits Plan Template which provides the basis for a Group Customer's Plan instance and/or serves to group similar Group Benefits Plan Templates (referred to as "Sub-Products"). When Plans are created they will belong to either a Product or Sub-Product. Plans can be created through Group Customer screens or through Palette.

For group policy administration, user has the ability to create plan dynamically using OIPA application. That means user can create plan without using palette and avoids the need of migrating the changes in production environment. But this ability gets defeated as there is no way to give plan level security dynamically. System has the ability to define plan level security at parent product level and all plans should automatically inherit it. This feature will address this need and allow user to create /use plan dynamically without involving any changes to palette. It will also avoid the need for frequent migration of security data from development environment to production environment.

Security Consideration

When a Plan is created through OIPA, plans would receive a copy of parent Product security as default. All levels of security at company and plans as defined in palette , all security levels defined at the immediate parent and all transaction security available to the plan will be copied.

Any Change in permissions for a Plan from this default would need to be made through Palette for that plan

For the copied permissions for the plans created through OIPA to take effect, logout/login back to OIPA is permissible although not preferable

If created in Palette; the security is set up through Plan Security Node in Admin Explorer. When Created through OIPA, at present user is forced to

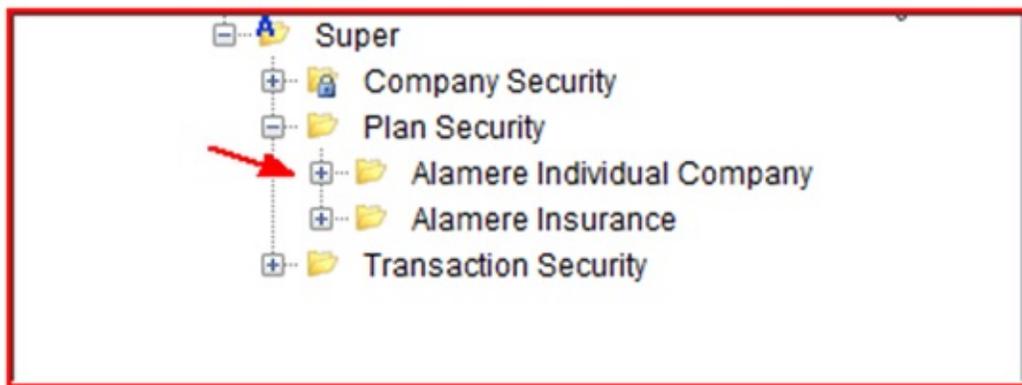
set up security via palette to access the plan pages in OIPA. The palette changes and security changes defined in this document will allow user to set up security at Product level or sub product level which a plan can use as default. For Plans created in OIPA, this would allow user to access plan Pages without the need to set up security permissions via palette.

Setting up Plan Pages security at Product Level or Child Product Level in Palette

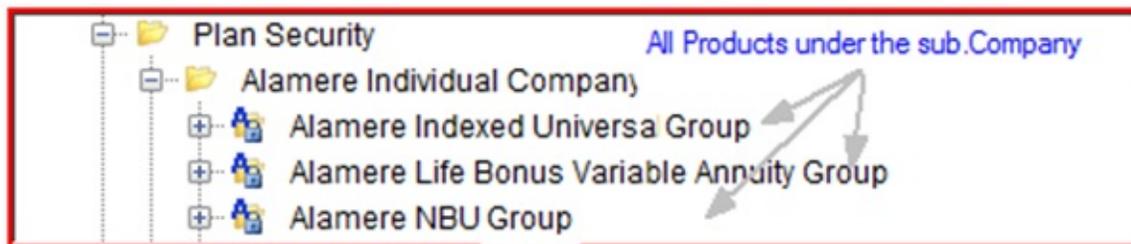
To set up Product level Security, user would need to navigate to the Admin Explorer in Palette

User opens Security | Application Security | Security Group | Name of the Security Group |Plan Security

Expanding Plan Security Node currently lists all companies and subsidiary Companies in OIPA at the same level

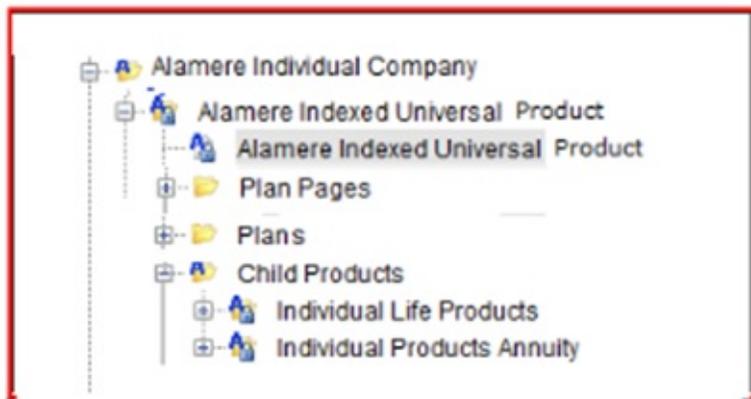


Under each subsidiary company, all products available under it will be displayed



Expanding a Product name will display the following :

1. Product Name Node
2. Plan Pages Folder
3. Plan Folder (Contains individual plan nodes for all plans available under this Product)
4. hierarchically display Child Product Nodes if any

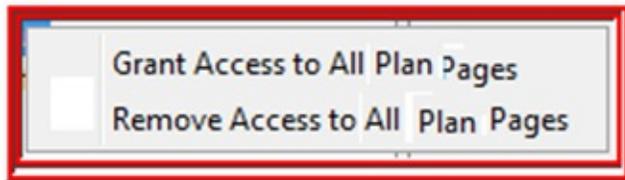


Security can be set up at Product level for the plan pages. Plans under the product will not inherit this security automatically .Any changes to the Plan pages security at product level would NOT propagate downstream. But any plan created in OIPA under the product will copy this security set up as default .

For any Child Products under the Product hierarchy. Plans under the child product will not inherit child product security automatically .Any changes to the Plan pages security at Child product level would NOT propagate downstream . But any plan created in OIPA under the child product will copy this security at set up as default .

When a Product or Child Product or Plans are created in Palette, there will not be any default security. User would need to go to Admin Explorer and set up the security permissions for plan pages explicitly at any level

Provide right click menu options at each parent or child Product name nodes that will allow user provision to Grant Access to all plan pages or Remove access to all plan pages. User can also individually set security by opening plan pages folder and navigating to each plan page security node



Additional to above two right click menu options, provide option at Child Product/s and plan levels on right click to "Copy Parent Access". This will allow option for user to copy a parents security permissions to a child instead of creating from scratch. User can then edit those permissions or keep it as is for that level. Please note that this option to copy parent access is available only if security permissions are available at immediate parent. If security is not set up at immediate parent , do not show this option on right click.

Wherever copy parent access option is allowed and user selects it , give a message which user would confirm with OK as: "Existing security will be overwritten if this operation is completed.

Before a plan security is set up , it is not required to set up Product and Child Product security in the hierarchical structure, but it is up to the configure to take care of the fact that if a plan is created dynamically in OIPA, there needs to be security permissions available at the immediate parent for plans to be accessible for the users without setting up permissions in palette for those plans

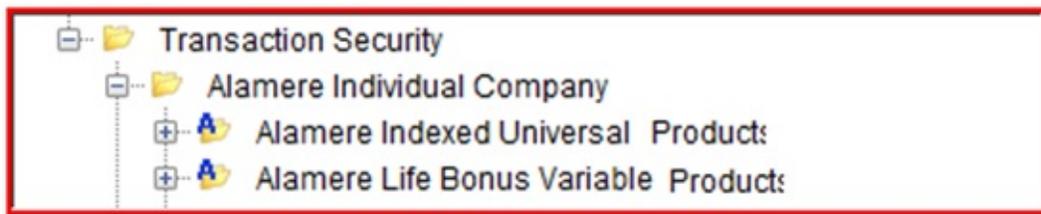
While checking in security permissions at product and Child product level in Palette, give the following warning message when there are children in lower hierarchy (plans or child products) available : "The security permissions will not be automatically copied down to products or plans in lower hierarchy"

All Plan level security set up functionality available today will not change and will continue to be available as is.

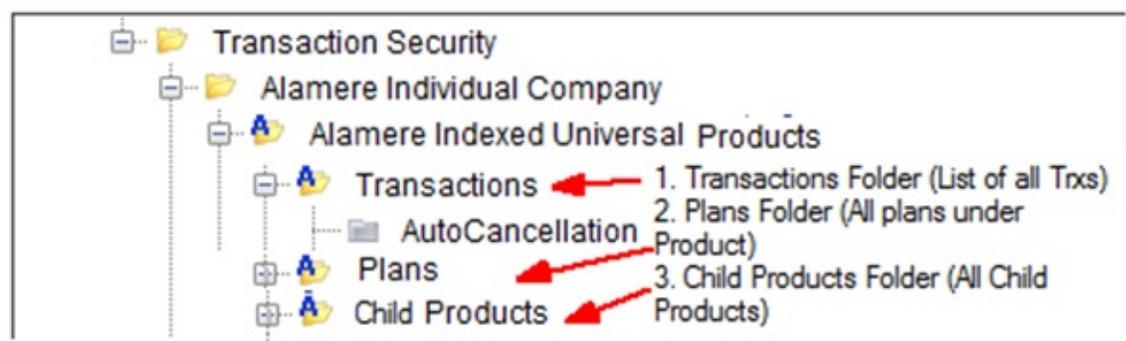
Setting up Transaction Security in Palette at Product/ Child product Levels

After Company, Product and /or Plan security have been defined, the transactions associated with the company, Product and plans are displayed under the Transaction Security folder in Admin Explorer

Currently all companies and subsidiary companies will be displayed under the transaction security node at the same level like in plan security Under each subsidiary company , all product name nodes will appear

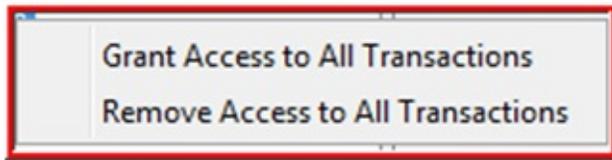


Opening the individual products will display Transactions Folder that would list all transactions under the product , Plans Folder ,Child Products Folder in that specific order. Within Plan Folder, all Plans under the Product will be displayed . Opening the node for the plans will display all plan transactions



Within the child product folder all Child product names available will be displayed. Under each Child Products, the hierarchical display of folders

will be similar to Product, as shown in below image



Security can be added or removed to all transactions in a product by right-clicking on the product name node. Security can also be assigned to individual transactions by opening the Product folder and then the transactions folder and then selecting specific transaction.

Security can be added or removed to all transactions in a child product/s by right-clicking on the child product's name. Security can also be assigned to individual transactions by opening the child Product folder and selecting a specific transaction. Plans under the child product will be displayed under child product hierarchically

Opening up plan folder will list all plans under the Product or child product/s. Security can be set up for Plan transactions as it is currently

When a transaction is created at Product level , those transaction nodes will be added to downstream child products and Plans in Palette currently. When transactions are created at child product level, those transaction nodes are also added to Plans underneath. In palette transaction security , allow option at Child Product and Plans below to copy parent transaction security permissions access on right click. This option will be available only if security permission setting are available for those transactions at immediate parent level. The right click menu option will be "Copy Parent Access ". When user selects this option , give a warning as : "Existing security will be overwritten if this operation is completed. Do you wish to continue?"

While checking in security permissions at product and Child product level transactions in Palette, give the following warning message when there are children in lower hierarchy (plans or child products) available : "The security permissions will not be automatically copied down to products or

plans transactions in lower hierarchy"

Configuration Detail :

This feature has not introduced any new configuration or changes to existing configuration, but plan pages under product and transactions currently within the product hierarchy would need security manually added.

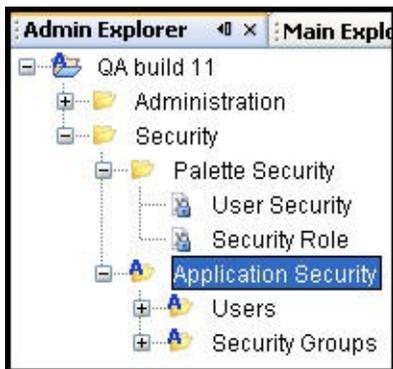
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Security > Application Security > Application Security

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

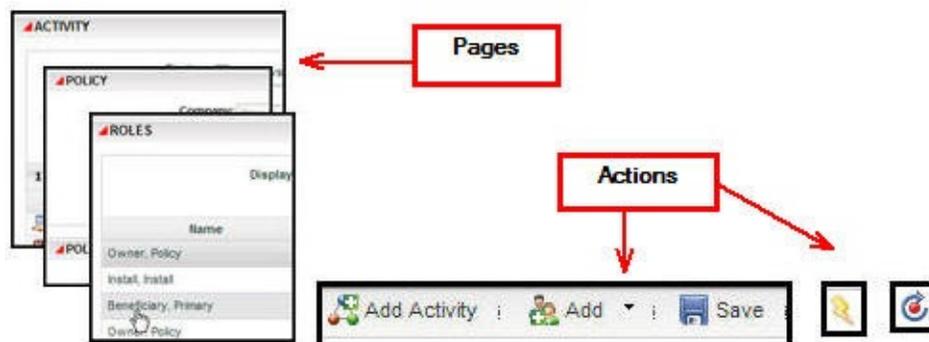
Application Security

Security for the Oracle Insurance Policy Administration (OIPA) system is configured in the Rules Palette. The [Admin Explorer](#) has a link for Application Security, which allows a security manager to enter new user information, edit user information, delete user information and [manage security groups](#).



Application Security Folder in Admin Explorer

OIPA security can be configured at three levels: [company](#), [plan](#) and [transaction](#). Within each level, access can be granted to users for pages, fields, masks and actions.



Levels for OIPA Security

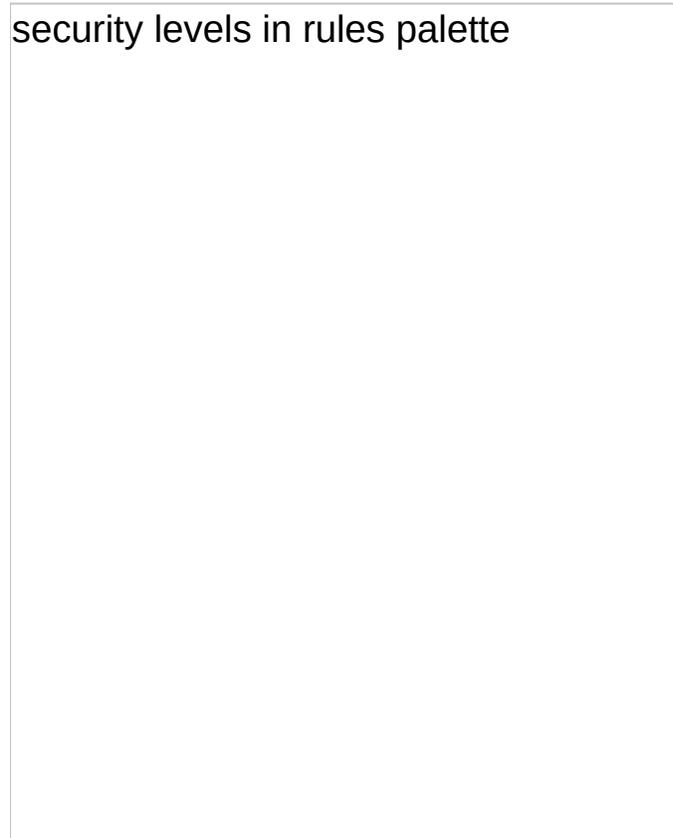
At each level, general access can be assigned at the top, which would allow a user to access all buttons on a particular page. Access can also be assigned at a very granular level, where a user may only be able to access one button on a particular page. Users can also be restricted from

performing certain tasks at the activity level. For example, a user may only be able to process activities on the activity page, but may not be able to delete activities. Security in OIPA allows for precise and detailed levels of restriction for all types of system users.



Each security group should only grant access to one primary company. Multiple primary companies cannot be viewed by one user login in OIPA. If a user needs access to multiple primary companies, then separate security groups should be created for each company and the user should be assigned multiple login IDs.

security levels in rules palette



Security Levels in Rules Palette

You are here: Security > Application Security > Work With Security Groups

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Work With Security Groups

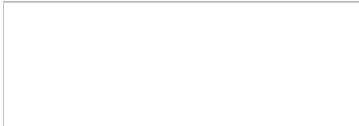
A security group is made up of a set of privileges that are assigned to users, granting them access to pages, plans and transactions (activities) in OIPA. Users that need full access to OIPA can be assigned to a security group that provides the highest level of security to all pages, plans and transactions, giving them unrestricted access to OIPA. Other users may need more restricted access, therefore, a security group can be created to define the level of access those users will need in OIPA. Depending on the granularity with which security groups are defined, users may belong to multiple security groups. In this situation, the actual security provided to a user is the union of the security information.

Where conflicts occur - when security access to a specific item is indicated in more than one security group - the highest level of security defined by the competing security groups is honored.

Security groups can be [created](#), [edited](#) and [deleted](#) from the Application Security folder in the Admin Explorer.

Explanation of Security Folder Organization

The main security folder resides in the Admin Explorer. Security is divided at the top level into two folders: Application Security and Palette Security. Application Security defines the pages, buttons, fields and activities a user can access in OIPA. Palette Security defines the configuration tasks that users can perform in the Rules Palette.



Main Security Folders in Admin Explorer

Application Security Folder Structure

[Application Security](#) is divided into two main folders: Users and Security Groups. The **Users** folder is where users are [created, maintained and assigned to security groups](#).



Application Security Folders in Admin Explorer

The **Security Groups** folder contains all the various combinations of security privileges that can be assigned to OIPA users. Every security group has a unique folder listed under the Security Group section. The security group is divided into three sections: Company, Plan and Transaction.



Security Group Folders in Admin Explorer

- **Company Security**: Identifies the company the security group will be able to access in OIPA. Only one primary company may be identified.
- **Plan Security**: Identifies all the plans associated with the company that was chosen in the Company Security file. Each plan is broken into individual pages where security can be assigned on a page by page basis.
- **Transaction Security**: Places specific security around transactions that will process as activities in OIPA.

Steps to Create a new Security Group

1. Click the Admin Explorer tab and open the **Security | Application Security** folder.
2. Right-click the Security Groups folder and select **New Security Group**.
3. Enter a name for the new security group.



Each security group should only grant access to one primary company. Multiple primary companies cannot be viewed in OIPA. If a user needs access to multiple primary companies, then separate security groups should be created for each company and the user should have multiple login IDs.

4. Select **Finish**. The new security group will be listed in the Security Groups folder.
5. Double-click on the name of the security group that was created. This will reveal the Company Security, Plan Security and Transaction

Security folders. Only the Company Security folder is enabled.

6. Double-click the Company Security folder to reveal the XML file. Double-click the XML file to open it in the Configuration Area.
7. Add or remove any checked options.
8. Check in the XML file to save the changes. All other security folders for the security group are now enabled.

Steps to Edit a Security Group

1. Click the Admin Explorer tab and open the **Security | Application Security** folders.
2. Open the Security Groups folder and double-click on the security group to edit.
3. Right-click on the XML file and select **Check-out**.
4. Select or de-select checkboxes next to the privileges to edit.
5. Right-click on the XML file and select **Check-in**. This will save the changes to the database.

Steps to Delete a Security Group

1. Open the Security folder in the Admin Explorer tab.
2. Open the Security Groups folder.
3. Open the specific security group folder that should be deleted.
4. Right-click on the XML file and select **Delete Security Group**.
5. Select **Yes** when the confirmation messages appears.



Security Groups Right-Click Options

You are here: Security > Application Security > Assign Users to Security Groups

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Assign Users to Security Groups

Security must be assigned to each user of the OIPA system. After the [security groups are created](#), OIPA users are added and associated with a security group. A security manager will need to send the user name and password information to the user once the user has been added.

Steps to Add a New User

1. Click the Admin Explorer tab and open the **Security | Application Security** folders.
2. Right-click on the Users folder and select **Add New**.
3. Enter the user information as shown in the image below. This is where the user's log-in name and password are created. Send this information to the user once it has been saved.
4. Select the primary company the user will be working with.
5. Select the locale where the user is based. The locale determines the language that dynamic fields and transaction names will display in when the user logs into OIPA.
6. Select the Security Group to assign to the user. If the security group is associated with a company that is different from the Primary Company selected in this record, then the user will have access to the company identified in the security group when logged into OIPA. The security group company takes precedence.
7. Select **Finish** when all of the information has been selected. The user information will appear as an individual XML file under the User node.

Steps	User name
1. User name	
Login Name:	<input type="text" value="user1"/>
First Name:	<input type="text" value="First name of user"/>
Last Name:	<input type="text" value="Last name of user"/>
Password:	<input type="password" value="*****"/>
Confirm Password:	<input type="password" value="*****"/>
Gender:	<input type="text" value="F"/>
Email:	<input type="text"/>
Primary Company:	<input type="text" value="Holding Company"/>
Locale:	<input type="text" value="English (Unites States of America)"/>
Security Group:	<input type="text" value="Super"/>

< Back Next > **Finish** Cancel Help

New User Security Wizard

Steps to Edit User Information

1. Click the Admin Explorer tab and open the **Security | Application Security** folders.
2. Open the Users folder.
3. Double-click on the user folder to edit. It will reveal the XML file.
4. Double-click the XML file to open it in the Configuration Area.
5. Make the changes to the user file.
6. Click **Save** on the Tool bar when finished. This will save the changes to the database.
7. Close the file when finished. This is done by clicking the Close icon  next to the name of the file at the top of the Configuration Area.

Steps to Delete a User

1. Right-click on the XML file in the user's folder.
2. Select **Delete User**.
3. Select **Yes** on the Delete Confirmation window. The user will be removed.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Security > Application Security > Username and Company Associations

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Username and Company Associations

When an OIPA user account is created in the Rules Palette, it is associated with a primary company. Each user account has a login name and password. When the OIPA user logs into the application, the primary company associated with that account record will be available for actions in OIPA. The primary company to which a user belongs must also have an associated subsidiary company for a user to be able to log into OIPA.

A user can have multiple accounts, one for each company the user plans to work with in OIPA.

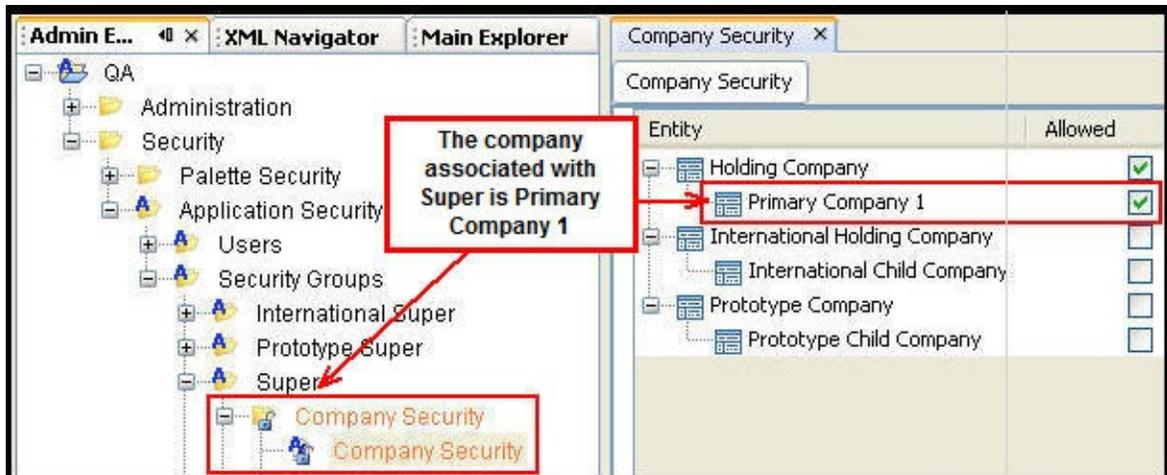
Workflow of Username and Company Associations

1. [Create a new user.](#)

Steps	User name
1. User name	
	Login Name: user1
	First Name: First name of user
	Last Name: Last name of user
	Password: *****)
	Confirm Password: *****)
	Gender: F
	Email:
	Primary Company: Holding Company
	Locale: English (Unites States of America)
	Security Group: Super

New User File with Login Information Highlighted

2. [Make sure the user and the company have been assigned security privileges.](#) The user record has a field for Security Group. This defines the security privileges the user has when logged into OIPA using that user account. In addition to assigning a user to a security group, the company must also be assigned security. This is done through the **Admin Explorer | Security | Application Security**.



Company Security Defined in Admin Explorer Security Folder

3. Log into OIPA using the login name and password associated with the company the user wants to work with in OIPA. The user will be able to work with the company identified in the user account as long as security was assigned to the company.

You are here: Security > Application Security > Company Security > Company Security

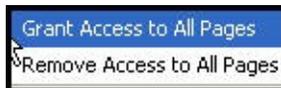
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Company Security

Each security group is associated with one primary company. Company Security is the place where the associated primary company is identified. If more than one company is selected, then a validation error will appear asking the user to restrict the company selection to one company. Errors must be fixed before the Company Security file can be checked-in.

There are three items in the Company Security folder:

- **Company Security file**: this is where the primary company is identified. This company will be visible to users of this security group when logged into OIPA.
- **Company Pages**: this folder contains all company level pages. Right-click on the Company folder under this node to grant access to all company pages or remove access from all company pages. Security can also be assigned on a page by page basis.

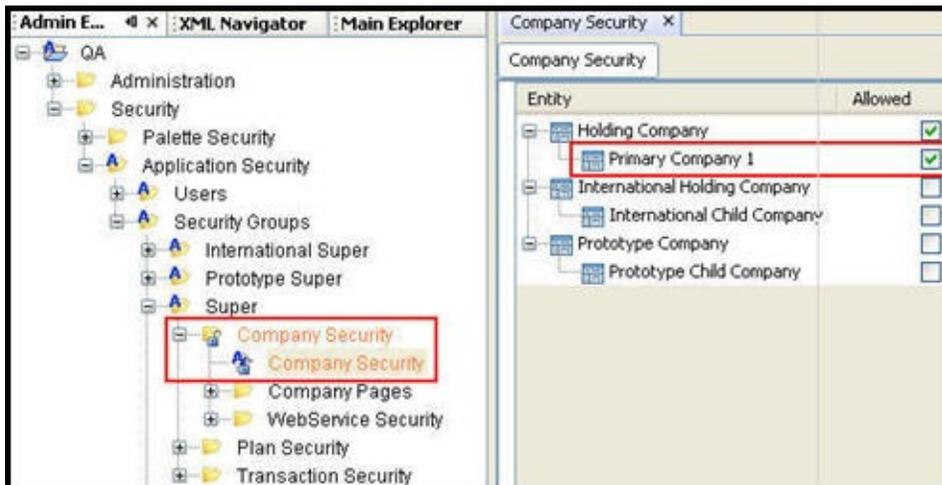


Company Pages Right-Click Option

- **WebService Security**: this folder contains a list of Web Services associated with the company. Security can be assigned to each Web Service individually.

Steps to Edit the Company Security File

1. Navigate to the Admin Explorer.
2. Open **Security | Application Security | Security Group | Name of the Security Group | Company Security**.
3. Right-click on the Company Security file and select **Check-out**. A list of available companies will display in the Configuration Area.
4. Click the check-box next to the company that should be associated with the security group. Multiple subsidiary companies may be selected, but select only one primary company.
5. Right-click on the Company Security file and select **Check-in**.



Company Security in Admin Explorer



Company Available in OIPA

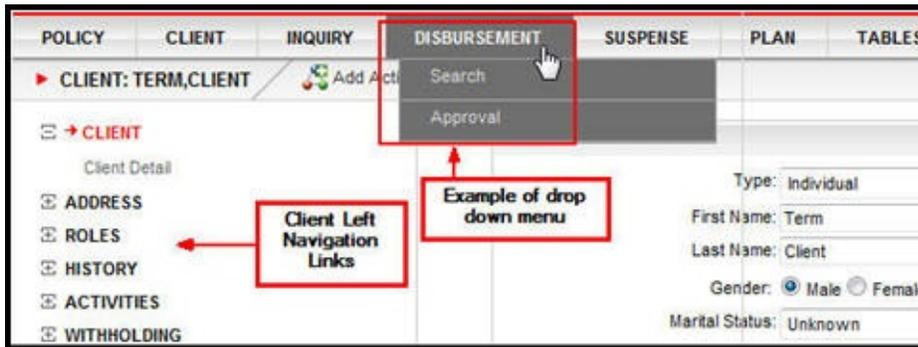
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Security > Application Security > Company Security > Company Pages

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Company Pages

The Company Pages folder contains a separate page for all Client Left Navigation links and all Main Menu buttons and drop down options in OIPA.



Client Left Navigation Links and Drop Down Main Menu in OIPA

Right-click on the name of a specific Company folder under this node to grant access to all pages or remove access from all pages. Security can also be assigned on a page by page basis.



Company Pages Right-Click Options

Security is applied from the top down. Once Company security is defined in the Company file and the Company pages, then the Plan Security folder will populate with available plans. Once Plan security is defined, then the Transaction Security folder will populate with available transactions.

Company Pages

The following is a list of all Company pages, along with an explanation of the button security available for each Company page.

- **AddAgreementProduct:** this page controls the buttons and fields inside the Add Agreement Product window. This window is launched when the Add button is clicked from the Product tab on the Agreement screen.
 - **OK:** this controls the availability of the OK button at the bottom of the Add Agreement Product window.
- **AddAgreementRole:** this page controls the links and buttons inside the Add Agreement Role window. This window is launched when the Add button is clicked from the Roles tab on the Agreement screen.
 - **AddClient:** this opens a new client detail section in the window so new clients can be created and added as roles.
 - **FindClient:** this opens a client search section in the window so existing clients can be found and added as roles.
 - **FindCustomer:** this opens a customer search section in the window so existing customers can be found and added as roles.
 - **OK:** this controls the OK button at the bottom of the Add Agreement Role window.

ADD AGREEMENT ROLE: FULLY INSURED AGREEMENT

Find Client : New Client : Find Customer

AGREEMENT ROLE

Type: Bank Letter

CUSTOMER SEARCH CRITERIA

Group Customer Name:

Legal Name:

Tax ID:

Alternate Name:

Find

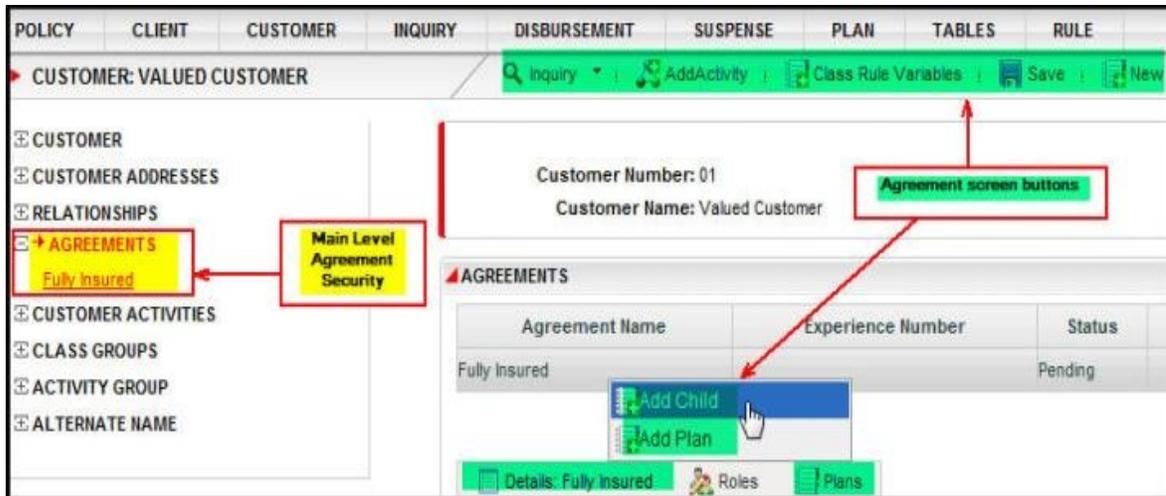
CUSTOMER SEARCH RESULTS

Group Customer Name	Tax ID	Customer Number	Status	InCo
No Results				

Ok

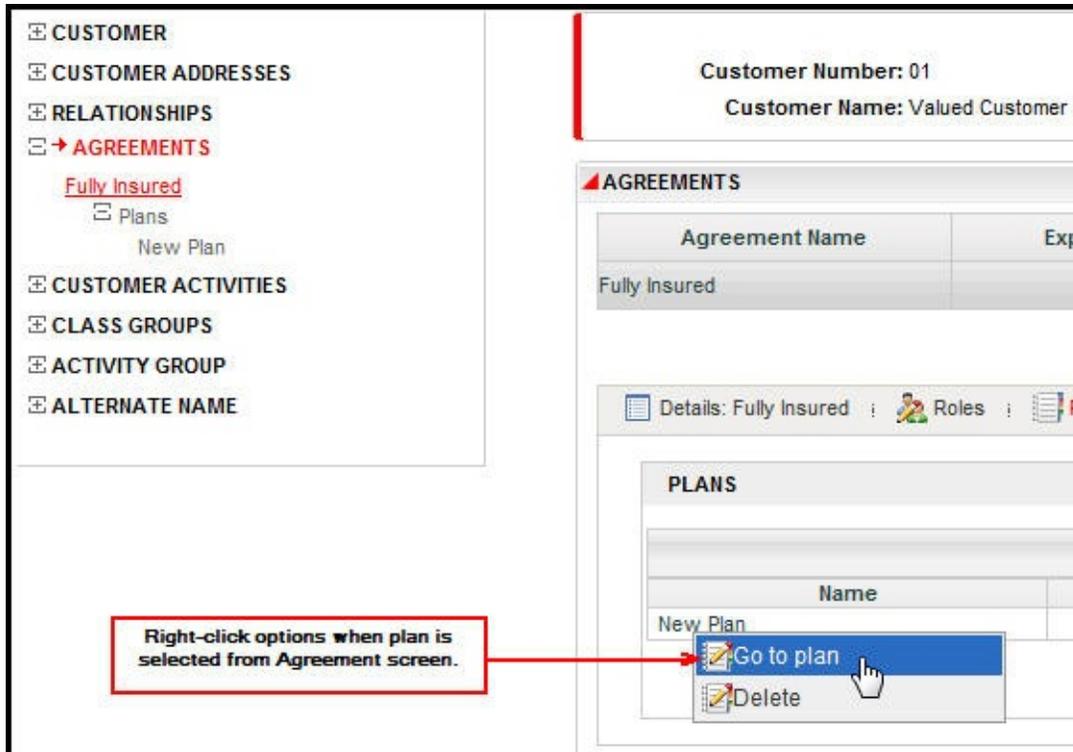
Add Agreement Role Window With Security Applied

- **Agreement:** this page controls the buttons and right-click options on the Agreement screen. The Agreement screen can be accessed when a customer record is open in OIPA. The Customer Left Navigation menu contains an **Agreement** link, which launches this screen.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Agreement screen is open in OIPA.
 - **AddChildAgreement:** this controls the right-click option to add a child agreement. If a user right-clicks on an agreement listed on the Agreement screen, then this option will display.
 - **AddParentAgreement:** this controls the right-click option to add a parent agreement. If a user right-clicks on an agreement listed on the Agreement screen, then this option will display.
 - **AddPlan:** this controls the right-click option to add a plan to the agreement. If a user right-clicks on an agreement listed on the Agreement screen, then this option will display.
 - **AgreementDetail:** this controls the availability of the Details tab on the Agreement screen. The Details tab is at the bottom of the screen next to the Roles tab and the Plans tab.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Agreement screen is visible in OIPA. A client level Inquiry screen must also be configured in order for the link to appear.
 - **NewAgreement:** this controls whether the New link will display on the Secondary menu when the Agreement screen is visible in OIPA.
 - **OpenClassRuleVariables;** this controls whether the ClassVariableRules link will display on the Secondary menu when the Agreement screen is visible in OIPA.
 - **Save:** this controls whether the Save button will display on the Secondary menu when the Agreement screen is visible in OIPA.



Agreement Page and Available Links that Have Security Applied

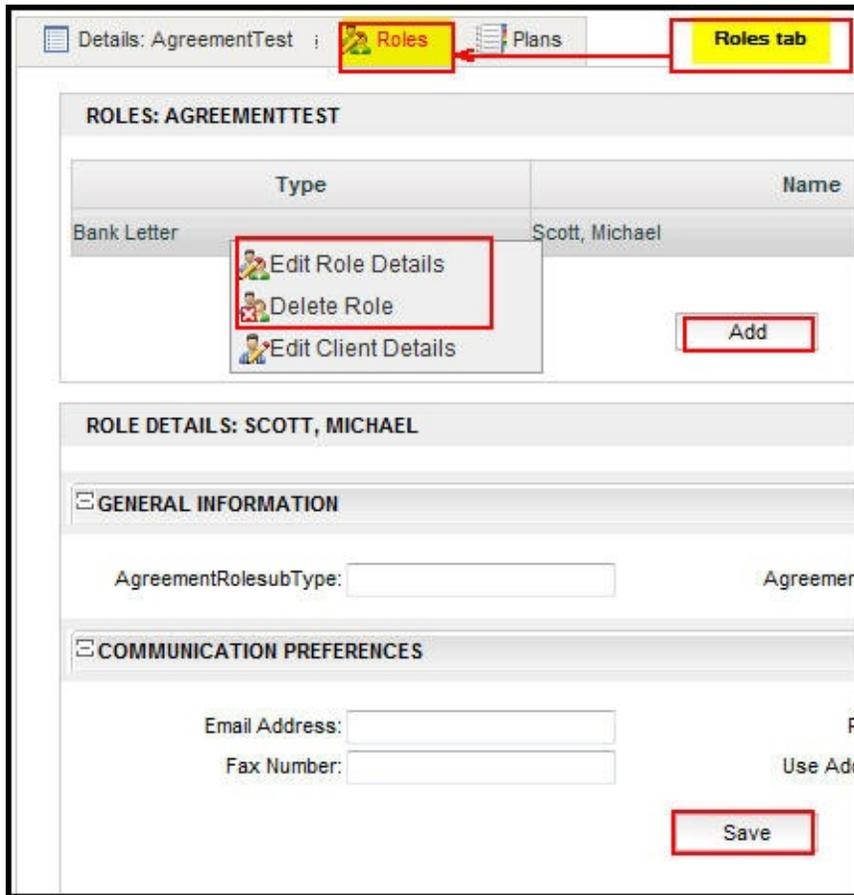
- **AgreementPlans:** this controls the options available on the **Plans** tab of the Agreement screen.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu.
 - **AddAgreementPlan:** This controls the Add Agreement Plan button on the Agreement screen's Plans tab.
 - **DeleteAgreementPlan:** this controls the **Delete** right-click option on the Agreement screen's Plans tab. This option allows the user to delete an agreement plan.
 - **EditAgreementPlan:** this controls the **Edit** right-click option on the Agreement screen's Plans tab. This option allows the user to edit an agreement plan's details.
 - **GoToPlan:** this takes the user to the Plan screen with the selected plan displaying on the screen. This is a right click option available after clicking on the **Plans** tab in the lower section of the Agreement screen.
 - **OpenClassRuleVariables:** this controls whether the **Class Rule Variables** button will display on the Secondary menu.
 - **Save:** this controls the availability of the **Save** button on the Agreement screen's Plans tab. This button allows the user to save an agreement plan.



Agreement Screen With Plan Right-Click Action with Security Added

- **AgreementProducts:** this controls the display of the Products tab at the bottom of the Agreement screen.
 - **AddAgreementProduct:** this controls the display of the Add button on the Products tab.
 - **DeleteAgreementProduct:** this controls the availability of the Delete option on the right-click menu when a Product is selected.
 - **EditAgreementProduct:** this controls the availability of the Edit option on the right-click menu when a Product is selected.
 - **Save:** this controls the display of the Save button on the Products tab.
- **AgreementRole:** this controls the display of the Roles tab at the bottom of the Agreement screen.
 - **AddAgreementRole:** this controls the display of the Add button in the Roles section.
 - **DeleteAgreementRole:** this controls the Delete option on the right-click menu when a role is selected.
 - **EditAgreementRole:** this controls the Edit option on the right-click menu when a role is selected.

- **Save:** this controls the Save button at the bottom of the Roles section.



Roles Tab on Agreement Screen with Security Applied

- **AlternateName:** this controls the display of the Left Navigation link called Alternate Name on the Customer screen. Open a customer record in OIPA to see this link in the Left Navigation menu.
 - **AddAlternateName:** this controls the display of the **New** link on the Secondary menu of the Customer screen.
 - **Save:** this controls the display of the **Save** link on the Secondary menu of the Customer screen.



Alternate Name Link in OIPA with Security Applied

- **Case:** this controls the buttons that display on the Case screen.
 - **Save:** this controls the display of the Save button on the Case screen.

- **ClassGroup:** this Company Page controls the buttons that appear on the Class Group screen, which displays when the Class Group link is selected on the left navigation panel on the Customer screen. Configurable controls are:
 - **Add:** this controls display of the button used to add a time slice for a class group, which displays above the class group time slice table.
 - **Delete:** this controls the "Delete" time slice action, which is controlled from within the class group time slice table.
 - **Diff:** this controls display of the "Diff" button used to view the differences between time slices. This button displays above the class group time slice table.
 - **Edit:** this controls the "Edit" time slice action, which is controlled from within the class group time slice table.
 - **Inquiry:** this controls display of the **Inquiry** button on the secondary menu.
 - **ClassGroupClasses:** this controls display of the **Classes** tab on the Class Group screen.
 - **ClassGroupDetail:** this controls display of the **Class Group**

Detail tab on the Class Group screen.

- **OpenClassRuleVariables:** this controls display of the **Class Rule Variables** button on the secondary menu.
- **Return to Draft:** this controls the "Return to Draft" time slice action, which is controlled from within the class group time slice table.
- **Save:** this controls display of the **Save** button on the secondary menu.
- **Submit:** this controls the "Submit" time slice action, which is controlled from within the class group time slice table.



Class Group Screen showing security applied

- **ClassGroupClasses:** this Company Page controls actions related to classes on the Class Group screen.
 - **AddChildClass:** this controls the ability to add a child class.
 - **NewClass:** this controls the ability to create a new class.
 - **CopyClass:** this controls the ability to copy a class.
 - **DeleteClass:** this controls the ability to delete a class
- **ClassGroupDetail:**
 - **Save:** this controls display of the **Save** button.
- **ClassGroupClassRuleVariables:** this controls the user's ability to use the buttons on the class group level Class Rule Variable screen.
 - **AddVariable:** this controls display of the **Add Variable** button.
 - **Delete:** this controls display of the **Delete** option on the context menu for the Class Rule Variables table.
 - **Insert:** this controls display of the **Insert** options (**Insert Above** and **Insert Below**) on the context menu for the Class Rule Variables table.
 - **Save:** this controls display of the **Save** button.



Class Rule Variables table on Class Group Screen showing security applied

- **ClassMember:** this Company Page controls the display of the buttons available on the Members tab of the Class screen.
 - **MemberDetail:** this controls users' ability to view Member details.
- **ClassRule:** this controls the user's ability to use the buttons on the Class Rule screen.
 - **AddVariable:** this controls display of the **Add Variable** button.
 - **Delete:** this controls display of the **Delete** option on the context menu for the Class Rule Variables table.
 - **Insert:** this controls display of the **Insert** options (**Insert Above** and **Insert Below**) on the context menu for the Class Rule Variables table.
 - **Save:** this controls display of the **Save** button.
- **Client:** this page controls the buttons and Secondary menu links on the Client screen in OIPA.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Client screen is visible in OIPA.
 - **ClientComments:** this controls whether the AddComment link is available on the Secondary menu of the Client screen. You will not be able to actually enter information for a comment until the [ClientComments](#) Company Page privilege is also added.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Client screen is visible in OIPA. A client level Inquiry screen must also be configured in order for the link to appear in OIPA.
 - **Phone:** this controls whether the **PhoneNumbers** link will display on the Client screen.
 - **Save:** this controls whether the Save link will display on the Secondary menu when the Client screen is visible in OIPA.
- **ClientActivity:** this page controls the buttons that display when the Left Navigation menu Activity link is clicked from a Client record in OIPA.
 - **ActivityComments:** this controls the display of the comment icon to the right of an activity listed on the Client Activity screen.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Client Activity screen is visible in OIPA.

- **AllowDelete:** this controls whether the Delete checkbox will display on the Client Activity screen.
- **AutoProcess:** this controls whether the AutoProcess checkbox will display on the Client Activity screen.
- **ClientComments:** this controls whether the AddComment link is available on the Secondary menu of the Client Activity screen. You will not be able to actually enter information for a comment until the [ClientComments](#) Company Page privilege is also added.
- **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Client Activity screen is visible in OIPA. A client level Inquiry screen must also be configured in order for the link to appear in OIPA.
- **ProcessAll:** this controls whether the Process All button will display on the Client Activity screen.
- **Save:** this controls whether the Save button will display on the Secondary menu when the Client Activity screen is visible in OIPA.
- **ClientAddress:** this page controls the buttons that display when the Left Navigation menu Address link is clicked from a Client record in OIPA.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Client Address screen is visible in OIPA.
 - **AddAddress:** this controls whether the New link will display on the Secondary menu when the Client Address screen is visible in OIPA.
 - **ClientComments:** controls whether the AddComment link is available on the Secondary menu of the Client Address screen. You will not be able to actually enter information for a comment until the [ClientComments](#) Company Page privilege is also added.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Client Address screen is visible in OIPA. A client level Inquiry screen must also be configured in order for the link to appear in OIPA.
 - **Save:** this controls whether the Save link will display on the Secondary menu when the Client Address screen is visible in OIPA.

- **ClientComments:** this page controls the buttons that display on the Client Comments screen.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Client Comments screen is visible in OIPA.

There is a **Comment Buttons** configuration section, which controls actions related to comments. The following comment button security is available for each action:

- **Add:** this controls whether client comments are able to be added.
 - **View:** this controls whether client comments are able to be viewed, or what type of client comments (i.e. all comments or only those made by the current user) are able to be viewed. If **Add** is set to **Yes**, then No is not an option for this action.
 - **Update:** this controls whether client comments are able to be modified, or what type of client comments (i.e. all comments or only those made by the current user) are able to be modified. If UpdateAllComments is selected, then ViewAllComments must also be selected.
 - **Delete:** this controls whether client comments are able to be deleted, or what type of client comments (i.e. all comments or only those made by the current user) are able to be deleted. If DeleteAllComments is selected, then ViewAllComments must also be selected.
 - **View History:** this controls whether a client's comment history is accessible, or what type of client comment (i.e. all comments or only those made by the current user) history is able to be viewed. If ViewAllCommentsHistory is selected, then ViewAllComments must also be selected.
-
- **ClientCommentsSearch:** this page controls the display of the Comments link on the Left Navigation menu of the Client screen. When the link is clicked, it allows you to search for client comments.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the client's Comments Search screen is visible in OIPA.

- **ClientComments**: controls whether the AddComment link is available on the Secondary menu of the Client Search screen. You will not be able to actually enter information for a comment until the [ClientComments](#) Company Page privilege is also added.
- **Refresh**: this controls whether the Refresh button will display on the client's Comments Search screen.
- **Shadow**: this controls whether the Shadow checkbox will be enabled on the client's Comments Search screen. This checkbox is used to specify if shadowed comments will be returned in the search results.
- **ClientGroup**: this page controls the Left Navigation menu option for Group on the Client screen.
 - **AddActivity**: this controls whether the Add Activity link will display on the Secondary menu when the Client Group screen is visible in OIPA.
 - **AddClientGroup**: this controls whether the New link will display on the Secondary menu when the Client Group screen is visible in OIPA.
 - **ClientComments**: controls whether the AddComment link is available on the Secondary menu of the Client Group screen. You will not be able to actually enter information for a comment until the [ClientComments](#) Company Page privilege is also added.
 - **Inquiry**: this controls whether the Inquiry link will display on the Secondary menu when the Client Group screen is visible in OIPA. A client level Inquiry screen must also be configured in order for the link to appear in OIPA.
 - **Save**: this controls whether the Save link will display on the Secondary menu when the Client Group screen is visible in OIPA.
- **ClientHistory**: this page controls the buttons that display when the Left Navigation menu History link is clicked from a Client record in OIPA.
 - **AddActivity**: this controls whether the Add Activity link will display on the Secondary menu when the Client History is visible in OIPA.
 - **ClientComments**: controls whether the AddComment link is available on the Secondary menu of the Client History screen. You will not be able to actually enter information for a comment until the

[ClientComments](#) Company Page privilege is also added.

- **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Client History screen is visible in OIPA. A client level Inquiry screen must also be configured in order for the link to appear in OIPA.
- **ClientRelationship:** this page controls the availability of actions when the Relationship link is clicked from the Client Left Navigation menu.
 - **Add:** this controls display of the **Add** button on the time slice grid used for adding a new time slice.
 - **AddActivity:** this controls display of the **Add Activity** button on the secondary menu.
 - **AddRelationship:** this controls display of the **Add Relationship** button on the secondary menu.
 - **Delete:** this controls display of the **Delete** button on the time slice grid actions used for deleting a new time slice.
 - **Edit:** this controls display of the **Edit** button on the time slice grid actions.
 - **EditClient:** this controls the Edit Client Details right-click option when a relationship record is clicked on the Relationship screen.
 - **EditCustomer:** this controls the Edit Customer Details right-click option when a relationship record is clicked on the Relationship screen.
 - **Enroll:** this controls the Enroll right-click option when a relationship record is clicked on the Relationship screen.
 - **Inquiry:** this controls display of the **Inquiry** button on the secondary menu.
 - **ReturnToDraft:** this controls the **Return to Draft** button on the time slice grid actions.
 - **Shadow:** this controls the **Shadow** checkbox on top of the time slice grid actions.
 - **Save:** this controls the Save button on the Secondary menu.
 - **Submit:** this controls the **Submit** button on the time slice grid actions.
- **ClientRole:** this page controls the buttons that display when the Left

Navigation menu Role link is clicked from a Client record in OIPA .

- **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Client Role screen is visible in OIPA.
- **ClientComments:** controls whether the AddComment link is available on the Secondary menu of the Client Role screen. You will not be able to actually enter information for a comment until the [ClientComments](#) Company Page privilege is also added.
- **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Client Role screen is visible in OIPA. A client level Inquiry screen must also be configured in order for the link to appear in OIPA.
- **Save:** this controls whether the Save link will display on the Secondary menu when the Client Role screen is visible in OIPA.
- **ClientSearch:** this page controls the availability of the Search drop down option on the Client Main Menu. There are no associated buttons. If this checkbox is left empty, then the Search option will not appear in the Client drop down list on the Main Menu. [Masks](#) are supported on this screen. Security is enabled based on the client type for client and client search functionality as well .This is for the customers who limits few users from viewing the clients of specific types as needed.
- **Enroll:**
- **ClientWithholding:** this page controls the buttons that display when the Left Navigation menu Withholding link is clicked from a Client record in OIPA.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Client Withholding screen is visible in OIPA.
 - **ClientComments:** controls whether the AddComment link is available on the Secondary menu of the Client Withholding screen. You will not be able to actually enter information for a comment until the [ClientComments](#) Company Page privilege is also added.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Client Withholding screen is visible

in OIPA. A client level Inquiry screen must also be configured in order for the link to appear in OIPA.

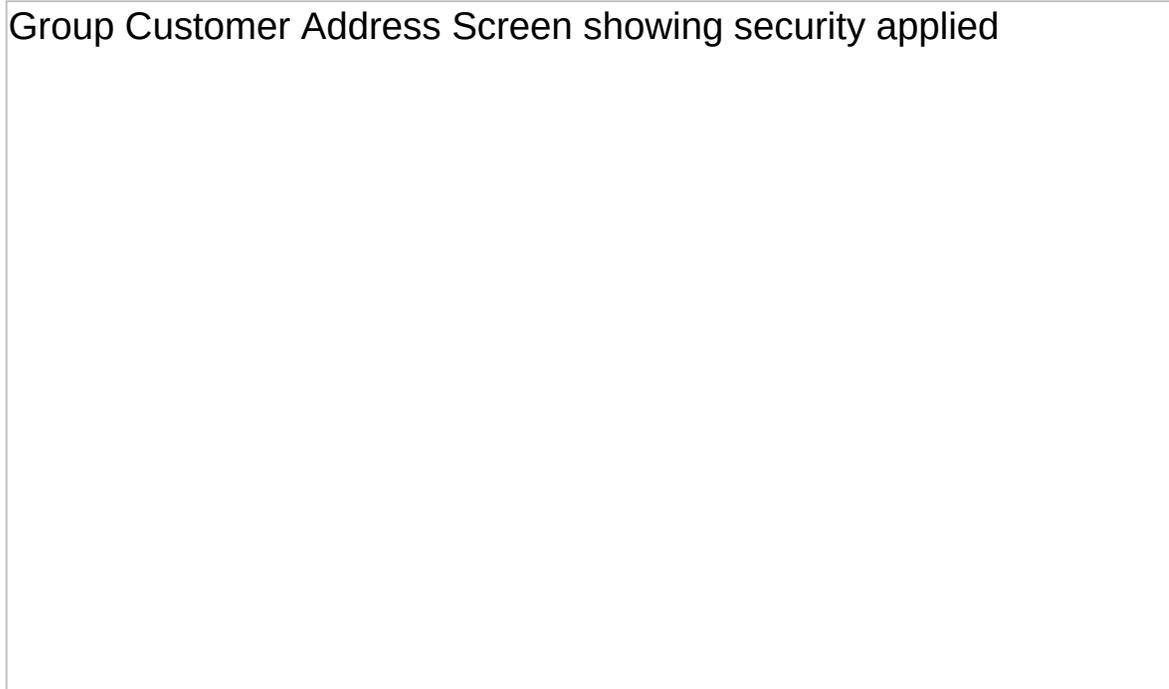
- **Save:** this controls whether the Save link will display on the Secondary menu when the Client Withholding screen is visible in OIPA.
- **CompanyActivity:** this page controls the buttons that display when the CompanyActivity link is clicked from the Company Main Menu in OIPA.
 - **ActivityComments:** this controls the display of the comment icon to the right of an activity listed on the Company Activity screen.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Company Activity screen is visible in OIPA.
 - **AllowDelete:** this controls whether the Delete checkbox will display on the Company Activity screen.
 - **AutoProcess:** this controls whether the AutoProcess checkbox will display on the Company Activity screen.
 - **ProcessAll:** this controls whether the Process All button will display on the Company Activity screen.
- **DisbursementApproval:** this page controls the buttons that display when the DisbursementApproval link is clicked from the Disbursement Main Menu in OIPA. [Masks](#) are supported on this screen.
 - **Find:** this controls whether the Find button displays on the Disbursement Approval screen in the Disbursement Approval Search section.
 - **Save:** this controls whether the Save button displays at the bottom of the Disbursement Approval screen.
- **DisbursementSearch:** this page controls the buttons that display when the Disbursement Search link is clicked from the Disbursement Main Menu in OIPA. [Masks](#) are supported on this screen.
 - **Detail:** this controls whether disbursement details will display when a disbursement record is selected on the Disbursement Search Results screen. If checked, disbursement details will display below the search results for the record that is clicked by the user.
 - **Find:** this controls whether the Find button displays on the

Disbursement Search screen.

- **ApplyApprovalLimits:** this controls whether the screen will use threshold limits for enabling or disabling the 'Approval Status' and 'Disapprove Reason' fields in each disbursement record. Once checked, the application will display the Threshold Limit pane.
 - **Threshold Limit pane:** This pane displays one record where the user can select the Field Name on which threshold limit check will be applied and also allows entering the minimum amount, maximum amount and currency that defines the range that is allowed for a given user profile for approval.
- **Enrollment:** this controls the user's ability to use the buttons on Enrollment screen.
 - **AddActivity:** this controls display of the **Add Activity** button.
 - **Delete:** this controls display of the **Delete** button.
 - **Enroll:** this controls display of the **Enroll** button.
 - **Inquiry:** this controls display of the **Inquiry** button.
 - **Save:** this controls display of the **Save** button.
- **GlobalClassRuleVariables:** this controls the user's ability to use the buttons on the global Class Rule Variable screen.
 - **AddVariable:** this controls display of the **Add Variable** button.
 - **Delete:** this controls display of the **Delete** option on the context menu for the Class Rule Variables table.
 - **Insert:** this controls display of the **Insert** options (**Insert Above** and **Insert Below**) on the context menu for the Class Rule Variables table.
 - **Save:** this controls display of the **Save** button.
- **GroupCustomer:** this page controls the buttons that display when an existing Customer record is opened.
 - **AddActivity:** this controls display of the **Add Activity** button on the secondary menu.
 - **ClassGroupDetail:** this controls the user's ability to open the Class Group Detail screen.

- **Inquiry:** this controls display of the **Inquiry** button on the secondary menu.
- **OpenClassRuleVariables:** this controls display of the **Class Rule Variables** button on the secondary menu.
- **Phone:** this controls whether the **PhoneNumbers** link will display on the Group Customer screen.
- **Save:** this controls display of the **Save** button on the secondary menu.
- **GroupCustomerActivity:** this page controls the availability of actions when the Customer Activities link is clicked on the Customer Left Navigation menu.
 - **AddActivity:** this controls display of the **Add Activity** button on the secondary menu.
 - **AutoProcess:** this controls display of the **Auto-Process** checkbox on the Customer Activities screen.
 - **Inquiry:** this controls display of the **Inquiry** button on the secondary menu.
 - **OpenClassRuleVariables:** this controls display of the **Class Rule Variables** button on the secondary menu.
 - **ProcessAll:** this controls whether the **Process All** button will display on the Customer Activities screen.
- **GroupCustomerActivityGroup**
 - **AddActivity:** this controls display of the **Add Activity** button on the secondary menu.
 - **OpenClassRuleVariables:** this controls display of the **Class Rule Variables** button on the secondary menu.
- **GroupCustomerAddress:** this page controls the availability of actions when the Customer Addresses link is clicked on the Customer Left Navigation menu
 - **AddActivity:** this controls display of the **Add Activity** button on the secondary menu.
 - **AddAddress:** this controls display of the **Add Address** button on the secondary menu.

- **Inquiry:** this controls display of the **Inquiry** button on the secondary menu.
- **OpenClassRuleVariables:** this controls display of the **Class Rule Variables** button on the secondary menu.
- **Save:** this controls display of the **Save** button on the secondary menu.



Group Customer Address Screen showing security applied

- **GroupCustomerClassRuleVariables:** this controls the user's ability to use the buttons on the Group Customer level Class Rule Variable screen.
 - **AddVariable:** this controls display of the **Add Variable** button.
 - **Delete:** this controls display of the **Delete** option on the context menu for the Class Rule Variables table.
 - **Insert:** this controls display of the **Insert** options (**Insert Above** and **Insert Below**) on the context menu for the Class Rule Variables table.
 - **Save:** this controls display of the **Save** button.



Class Rule Variable screen security

- **GroupCustomerRelationship:** this page controls the availability of actions

when the Relationship link is clicked from the Customer Left Navigation menu.

- **Add**: this controls display of the **Add** button on the time slice grid used for adding a new time slice.
- **AddActivity**: this controls display of the **Add Activity** button on the secondary menu.
- **AddRelationship**: this controls display of the **Add Relationship** button on the secondary menu.
- **Delete**: this controls display of the **Delete** button on the time slice grid actions used for deleting a new time slice.
- **Diff**: this controls display of the **View Diff** button on the time slice grid used for viewing differences between two time slices.
- **Edit**: this controls display of the **Edit** button on the time slice grid actions.
- **EditClient**: this controls the Edit Client Details right-click option when a relationship record is clicked on the Relationship screen.
- **EditCustomer**: this controls the Edit Customer Details right-click option when a relationship record is clicked on the Relationship screen.
- **Enroll**: this controls the Enroll right-click option when a relationship record is clicked on the Relationship screen.
- **Inquiry**: this controls display of the **Inquiry** button on the secondary menu.
- **OpenClassRuleVariables**: this controls the Class Rules Variables link on the Secondary menu.
- **ReturnToDraft**: this controls the **Return to Draft** button on the time slice grid actions.
- **Save**: this controls the Save button on the Secondary menu.
- **Shadow**: this controls the **Shadow** checkbox on top of the time slice grid actions.
- **Submit**: this controls the **Submit** button on the time slice grid actions.
- **GroupCustomerRelationshipActivity**: this page controls the availability of the activity actions on the Relationship screen when a Group Customer

record is open.

- **AddActivity**: this controls display of the **Add Activity** button on the secondary menu.
- **AutoProcess**: this controls display of the **Auto-Process** checkbox on the Relationship Activities screen.
- **Inquiry**: this controls display of the **Inquiry** button on the secondary menu.
- **ProcessAll**: this controls whether the **Process All** button will display on the Relationship Activities screen.
- **Save**: this controls the Save button on the Secondary menu.
- **GroupCustomerRelationshipSearch**: this page controls the Search link under the Left Navigation menu link for Relationships on the Group Customer screen. If this link is present, then users will be able to search by primary relationships, secondary relationships, relationship status and any combination of the three.
 - **Find**: this controls the Find button on the Group Customer Relationship Search screen.

The screenshot displays a web application interface with a top navigation bar containing tabs for POLICY, CLIENT, CUSTOMER, INQUIRY, DISBURSEMENT, SUSPENSE, and PLA. Below the navigation bar, the main content area is titled 'CUSTOMER: VALUED CUSTOMER'. On the left side, there is a navigation menu with expandable sections: CUSTOMER, CUSTOMER ADDRESSES, RELATIONSHIPS (highlighted in red), AGREEMENTS, CLASS GROUPS, ACTIVITY GROUP, and ALTERNATE NAME. Under the RELATIONSHIPS section, a 'Search' button is highlighted with a red box. The main content area on the right shows customer details: 'Customer Number: 01' and 'Customer Name: Valued Customer'. Below this, there is a 'RELATIONSHIP SEARCH CRITERIA' section with three dropdown menus: 'Primary RelationshipType' set to 'All Primary Relationship Types', 'Secondary RelationshipType' set to 'All Secondary Relationship Types', and 'Relationship Status' set to 'Active'. A 'Find' button is highlighted with a red box at the bottom right of the search criteria section.

Relationship Search and Find Buttons in OIPA

- **GroupCustomerSearch**: this page has no controls that require setting security.
- **IntakeFileSearch**: this page controls the buttons and actions available for

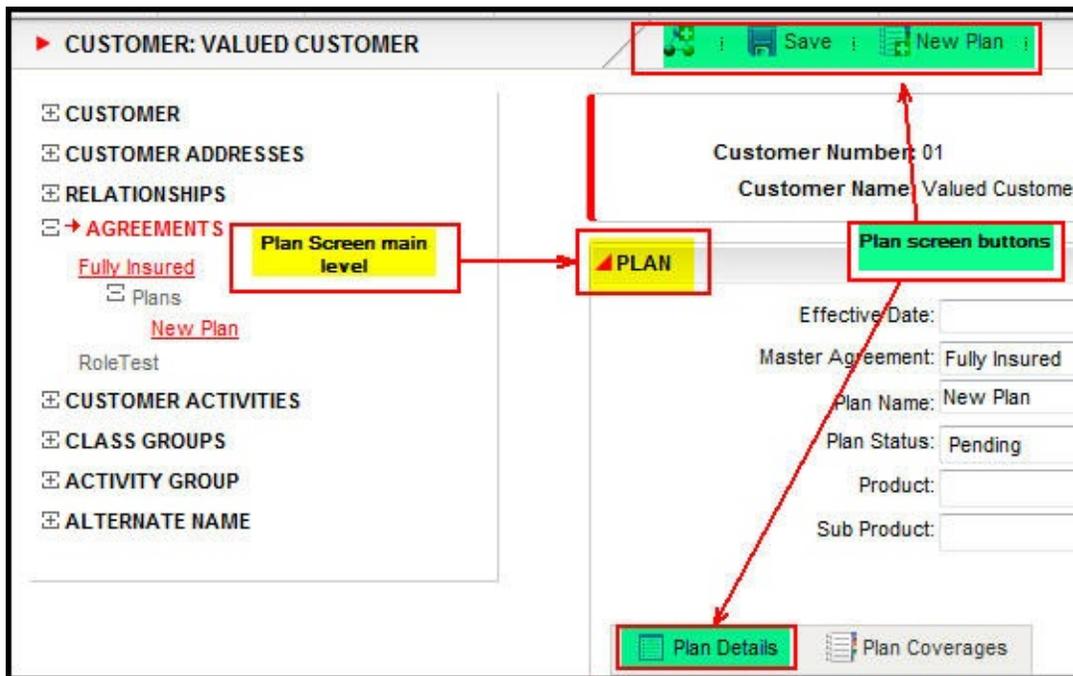
selection on the Intake File Search screen.

- **Delete:** this controls the availability of the Delete icon in the Actions column for the most recent File received for the group customer that has completed processing.
- **AddActivity:** this controls the availability of the Add Activity button in the Secondary Menu of the Intake File Search screen.
- **Inquiry:** this controls the availability of the Inquiry button in the Secondary Menu of the Intake File Search screen.
- **ActivityMath:** this controls the availability of the Activity Math tab on the Intake File Search screen. The Activity Math tab displays all of the math variables for the Intake File transaction once the activity has moved to Active status.
- **BusinessError:** this controls the availability of the Business Error tab on the Intake File Search screen. The Business Error tab appears when a business error is incurred during Intake File-level transaction processing, and displays an error table displaying the details of each error.
- **SystemError:** this controls the availability of the System Error tab on the Intake File Search screen. The System Error tab appears when a system error is incurred during Intake File-level transaction processing, and displays an error table displaying the details of each error.
- **IntakeProfile:** this page controls the buttons available for selection on the Intake Profile screen.
 - **Activate:** this controls the availability of the Activate button on the Intake Profile screen, which allows users to move Intake Profiles to "Active" status.
 - **AddActivity:** this controls the availability of the Add Activity button in the Secondary Menu of the Intake Profile screen.
 - **AddIntakeProfile:** this controls the availability of the Add Intake Profile drop-down on the Intake Profile screen.
 - **Deactivate:** this controls the availability of the Deactivate button on the Intake Profile screen, which allows users to move Intake Profiles to "Inactive" status.

- **Delete:** this controls the availability of the Delete icon in the Action column on the Intake Profile screen.
- **Inquiry:** this controls the availability of the Inquiry button in the Secondary Menu of the Intake Profile screen.
- **Save:** this controls the availability of the Save button when creating a new Intake Profile.
- **IntakeRecordSearch:** this page controls the buttons and actions available for selection on the Intake Record Search screen.
 - **AddActivity:** this controls the availability of the Add Activity button in the Secondary Menu of the Intake Record Search screen.
 - **Inquiry:** this controls the availability of the Inquiry button in the Secondary Menu of the Intake Record Search screen.
 - **ActivityMath:** this controls the availability of the Activity Math tab on the Intake Record Search screen. The Activity Math tab displays all of the math variables for the Intake File transaction once the activity has moved to Active status.
 - **ActivitySequence:** this controls the availability of the Activity Sequence tab on the Intake Record Search screen. The Activity Sequence tab displays a hierarchical "tree" of activities generated by the Intake Record transaction for the selected Intake Record.
 - **BusinessError:** this controls the availability of the Business Error tab on the Intake Record Search screen. The Business Error tab appears when a business error is incurred during Intake Record-level transaction processing, and displays an error table displaying the details of each error.
 - **SystemError:** this controls the availability of the System Error tab on the Intake Record Search screen. The System Error tab appears when a system error is incurred during Intake Record-level transaction processing, and displays an error table displaying the details of each error.
 - **RecordXML:** this controls the availability of the Record XML tab on the Intake Record Search screen. The Record XML tab displays the received Intake Record XML configuration for the selected Intake Record.

- **Inquiry:** this page controls the availability of the Inquiry option on the Main Menu in OIPA. There are no associated buttons. If this checkbox is left empty, then the Inquiry option will not appear in the Main Menu.
- **NetAssetValue:** this page controls whether the Unit Value table view is visible to the OIPA user and the buttons that display when the Unit Values link is clicked from the Tables Main Menu in OIPA. This page is hard coded and cannot be configured. This security setting just controls whether or not it is visible to the OIPA user.
 - **Find:** this controls whether the Find button displays on the Unit Values screen.
- **Open:** this page controls the ability to launch Policy-, Case- and Client-related screens directly from external systems, such as Oracle Insurance Data Capture (OIDC). There are no buttons or other actions associated with this page.
- **Phone:** this page controls the Phone screen. The Phone screen is accessed from the Phone button on the Customer Address screen.
 - **New:** this controls whether the New link displays on the Secondary menu of the Phone screen.
 - **Save:** this controls whether the Save link displays on the Secondary menu of the Phone screen.
 - **Delete:** this controls whether the Delete link displays on the Secondary menu of the Phone screen.
- **Plan:** this page controls the Plan screen. The Plan screen is accessed from the Plans tab at the bottom of the Agreement screen.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Plan Activity screen is visible in OIPA.
 - **EditPlan:** this controls the **Edit** right-click option on the Agreement screen. This option allows the user to edit an agreement's details.
 - **GoToAgreement:** this controls the **Go to Agreement** right-click option on the Agreement screen. This option allows the user to view an agreement's details.
 - **Inquiry:** this controls whether the **Inquiry** drop-down displays on the Secondary menu of the Plan screen.

- **NewPlan:** this controls whether the **New** link displays on the Secondary menu of the Plan screen.
- **OpenClassRuleVariables:** this controls whether the Class Rule Variables link displays on the Secondary menu of the Plan screen.
- **PlanDetails:** this controls whether the Details tab is visible at the bottom of the Plan screen.
- **Save:** this controls whether the Save button displays at the bottom of the Plan screen.



Plan Screen With Security Applied to Links

- **PlanActivity:** this page controls the buttons that display when the Plan Activity link is clicked from the Plan Main Menu in OIPA.
 - **ActivityComments:** this controls the display of the Activity Comments icon to the right of an activity on the Plan Activity page.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Plan Activity screen is visible in OIPA.
 - **AllowDelete:** this controls whether the Delete checkbox will display on the Plan Activity screen.
 - **AutoProcess:** this controls whether the AutoProcess checkbox will display on the Plan Activity screen.

- **ProcessAll:** this controls whether the Process All button will display on the Plan Activity screen.
- **PlanAllocation:** this page controls whether or not the Plan Allocation table view is visible to the OIPA user.
- **Plan Coverage:** this page controls whether the Plan Coverage tab displays at the bottom of the Plan screen. Access the plan screen from the Agreement screen. A Plans tab is at the bottom. Click a plan and select GoToPlan. This will open the Plan screen and Plan Coverages will be available..
 - **GoToPlanCoverages:** this controls the right-click option for PlanCoverages when a user clicks on a plan coverage from the Plan screen. A Plan Coverages tab is located at the bottom of the Plan screen. Clicking the tab shows a list of all plan coverages. Right-click and select GoToPlanCoverage to see the details.
- **PlanFields:** this page controls the buttons that display when the **Plan Fields** link is clicked from the Plan Main Menu in OIPA.
 - **Save:** this controls the availability of the **Save** button on the Plan Fields page.
-
- **PlanSegmentName:** this Company Page does not currently contain any configurable button security.
- **PlanSegmentNameClass:** this Company Page controls security for the buttons on the Class Sub-Plans screen.
 - AssociateSubPlan: this controls whether the Attach Sub-Plans button can be selected on the Class Sub-Plans screen.
 - PlanSegmentNameClassDetails: this controls users' ability to select the Plan Coverage Details tab on the Class Sub-Plans screen.
 - RemoveSubPlan: this controls users' ability to terminate an association between a Sub-Plan and a Class by right-clicking on a Sub-Plan record and selecting Remove Association.
 - Save: this controls users' ability to save an association between a Sub-Plan and a Class.
- **PlanSegmentNameClassParticipant:** this Company Page controls

security for the Class Plan Participants tab of the Class Sub-Plans screen.

- **PlanSegmentNameClassParticipantDetail:** this controls users' ability to view Participant details on the Class Plan Participants tab.
- **PlanWithholding:** this page controls whether or not the Plan Withholding table view is visible to the OIPA user.
- **PolicySearch:** this page controls the buttons that display when the Search link is clicked from the Policy Main Menu in OIPA. [Masks](#) are supported on this screen.
- **Rates:** this page controls whether or not the Rates table view is visible to the OIPA user and the buttons that display when the Rates link is clicked from the Table Main Menu in OIPA. The Find and Filter buttons are controlled from this company page.
 - **Filter:** this controls whether the Filter button will display in the Rates Criteria section of the Rates screen.
 - **Find:** this controls whether the Find button displays in the Rates Search section of the Rates screen.
- **StateApprovals:** this page controls whether or not the State Approval table is available as an option on the Table menu in OIPA.
- **SuspenseComments:** this page controls a user's ability to add, delete and view suspense comment information.

There is a **Comment Buttons** configuration section, which controls actions related to suspense comments. The following comment button security is available for each action:

- **Add:** this controls whether suspense comments are able to be added.
- **View:** this controls whether suspense comments are able to be viewed, or what type of suspense comments (i.e. all comments or only those made by the current user) are able to be viewed. If **Add** is set to **Yes**, then No is not an option for this action.
- **Update:** this controls whether suspense comments are able to be modified, or what type of suspense comments (i.e. all comments or only those made by the current user) are able to be modified. If UpdateAllComments is selected, then ViewAllComments must also be selected.

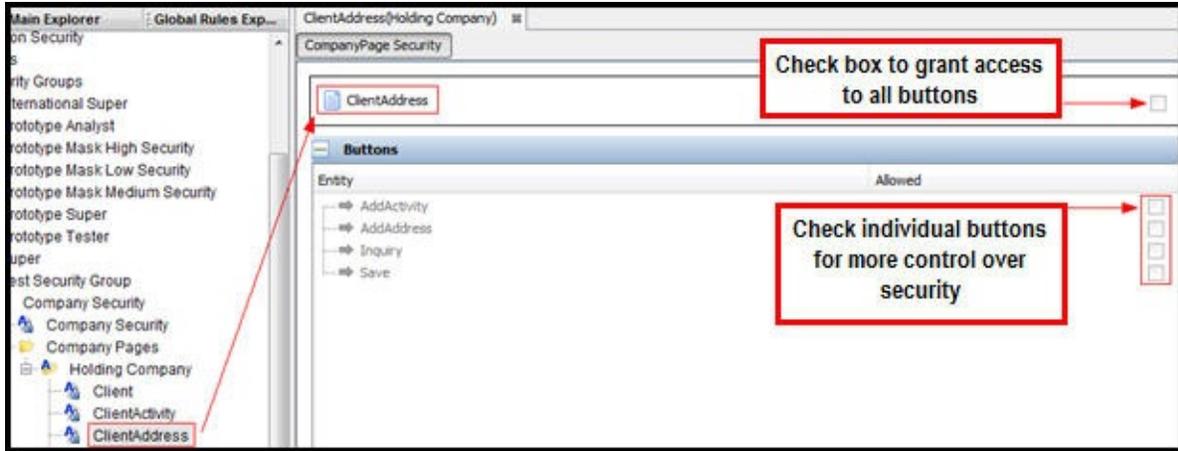
- **Delete:** this controls whether suspense comments are able to be deleted, or what type of suspense comments (i.e. all comments or only those made by the current user) are able to be deleted. If DeleteAllComments is selected, then ViewAllComments must also be selected.
 - **View History:** this controls whether suspense comment history is accessible, or what type of suspense comment (i.e. all comments or only those made by the current user) history is able to be viewed. If ViewAllCommentsHistory is selected, then ViewAllComments must also be selected.
- **SuspenseCommentsSearch:** this page controls the display of the Comments link on the Left Navigation menu of the Suspense screen. When the link is clicked, it allows you to search for suspense comments.
 - **SuspenseComments:** this controls whether the Add Comments button will display in the Secondary Menu when the Suspense screen is open.
 - **Refresh:** this controls whether the Refresh button will display on the client's Comments Search screen.
 - **Shadow:** this controls whether the Shadow checkbox will be enabled on the client's Comments Search screen. This checkbox is used to specify if shadowed comments will be returned in the search results.
- **Suspense History:** this page controls whether the Suspense History link displays on the Left Navigation menu when a Suspense record is open in OIPA.
 - **SuspenseComments:** this controls whether the Add Comments button will display in the Secondary Menu when the Suspense History screen is open. You will not be able to actually enter information for a comment until the [ClientComments](#) Company Page privilege is also added.
- **SuspenseRecord:** this page controls the buttons that display when the Suspense Record link is clicked from the Suspense Main Menu in OIPA.
 - **Delete:** this controls whether the Delete button displays next to the Save and Find buttons at the bottom of the Suspense Records screen. The Delete button only displays after a user adds a suspense record or searches for a suspense record.

- **Find:** this controls whether the Find button displays at the bottom of the Suspense Record screen.
- **New:** this button is not supported in OIPA. New suspense records are just added directly on the Suspense Record screen.
- **Save:** this controls whether the Save button displays at the bottom of the Suspense Records screen.
- **SuspenseComments:** this controls the display of the AddComment link on the Secondary menu of suspense record. You will not be able to actually enter information for a comment until the [ClientComments](#) Company Page privilege is also added.
- **SuspenseSearch:** this page controls the availability of the Search drop down option on the Suspense Main Menu. There are no associated buttons. If this checkbox is left empty, then the Search option will not appear in the Suspense drop down list on the Main Menu. [Masks](#) are supported on this screen.
- **UnmatchedResults:** this page controls the buttons available on the Unmatched Result Search screen.
 - **Match:** this controls whether the button used to match unmatched requirement results to requirements is available for selection on the screen.

Explanation of a Company Page

To open a Company Page, right-click on the page and select **Check-out**. There are four sections that display in the Configuration Area.

1. **Page level security**: grant access to all buttons by clicking the checkbox at the top of the Configuration Area to the right of the page name.
2. **Button security**: grant access to individual buttons on the page by clicking the checkbox to the right of a button. Buttons that are unchecked will not be visible to the user in OIPA.
 - The **AllowDelete** button on a ClientActivity, CompanyActivity or PlanActivity Company page grants a user access to the Delete checkbox on the Activity screen. This checkbox allows [Non-reversible/Nonreversing activities](#) and spawns to be deleted or recycled.
3. **Field security**: grant access to individual fields. There are three options for field security:
 - Visible and Editable: the field is both enabled and the value held in the field is visible. This is the default setting and when selected no database entry will be made.
 - Hide field value: the field is disabled, but the value held in the field is hidden.
 - Disable field value: the field is disabled and any values are visible.
4. **Masking**: Add security to the mask applied to a text field. Security levels are defined in `AsCodeMaskSecurityLevel`.



Company Page Open in Configuration Pane

You are here: Security > Application Security > Company Security > Web Service Security

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Web Service Security

The Rules Palette supports versioning of Web Services records, the [Release Management](#) process for the Web Services records and security privileges for viewing and editing Web Services entries. Web Services security is managed through the Admin Explorer under **Security | Application Security | Security Groups | *Name of the Security Group* | Company Security**. Open the Web Services folder and check-out the Company file listed there. This will launch the Web Services security editor.

Each record from the AsAuthWebService table has a check box in the editor. If a checkbox is selected, then the users of the security group can access the Web Service. Users can be granted access to any or all available Web Services.

When the security group is checked in, AsAuthCompanyWebService records are created or removed depending on the value of the security check box and the changes made during the session. Each node with a security check box checked will have a record in AsAuthCompanyWebService that links AsAuthCompany to the unique Web Service record.

Versioning

Each Web Service name is versioned when the file is checked-in. The version's field XML clob contains the version guids of each AsAuthWebService record belonging to the Web Service name. The Web Service name's version is not incremented unless at least one of the records changes; meaning a record is modified, added and/or deleted. Each version guid in the XML clob contains the column data in individual IVSVersionField records. Versions are not incremented unless the AsAuthWebService record changes.

Examples of Versioning

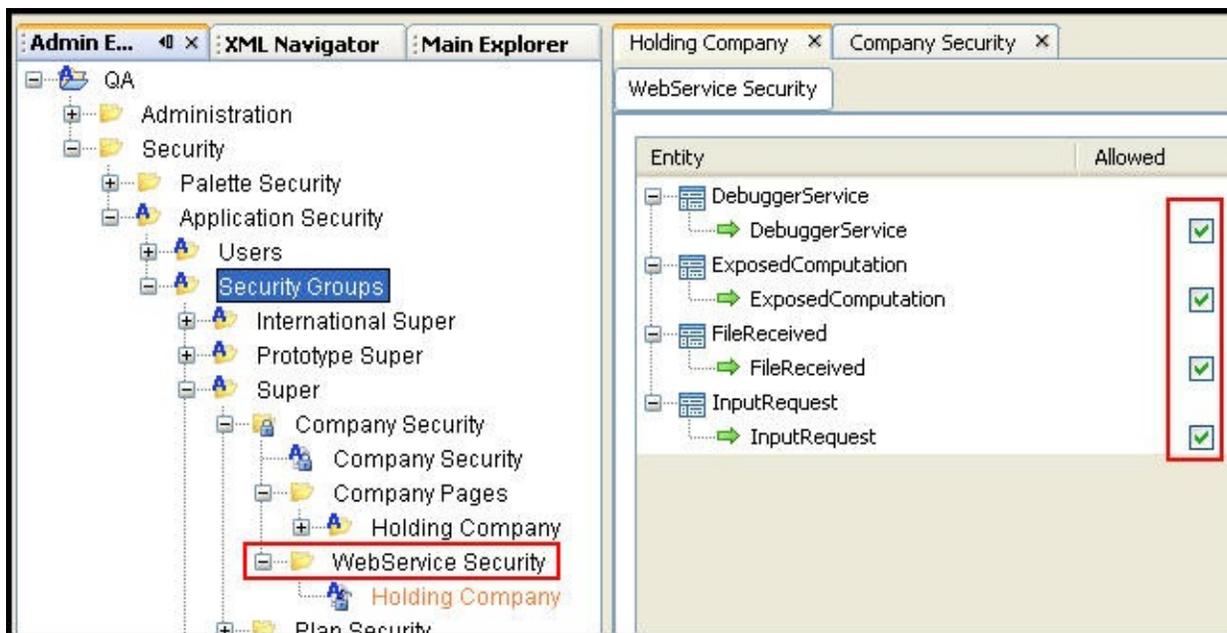
ServiceName1 exists as version 3. It contains three AsAuthWebService record versions with the versions 1,3 and 2. ServiceName1 may be checked out and the version will not be incremented if, upon check in, none of the AsAuthWebService records are changed.

ServiceName1 may be checked-out and the third record (with version 2) is updated. Upon check-in, ServiceName1 version is incremented to 4 and contains versions of AsAuthWebService records of 1, 3 and 3.

ServiceName1 may be checked-out and the first record (with version 1) is deleted. In addition, a new AsAuthWebService record is added to the Web Service name. When checked-in, ServiceName1 version is incremented to 5 and contains versions 3, 3, and 1.

Steps to Grant a User Access to a Web Service

1. Navigate to the Admin Explorer and open **Security | Application Security | Security Groups | Name of the Security Group | Company Security | WebService Security**.
2. Right-click on the Company name listed under the WebService Security folder and select **Check-out**. A list of all Web Service Methods and File ID combinations will display.
3. Click the checkbox next to each Web Service, Web Service Method and File ID combination the user should be able to access.
4. Right-click on the Company name listed under the WebService Security folder and select **Check-in**. The security settings will be saved.



Web Service Security in Admin Explorer

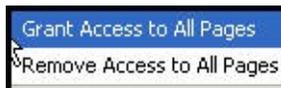
You are here: Security > Application Security > Plan Security

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Plan Security

Multiple plans may exist for a company. Plan security allows security levels for individual plans to be assigned on a plan by plan basis. The Plan Security folder holds a folder for the Primary company and any Subsidiary companies. Each company folder contains the plans associated with that company.

Security can be added to all pages in the plan by right-clicking on the plan name. Security can also be assigned on a page by page basis by opening the Plan Pages folder and selecting a specific plan page.



Plan Pages Right-Click Options



Security is applied from the top down. Once Primary Company security is defined in the Company file and the Company pages, then the Plan Security folder will populate with available plans. Once Plan security is defined, then the Transaction Security folder will populate with available transactions.

Plan Pages

The following is a list of all Plan pages, along with an explanation of what the page controls in OIPA.



The Add Requirement and Add Impairment checkboxes will only display if New Business Underwriting functionality is enabled.

- **ActivityComments:** this controls the buttons available on an activity's Comments screen, accessed by clicking the Comments icon next to an activity on a policy's Activity screen.
 - **New:** this allows the user to create a new activity comment.
 - **Save:** this controls whether the Save link will display on the Secondary menu when an activity's Comments screen is visible in OIPA.
 - **Cancel:** this allows the user to cancel the activity comment currently being added.

Plan Pages relating to comment screens (but not comment search screens) also contain a **Comment Buttons** configuration section, which controls actions related to comments. The following comment button security is available for this page:

- **Add:** this controls whether activity comments are able to be added.
- **View:** this controls whether activity comments are able to be viewed, or what type of activity comments (i.e. all comments or only those made by the current user) are able to be viewed.
- **Update:** this controls whether activity comments are able to be modified, or what type of activity comments (i.e. all comments or only those made by the current user) are able to be modified.
- **Delete:** this controls whether activity comments are able to be deleted, or what type of activity comments (i.e. all comments or only those made by the current user) are able to be deleted.
- **View History:** this controls whether an activity's comment history is

accessible, or what type of activity comment (i.e. all comments or only those made by the current user) history is able to be viewed.

- **ActivityDetail:** this is the window that opens when the Add Activity link is clicked from the Secondary menu. If this page is unchecked, the Add Activity link will not be available on the Secondary menu.
 - **ActivityAddress:** this controls the Address link inside the Activity Details window that opens when the Add Activity link is clicked on the Secondary menu. If this is unchecked, an Address link will not display.
 - **ActivityAllocations:** this controls the Allocation link inside the Add Activity window that opens when the Add Activity link is clicked on the Secondary menu. If this is unchecked, an Allocation link will not display.
 - **ActivitySuspense:** this controls the Suspense link inside the Add Activity window that opens when the Add Activity link is clicked on the Secondary menu. If this is unchecked, a Suspense link will not display.
 - **ActivityWithholding:** this controls the Withholding link inside the Add Activity window that opens when the Add Activity link is clicked on the Secondary menu. If this is unchecked, a Withholding link will not display.
- **ActivityDocument:** this controls the document formats that are available for the user to select when working with activities that generate documents.
- **ActivityResult:** this controls the links that display when the Activity Detail icon is clicked to the left of a processed activity.
 - **ActivityAddress:** this controls whether the Address link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.
 - **ActivityAccounting:** this controls whether the Accounting link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.
 - **ActivityAllocations:** this controls whether the Allocations link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.
 - **ActivityClassMembership:** this controls whether the Class

Membership link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.



Class Membership Link in Activity Results When Security is Applied

- **ActivityDetail:** this controls whether the Detail link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.
- **ActivityDisbursement:** this controls whether the Disbursement link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.
- **ActivityDocument:** this controls whether the Document link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.
- **ActivityEntryFields:** this controls whether the Entry Fields link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.
- **ActivityMath:** this controls whether the Math link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.
- **ActivitySpawn:** this controls whether the Spawn link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.
- **ActivitySummary:** this controls whether the Summary link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.
- **ActivitySuspense:** this controls whether the Suspense link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.
- **ActivityValuation:** this controls whether the Valuation link will display in the Activity Results window when the Activity Detail icon is clicked to the left of a processed activity.
- **AllocationHistory:** this page controls the buttons that display when the

History sub node is clicked from the Allocation link on the Policy screen Left Navigation menu.

- **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu of the Allocation History screen.
- **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu of the Allocation History screen.
- **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Allocation History screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.
- **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu of the Allocation History screen.
- **AddImpairment:** This controls whether the Add Impairment link will display on the Secondary menu of the Allocation History screen.
- **ExternalClient:** this page controls the buttons that display when the External Client Details screen displays.
 - **Save:** this controls whether the Save link will display on the Secondary menu when the External Client Detail screen is visible in OIPA.



Refer to the Extensibility document on the [Oracle Technology Network](#) for 10.1.2.0 for information on configuring external clients.

- **ImpairmentComments:** this page controls a user's ability to add, delete and view Impairment comment information.

Plan Pages relating to comment screens (but not comment search screens) also contain a **Comment Buttons** configuration section, which controls actions related to comments. The following comment button security is available for this page:

- **Add:** this controls whether Impairment comments are able to be added.
- **View:** this controls whether Impairment comments are able to be

viewed, or what type of comments (i.e. all comments or only those made by the current user) are able to be viewed. If **Add** is set to **Yes**, then No is not an option for this action.

- **Update:** this controls whether Impairment comments are able to be modified, or what type of comments (i.e. all comments or only those made by the current user) are able to be modified. If UpdateAllComments is selected, then ViewAllComments must also be selected.
- **Delete:** this controls whether Impairment comments are able to be deleted, or what type of comments (i.e. all comments or only those made by the current user) are able to be deleted. If DeleteAllComments is selected, then ViewAllComments must also be selected.
- **View History:** this controls whether an Impairment's comment history is accessible, or what type of comment (i.e. all comments or only those made by the current user) history is able to be viewed. If ViewAllCommentsHistory is selected, then ViewAllComments must also be selected.
- **ImpairmentDetail:** this page contains no buttons that require security configuration.
- **Policy:** this page controls the buttons that displays when the Policy screen is visible in OIPA.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Policy screen is visible in OIPA.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Policy screen is visible in OIPA.
 - **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Policy screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.
 - **Save:** this controls whether the Save link will display on the Secondary menu when the Policy screen is visible in OIPA.
 - **ShadowPolicy:** this controls whether the Shadow Policy link will display on the Secondary menu when the Policy screen is visible in OIPA.

- **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu of the Policy screen.
- **AddImpairment:** This controls whether the Add Impairment link will display on the Secondary menu of the Policy screen.
- **PolicyActivity:** this page controls the buttons that display when the Left Navigation menu Activity link is clicked from the Policy screen in OIPA.
 - **ActivityComments:** this controls the display of the comment icon to the right of an activity listed on the Policy Activity screen.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu on the Policy Activity screen.
 - **AllowDelete:** this controls whether the Delete checkbox will display on the Policy Activity screen.
 - **AutoProcess:** this controls whether the AutoProcess checkbox will display on the Policy Activity screen.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu on the Policy Activity screen.
 - **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Policy Activity screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.
 - **ProcessAll:** this controls whether the Process All button will display on the Policy Activity screen.
 - **Save:** this controls whether the Save button will display on the Secondary menu on the Policy Activity screen.
 - **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu on the Policy Activity screen.
 - **AddImpairment:** this controls whether the Add Impairment link will display on the Secondary menu on the Policy Activity screen.
- **PolicyAllocation:** this page controls the buttons that display when the Left Navigation menu Allocation link is clicked from the Policy screen in OIPA.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Allocation link is clicked in the Policy Left Navigation menu.
 - **Inquiry:** this controls whether the Inquiry link will display on the

Secondary menu when the Allocation link is clicked in the Policy Left Navigation menu.

- **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Allocation screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.
- **Save:** this controls whether the Save link will display on the Secondary menu when the Allocation link is clicked in the Policy Left Navigation menu.
- **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu when the Allocation link is clicked in the Policy Left Navigation menu.
- **AddImpairment:** this controls whether the Add Impairment link will display on the Secondary menu when the Allocation link is clicked in the Policy Left Navigation menu.
- **PolicyAssessment:** this page controls the buttons that display on the Assessment screen.
 - **AddActivity:** this controls whether the Add Activity link will display in the Secondary menu on the Assessment screen.
 - **AddImpairment:** this controls whether the Add Impairment link will display in the Secondary menu on the Assessment screen.
 - **AddRequirement:** this controls whether the Add Requirement link will display in the Secondary menu on the Assessment screen.
 - **AllowImpairmentDelete:** this controls whether the trash can icon will display for impairments on the Assessment screen, as well as whether the "Show Shadows" checkbox will display on the Assessment screen.
 - **ImpairmentComments:** this controls whether the comment button will display for impairment records on the Assessment screen.
 - **ImpairmentDetail:** this controls whether the Impairment Detail icon will display for impairment records displayed on the Assessment screen.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu on the Assessment screen.
 - **PolicyComments:** this controls whether the AddComment link will

display in the Secondary menu on the Assessment screen.

- **PolicyComments:** this page controls a user's ability to add, delete and view policy comment information.

There is a **Comment Buttons** configuration section, which controls actions related to comments. The following comment button security is available for each action:

- **Add:** this controls whether policy comments are able to be added.
- **View:** this controls whether policy comments are able to be viewed, or what type of comments (i.e. all comments or only those made by the current user) are able to be viewed. If **Add** is set to **Yes**, then No is not an option for this action.
- **Update:** this controls whether policy comments are able to be modified, or what type of comments (i.e. all comments or only those made by the current user) are able to be modified. If UpdateAllComments is selected, then ViewAllComments must also be selected.
- **Delete:** this controls whether policy comments are able to be deleted, or what type of comments (i.e. all comments or only those made by the current user) are able to be deleted. If DeleteAllComments is selected, then ViewAllComments must also be selected.
- **View History:** this controls whether a policy's comment history is accessible, or what type of comment (i.e. all comments or only those made by the current user) history is able to be viewed. If ViewAllCommentsHistory is selected, then ViewAllComments must also be selected.
- **PolicyCommentsSearch:** this page controls the display of the Comments link on the Left Navigation menu of the Policy screen. When the link is clicked, it allows you to search for policy comments.
 - **Add Activity:** this controls whether the Add Activity link will display on the Secondary menu when the Policy Overview link is clicked in the Policy Left Navigation menu.
 - **PolicyComments:** this checkbox controls the display of the

AddComment link on the Secondary menu of the Policy Comment Search screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.

- **Refresh:** this controls whether the Refresh button will display on the Policy Comment Search screen.
- **Shadow:** this controls whether the Shadow checkbox will be enabled on the Policy Comments Search screen. This checkbox is used to specify if shadowed comments will be returned in the search results.
- **PolicyOverview:** this page controls the buttons that display when the Left Navigation menu Policy Overview link is clicked from the Policy screen in OIPA.
 - **Add Activity:** this controls whether the Add Activity link will display on the Secondary menu when the Policy Overview link is clicked in the Policy Left Navigation menu.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Policy Overview link is clicked in the Policy Left Navigation menu.
 - **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Policy Overview screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.
 - **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu when the Policy Overview link is clicked in the Policy Left Navigation menu.
 - **AddImpairment:** This controls whether the Add Impairment link will display on the Secondary menu when the Policy Overview link is clicked in the Policy Left Navigation menu.
- **PolicyRequirement:** this page controls the buttons that display on the Policy Requirement screen.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu on the Policy Requirement screen.
 - **AddImpairment:** this controls whether the Add Impairment link will display on the Secondary menu on the Policy Requirement screen.
 - **AddRequirement:** this controls whether the Add Requirement link

- will display on the Secondary menu on the Policy Requirement screen.
- **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu on the Policy Requirement screen.
 - **PolicyComments:** this controls whether the Add Comment link will display on the Secondary menu on the Policy Requirement screen.
 - **RequirementComments:** this controls whether the comment button will display for each requirement record in the Requirements table on the Policy Requirement screen.
 - **RequirementDetail:** this controls whether the Requirement Detail icon will display for Requirement records displayed on the Policy Requirement screen.
 - **RequirementResults:** this controls whether the Requirement Results button will display on the Policy Requirement screen.
 - **RequirementResultsSearch:** this controls whether the Requirement Results Search button will display on the Policy Requirement screen.
- **PolicyRole:** this page controls the buttons that display when the Left Navigation menu Role link is clicked from the Policy screen in OIPA.
- **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Role link is clicked in the Policy Left Navigation menu.
 - **AlsoAsRole:** this allows the Also As option to appear in the right-click menu when a user right-clicks a policy role on the Role screen.
 - **DeleteRole:** this allows the Delete Role option to appear in the right-click menu when a user right-clicks a policy role on the Role screen.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Role link is clicked in the Policy Left Navigation menu.
 - **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Role screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.
 - **RoleAddClient:** this controls whether the Add button will display at the bottom of the Policy Role screen.
 - **RoleDetail:** this controls whether the Role Detail tab will display with

the other tabs in the middle of the Policy Role screen.

- **RoleFindClient:** this controls whether the Find button will display in the middle of the Policy Role screen.
- **Save:** this controls whether the Save link will display on the Secondary menu when the Role link is clicked from the Left Navigation menu on the Policy screen.
- **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu on the Policy Role screen.
- **AddImpairment:** this controls whether the Add Requirement link will display on the Secondary menu on the Policy Role screen.
- **PolicySegment:** this page controls the buttons that display when the Left Navigation menu Segment link is clicked from the Policy screen in OIPA.
 - **Add Activity:** this controls whether the Add Activity link will display on the Secondary menu when the Segment link is clicked in the Policy Left Navigation menu.
 - **AddSegment:** this controls whether the Add Segment link will display on the Secondary menu when the Segment link is clicked in the Policy Left Navigation menu.
 - **Calculate:** this controls whether the Calculate button will display on the Segment screen.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Segment link is clicked in the Policy Left Navigation menu.
 - **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Segment screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.
 - **Save:** this controls whether the Save link will display on the Secondary menu when the Segment link is clicked from the Left Navigation menu on the Policy screen.
 - **SegmentComments:** this controls the display of the comment icon to the right of a segment on the Segment screen. You must have at least View privileges assigned in the Comment Button section of the [SegmentComments](#) Plan Page or the comment icon will not appear.

- **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu when the Segment link is clicked from the Left Navigation menu on the Policy screen.
- **AddImpairment:** this controls whether the Add Impairment link will display on the Secondary menu when the Segment link is clicked from the Left Navigation menu on the Policy screen.
- **PolicyValue:** this page controls the buttons that display when the Left Navigation menu Value link is clicked from the Policy screen in OIPA.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Value link is clicked in the Policy Left Navigation menu.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Value link is clicked in the Policy Left Navigation menu.
 - **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Value screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.
 - **Save:** this controls whether the Save link will display on the Secondary menu when the Value link is clicked in the Policy Left Navigation menu.
 - **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu when the Value link is clicked from the Left Navigation menu on the Policy screen.
 - **AddImpairment:** this controls whether the Add Requirement link will display on the Secondary menu when the Value link is clicked from the Left Navigation menu on the Policy screen.
- **PolicyWithholding:** this page controls the buttons that display when the Left Navigation menu Withholding link is clicked from the Policy screen in OIPA.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Withholding link is clicked in the Policy Left Navigation menu.
 - **Inquiry:** this controls whether the Inquiry link will display on the

Secondary menu when the Withholding link is clicked in the Policy Left Navigation menu.

- **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Withholding screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.
- **Save:** this controls whether the Save link will display on the Secondary menu when the Withholding link is clicked in the Policy Left Navigation menu.
- **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu when the Withholding link is clicked from the Left Navigation menu on the Policy screen.
- **AddImpairment:** this controls whether the Add Requirement link will display on the Secondary menu when the Withholding link is clicked from the Left Navigation menu on the Policy screen.
- **Program:** this controls the action icons on the Program screen.
 - **AddProgram:** this controls whether the Add button is available. If the program is attached to a policy or segment that is in a status not defined in the <PolicyProgram> or <SegmentProgram> rule, then this button may not be visible even when security is applied.
 - **AuditHistory:** this controls whether the Audit History tab is available for a user at the bottom of the Program screen.
 - **Change:** this controls whether the **Change** icon is available to the right of a program listed on the Program screen.
 - **ChangePending:** this controls whether the **Change Pending** tab is available on the Program screen.
 - **Delete:** this controls whether the trash can icon is available to the right of a program listed in the Views section of the Program screen.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Program link is clicked in the Policy Left Navigation menu.
 - **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Program screen. You will not be able to actually enter information for a comment until the

[PolicyComments](#) Plan Page privilege is also added.

- **Save:** this controls whether the Save button is available. If the program is attached to a policy or segment that is in a status not defined in the <PolicyProgram> or <SegmentProgram> rule, then this button may not be visible even when security is applied.
- **Start:** this controls whether the green start icon is available next the trash can icon for a program in the View section of the Program screen.
- **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu on the Program screen.
- **AddImpairment:** this controls whether the Add Requirement link will display on the Secondary menu on the Program screen.
- **RequirementComments:** this page controls a user's ability to add, delete and view requirement comment information.

Plan Pages relating to comment screens (but not comment search screens) also contain a **Comment Buttons** configuration section, which controls actions related to comments. The following comment button security is available for this page:

- **Add:** this controls whether requirement comments are able to be added.
- **View:** this controls whether requirement comments are able to be viewed, or what type of comments (i.e. all comments or only those made by the current user) are able to be viewed. If **Add** is set to **Yes**, then No is not an option for this action.
- **Update:** this controls whether requirement comments are able to be modified, or what type of comments (i.e. all comments or only those made by the current user) are able to be modified. If UpdateAllComments is selected, then ViewAllComments must also be selected.
- **Delete:** this controls whether requirement comments are able to be deleted, or what type of comments (i.e. all comments or only those made by the current user) are able to be deleted. If DeleteAllComments is selected, then ViewAllComments must also be selected.

- **View History:** this controls whether a requirement's comment history is accessible, or what type of comment (i.e. all comments or only those made by the current user) history is able to be viewed. If ViewAllCommentsHistory is selected, then ViewAllComments must also be selected.
- **RequirementHistory:** this page controls the buttons available on the Requirement History screen.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu on the Requirement History screen.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu on the Requirement History screen.
 - **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu on the Requirement History screen.
 - **AddImpairment:** this controls whether the Add Impairment link will display on the Secondary menu on the Requirement History screen.
 - **PolicyComments:** this controls whether the AddComment link will display on the Secondary menu on the Requirement History screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.
- **RoleHistory:** this page controls whether the History sub menu is visible under the Role Left Navigation menu on the Policy screen in OIPA. It also controls the buttons that display in the Secondary menu when the Role link is clicked and the Role screen displays.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the History link is clicked under Roles in the Policy Left Navigation menu.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the History link is clicked under Roles in the Policy Left Navigation menu.
 - **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Role History screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.
 - **AddRequirement:** this controls whether the Add Requirement link

will display on the Secondary menu on the Role History screen.

- **AddImpairment:** this controls whether the Add Requirement link will display on the Secondary menu on the Role History screen.
- **SegmentComments:** this page controls a user's ability to add, delete and view policy comment information.
 - **New:** this allows the user to create a new segment comment.
 - **Save:** this controls whether the Save link will display on the Secondary menu when a segment's Comments screen is visible in OIPA.
 - **Cancel:** this allows the user to cancel the segment comment currently being added.

Plan Pages relating to comment screens (but not comment search screens) also contain a **Comment Buttons** configuration section, which controls actions related to comments. The following comment button security is available for this page:

- **Add:** this controls whether segment comments are able to be added.
- **View:** this controls whether segment comments are able to be viewed, or what type of segment comments (i.e. all comments or only those made by the current user) are able to be viewed.
- **Update:** this controls whether segment comments are able to be modified, or what type of segment comments (i.e. all comments or only those made by the current user) are able to be modified.
- **Delete:** this controls whether segment comments are able to be deleted, or what type of segment comments (i.e. all comments or only those made by the current user) are able to be deleted.
- **View History:** this controls whether a segment's comment history is accessible, or what type of segment comment (i.e. all comments or only those made by the current user) history is able to be viewed.
- **SegmentRole:** this page controls whether the Role sub menu is visible under the Segment Left Navigation menu on the Policy screen in OIPA. It also controls the buttons that display in the Secondary menu when the Role link is clicked and the Segment Role screen displays.
 - **AddActivity:** this controls whether the Add Activity link will display

on the Secondary menu.

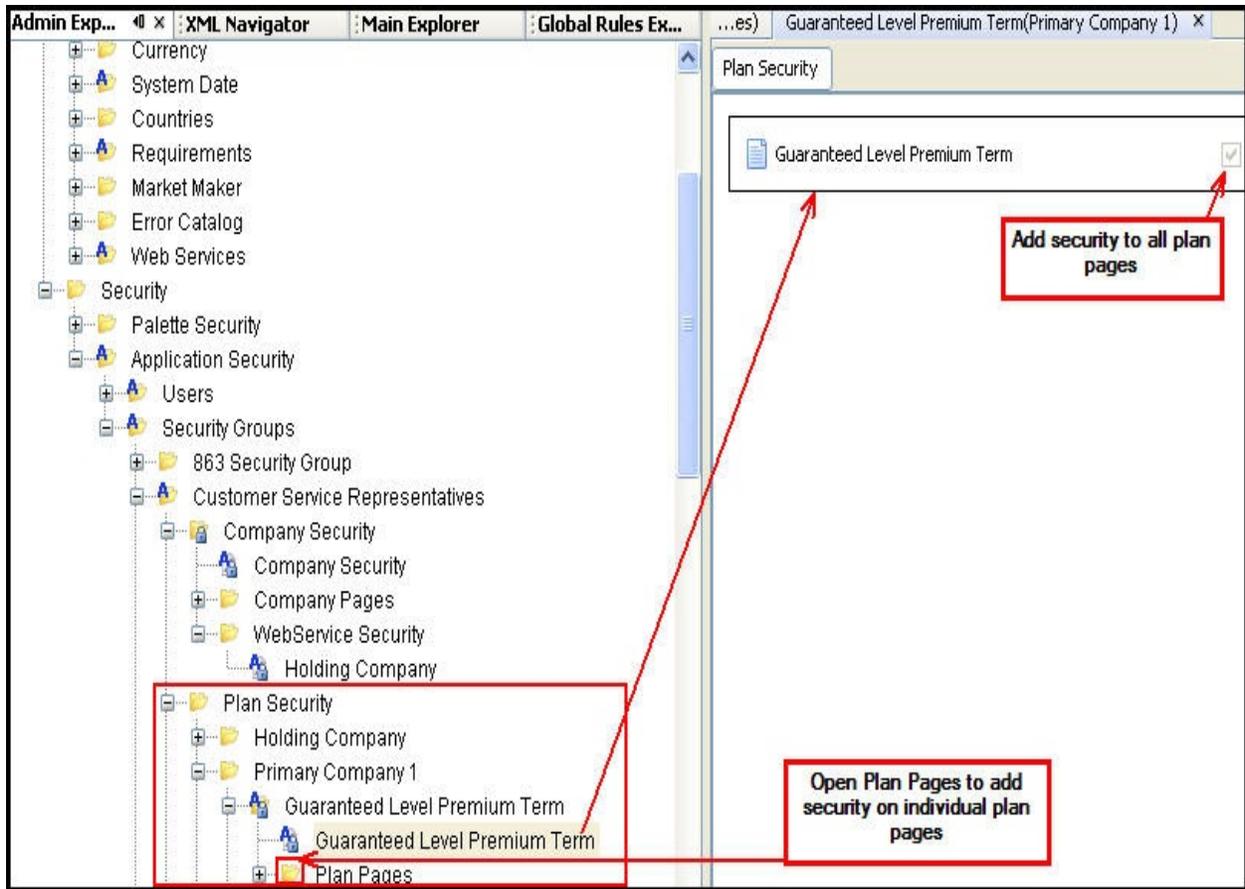
- **DeleteRole:** this allows the Delete Role option to appear in the right-click menu when a user right-clicks a segment role.
- **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu.
- **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Segment Role screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.
- **RoleAddClient:** this controls whether the Add button will display at the bottom of the Segment Role screen.
- **RoleDetail:** this controls whether the Role Detail tab will display with the other tabs in the middle of the Segment Role screen.
- **RoleFindClient:** this controls whether the Find button will display in the middle of the Segment Role screen.
- **Save:** this controls whether the Save link will display on the Secondary menu.
- **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu on the Segment Role screen.
- **AddImpairment:** this controls whether the Add Requirement link will display on the Secondary menu on the Segment Role screen.
- **SegmentRoleHistory:** this page controls whether the History sub menu is visible under the Segment Left Navigation menu on the Policy screen when the Role sub node is selected.
 - **AddActivity:** this controls whether the Add Activity link will display on the Secondary menu when the Segment Role History link is clicked under Segment in the Policy Left Navigation menu.
 - **Inquiry:** this controls whether the Inquiry link will display on the Secondary menu when the Segment Role History link is clicked under Segment in the Policy Left Navigation menu.
 - **PolicyComments:** this checkbox controls the display of the AddComment link on the Secondary menu of the Segment Role History screen. You will not be able to actually enter information for a comment until the [PolicyComments](#) Plan Page privilege is also added.

- **AddRequirement:** this controls whether the Add Requirement link will display on the Secondary menu on the Segment Role History screen.
- **AddImpairment:** this controls whether the Add Requirement link will display on the Secondary menu on the Segment Role History screen.

Explanation of a Plan Page

To open a Plan Page, right-click on the page and select **Check-out**. There are four sections that display in the Configuration Area.

1. **Plan level security**: grant access to all plan pages, buttons and fields by clicking the checkbox at the top of the Configuration Area to the right of the page name.
2. **Button security**: grant access to individual buttons on the page by clicking the checkbox to the right of a button. Buttons that are unchecked will not be visible to the user in OIPA.
 - The **AllowDelete** button on a PolicyActivity Plan page grants a user access to the Delete checkbox on the Activity screen. This checkbox allows [Nonreversible/Nonreversing activities](#) and spawned activities to be deleted or recycled.
3. **Field security**: grant access to individual fields. There are three options for field security:
 - Visible and Editable: the field is both enabled and the value held in the field is visible. This is the default setting and when selected no database entry will be made.
 - Hide field value: the field is disabled, but the value held in the field is hidden.
 - Disable field value: the field is disabled and any values are visible.
4. **Masking**: Add security to the mask applied to a text field. Security levels are defined in `AsCodeMaskSecurityLevel` and `AsMaskDetail`.



Plan Security in Admin Explorer

You are here: Security > Application Security > Transaction Security in Admin Explorer

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transaction Security

After Company and Plan security have been defined, the transactions associated with the company and plans are displayed under the Transaction Security folder. Plan folders are listed inside the associated company folder.

Security can be added to all transactions in a plan by right-clicking on the plan name. Security can also be assigned to individual transactions by opening the Plan folder and selecting a specific transaction.



Transaction Security Right-Click Options



Security is applied from the top down. Once Primary Company security is defined in the Company file and the Company pages, then the Plan Security folder will populate with available plans. Once Plan security is defined, then the Transaction Security folder will populate with available transactions.

Explanation of Transaction Security

To open a Transaction Security editor, right-click on the transaction and select **Check-out**. There are four sections that display in the Configuration Area.



If the underlying rule has a context that allows state overrides, the Rules Palette will prompt the user to select a state value from a Context pop-up window. This state value is used to resolve potential copybooks for their field information.

1. **Transaction level security:** grants access to all buttons and fields associated with the transaction by clicking the checkbox at the top of the Configuration Area to the right of the transaction's name.
2. **Button security:** grant access to individual buttons and actions for the transaction by clicking the checkbox to the right of a button. Buttons that are unchecked will not be visible to the user in OIPA. Override buttons receive security from this section. These override buttons are related to the [TransactionTimes business rule](#).
 - **ActivityAddOverride:** controls the display of the Override option on the Add Activity screen when an activity error occurs due to the TransactionTimes business rule.
 - **ActivityDelete:** When an activity is pending this controls the display of the trash can  icon to the right of the activity. This icon allows a user to delete an activity.
 - **ActivityDeleteOverride:** controls the ability to override and delete an activity that could not process due to the attached TransactionTimes business rule.
 - **ActivityDetail;** controls the display of activity detail inside the Add Activity window when a new activity is added.
 - **ActivityError:** controls the display of the error  icon in the Action column on the Activity screen. This icon displays when an activity cannot process due to errors.

- **ActivityErrorOkOverride:** controls the display of the OK button on the Error Override window when the activity is halted due to the TransactionTimes business rule. A user cannot override the error without this privilege.
- **ActivityPostAssignmentError:** controls the display of the PostAssignment error icon in the Action column on the Activity screen. This icon displays when an activity cannot process due to a post assignment validation error [post assignment validation error](#).
- **ActivityPostAssignmentErrorOkOverride:** controls the display of the OK button on the Post Assignment Error Override window. A user cannot override an error without this privilege.
- **ActivityProcess:** controls the display of the lightning bolt ⚡ icon in the Action column on the Activity screen. This icon allows a user to process an activity.
- **ActivityProcessNUVPending:** controls the display of the lightning bolt icon to process an activity that is in NUV Pending status.
- **ActivityProcessNUVPendingOverride:** controls the display of the override option when an activity in NUV Pending fails to process.
- **ActivityProcessOverride:** controls the display of the override option when normal activity processing is interrupted.
- **ActivityRecycle:** controls the display of the recycle ♻️ icon to the right of the activity. It allows a user to recycle a processed activity.
- **ActivityRequirement:** controls the display of the requirement icon in the Action column on the Activity screen. This icon only applies to activities that have [requirements](#) that must be satisfied before the activity can process.
- **ActivityRequirementDeleteOverride:** controls the display of the Delete option in the Activity Requirements window, when TransactionTimes halts activity processing.
- **ActivityRequirementOkOverride:** controls the display of the Override option in the Requirement window. This checkbox is

accessed by clicking the requirement icon in the Action column on the Activity screen.

- **ActivityResult**: controls the display of the Activity Detail icon to the left of a processed activity. When clicked it shows the results of the processed activity.
- **ActivityReverse**: controls the display of the trash can  icon for activities that have already processed.
- **ActivityReverseOverride**: controls the display of the recycle  icon when TransactionTimes halts the processing of an activity.
- **ActivityUpdateOverride**: controls the display of an update option for an activity in pending status that has not been able to process due to the TransactionTimes business rule.
- **Ok**: controls the display of the OK button when the Add Activity window is open.
- **Quote**: controls the display of the Quote button in the Add Activity window when an activity is initially added. This button is only available for Client Financial and Policy Financial transactions.
- **RequirementDelete**: controls the display of the Delete option when a user right-clicks on a requirement from the Requirement window. Click the requirement icon in the Action column on the Activity screen to reveal a list of requirements with right-click menus.
- **RequirementDetail**: controls the display of the Requirement.Detail option when a user right-clicks on a requirement from the Requirement window. Click the requirement  icon in the Action column on the Activity screen to reveal a list of requirements with right-click menus.
- **RequirementOK**: controls the display of the OK button when the Requirement window is open.
- **Verify**: controls the display of the Verify button in the Add Activity window when a new activity is initially added. This button is only available for Client Financial and Policy Financial transactions.

3. **Error Overridability Security:** Specifies whether users belonging to the security group have the ability to override individual errors displayed during activity processing. The exact errors that can have their overridability configured on this pane are specified in the ValidateExpressions or PostAssignmentValidateExpressions business rule attached to the transaction. Each error on this pane has a Security drop-down box used for designating whether the error should be overridable by users belonging to the security group. The options available for selection in the Security drop-down box are as follows:
 - **Override Allowed:** Users belonging to the security group will be able to override the error. The corresponding security record will be removed from AsAuthTransactionError, the database table that holds the security data for each error number. By default, all errors are set to Override Allowed for all security groups.
 - **Override Not Allowed:** Users belonging to the security group will not be able to override the error. A security record will be added to the AsAuthTransactionError.
4. **Field security:** grant access to individual fields. There are three options for field security:
 - Visible and Editable: the field is both enabled and the value held in the field is visible. This is the default setting and when selected no database entry will be made.
 - Hide field value: the field is disabled, but the value held in the field is hidden.
 - Disable field value: the field is disabled and any values are visible.
5. **Requirement Fields:** this section will be enabled for transactions that were configured with [requirements](#). Specific levels of access to requirement fields in OIPA are set here.
6. **Masking:** Add security to the mask applied to a [text field](#). Security levels are defined in AsCodeMaskSecurityLevel and AsMaskDetail.



Transaction Security Folder Structure

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Security > Application Security > Masking

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Masking

A mask allows the configurator to define restrictions on the characters and length of the input and to format the display including the ability to hide all or part of a text field's value from view in OIPA. Here are the high level steps involved in setting up and applying a mask.

Configure a Mask

1. Define the mask security levels in [AsCode](#). Open **Admin Explorer | Administration | Code Names** and scroll to **AsCodeMaskSecurityLevel**. Check out this file to add or revise mask security levels. These security levels must be defined before a mask can be created.
2. Create a mask in [the Mask editor](#). Open **Admin Explorer | Administration | Masks**. Right-click on the Masks folder and select **New Mask Detail**. Enter the mask details. Any number of security levels may be created as needed for the intended use of the mask. For example, a phone number mask may need only one level of mask as it may not be necessary to hide the entry data based on the user. A tax ID mask may require that a few users actually view the data and all others view a set of replacement characters for the actual data. The number of security levels for this mask depends on the number of unique obfuscated views of the data in addition to the non-obfuscated view.
3. Create a transaction or screen (or open an existing transaction or screen). In the Fields pane, click on the [text field](#) that will contain a mask. The Field Properties Window will open. The last property listed is the mask property. Select a mask format. These formats are defined in the AsMaskDetail table and can be edited in the Mask editor in the Admin Explorer.
4. [Add security to Mask](#). Open **Admin Explorer | Security | Application Security | Security Groups**. Open the Security Group folder and drill down to the security folder containing the page that holds the new mask. Open it and select the transaction or page. When it opens in the Configuration Area, scroll to the bottom of the screen to the Mask section and select a level of Mask security. These levels are saved in AsCodeMaskSecurityLevel and AsMaskDetail. The former can be edited through the AsCode editor in the Admin Explorer. The latter can be edited through the Mask editor in the Admin Explorer.

Mask Formats

A mask consists of two parts. First, an input format must be defined. This is used to validate the information a user enters in OIPA. The input mask must be a regular expression. The regular expression will validate the type of characters and the number of characters entered in OIPA. The regular expression also defines how the input characters are grouped. A grouping is defined by a pair of parenthesis. If a particular group should not be included in the output a question mark and a colon can be placed before the information inside the parenthesis and the section will not be included. Ex: (?:.....)

Second, an output format must be defined to tell OIPA how to display the input by the defined groups and supplied characters such as hyphens, parenthesis, etc.

Example: Phone number with masking

Input format: (\d{3})(\d{3})(\d{4})

Output format: (\$1) \$2-\$3

Each input group is surrounded by a pair of parenthesis. In the example above there are three groups:

- **group one:** (\d{3}) /d indicates numeric characters. {3} indicates the number of characters. (\d{3}) is one group and because it is first in the definition it is assigned to \$1.
- **group two:** (\d{3}) /d indicates numeric characters. {3} indicates the number of characters. (\d{3}) is one group and because it is second in the definition it is assigned to \$2.
- **group three:** (\d{4}) /d indicates numeric characters. {4} indicates the number of characters. (\d{4}) is one group and because it is third in the definition it is assigned to \$3.

The output format shows that group one should be displayed inside a pair of parenthesis with a space before group two and a hyphen between group two and group three.

In total 10 digits are expected in the input with no additional format characters (hyphens, parenthesis, etc.) allowed in the entered data. A system error displays if any alpha character is entered or if the incorrect number of characters is entered. The output format defines, by group, how to re-display the input. In this case they are displayed in the order they were input. That does not have to be the case. The output format can mix up the groups as desired. The parenthesis, the hyphen and the space are added to create the format. With this definition the valid entry of 1234567890 becomes (123) 456-7890. If the output format was defined like this, \$1-\$2-\$3, the re-display would be 123-456-7890 (\$1.\$2.\$3 would be 123.456.7890).

Example: SSN with masking

Input format: (\d{3})(\d{2})(\d{4})

Output format: ***-**- \$3

The asterisks in the output format tell OIPA to hide the input value. The \$3 tells OIPA that the third group should be displayed. This demonstrates replacement of input characters. Entry of 123456789 becomes ***-**-6789.

Example: Entire value is hidden

Input format: (\d{3})(\d{2})(\d{4})

Output format: ***

The three asterisks in the output format tell OIPA that all three groups of input values should be hidden. This demonstrates replacement of all input characters. Entry of 123456789 becomes ***.

You are here: Security > Application Security > Mask Security

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Mask Security

Masks can be applied to [text fields](#) to maintain input character length, character types (digits, letters, etc) and format the entry including the ability to obfuscate field values from view in the OIPA application. Security for masks is applied at the transaction level, by security group. The levels of security are defined in AsCodeMaskSecurityLevel and AsMaskDetail. The [AsCode](#) and Mask editors can be used to make any adjustments to the security levels.

Masks are created through the [Mask Editor](#) in **Admin Explorer | Administration**.

Steps to Add Security to a Mask

1. Open **Admin Explorer | Security | Application Security | Security Groups**.
2. Open the Security Group folder and drill down to the Transaction Security folder or page containing the mask.
3. Open the Transaction Security or page folder, right-click on the transaction or page name and select **Check-out**. The transaction or page security file will open in the Configuration Area.
4. Scroll to the bottom of the screen and expand the Mask section. A mask will only display if it was added to a text field on the transaction or page.
5. Select a level of Mask security from the drop down box.
6. Check-in the transaction or page security file.

You are here: Security > Security on Main Menu & Client View Buttons

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Security on Main Menu & Client View Buttons

Security is enhanced to allow authorization to add a new Policy, Client or Group Customer versus allowing a User to make and save changes to an existing record. Security can be granted on the Policy, Client or GroupCustomer pages for the 'New' button and/or the 'Save' button. If a Security Group has access to the 'New' button they will automatically be granted access to the 'Save' button. If the Security Group does not have access to the applicable 'New' button they will not see the Main Menu sub menu option of 'New' under the Policy, Client, or Group Customer menus. If the User does not have access to the 'Save' button, the 'Save' button will not be visible on the applicable screen.

Prototype Configuration Detail

Prerequisites

The following are the security user considered for showcasing of prototype scenarios:

Prototype Super
Prototype Tester
Prototype Analyst

Existing Items

'Prototype Super' Security Group:

'Policy' Plan Page:

Security for 'New' button is present by default for the entire plans of the below companies:

Prototype Group Child Company
Prototype Individual Child Company

1. Add New - 'Policy' Plan Page:

Security for 'New' button is present by default for the entire plans of the below companies:

Prototype Group Child Company
Prototype Individual Child Company

'Policy' plan page:

Admin Explorer > Security > Application Security > Security Groups > Prototype Super > Plan Security > Prototype Individual Child Company > Dynamic Prototype Plan > Plan Pages > Policy

Once security for 'New' button is granted security for 'Save' button is also granted.

Result: In OIPA when the user associated to 'Prototype Super' security logs in then the 'New' sub menu option under 'Policy' main menu is shown and also user is able to 'Save' the existing/new policy content.

2. Add New - 'Client' Company Page:

Security for 'New' button is present by default in 'Client' Company page which is in the path as mention below:

Admin Explorer > Security > Application Security > Security Groups > Prototype Super > Company Security > Company Pages > Prototype > Client

Once Security for 'New' button is granted security for 'Save' button is also

granted.

Result: In OIPA, when user associated to 'Prototype Super' security group is logged in then the 'New' sub menu option under 'Client' main menu is shown and also user is able to 'Save' the existing/new client content.

3. Add New - 'GroupCustomer' Company Page:

Security for 'New' button is present by default in 'GroupCustomer' Company page which is in the path as mention below:

Admin Explorer > Security > Application Security > Security Groups > Prototype Super > Company Security > Company Pages > Prototype > GroupCustomer.

Once 'New' button is granted security for 'Save' button is also granted.

Result: In OIPA, when user associated to 'Prototype Super' security is logged in then the 'New' sub menu option under 'Group Customer' main menu is shown and also user is able to 'Save' the existing/new Group Customer content.

'Prototype Tester' Security user:

1. Save only - 'Policy' Plan Page:

Security is removed for 'New' button and security to 'Save' button is given in the 'Policy' Plan Page of Functional Prototype Plan which is in the path as mention below:

Admin Explorer > Security > Application Security > Security Groups > Prototype Tester > Plan Security > Prototype Individual Child company > Functional Prototype Plan > Plan Pages > Policy

Result: In OIPA, when the user associated to 'Prototype Tester' is logged in, the 'New' sub menu option under 'Policy' main menu is not shown.. The user is able to 'Save' the existing Policy content of Functional Prototype Plan policy.

2. Save only - 'Client' Company Page:

Security is removed for 'New' button and security to 'Save' button is given in the 'Client' Company page which is in the path as mention below:

Admin Explorer > Security > Application Security > Security Groups > Prototype Tester > Company Security > Company Pages > Prototype > Client

Result: In OIPA, when the user associated to 'Prototype Tester' security is logged in, the 'New' sub menu option under 'Client' main menu is not shown. The user is able to 'Save' an existing Client contents.

3. Save only - 'Group Customer' Company Page:

Security is removed for 'New' button and security to 'Save' button is given in the 'Group Customer' Company page which is in the path as mention below:

Admin Explorer > Security > Application Security > Security Groups > Prototype Tester > Company Security > Company Pages > Prototype > Group Customer.

Result: In OIPA, when the user associated to 'Prototype Tester' security is logged in, the 'New' sub menu option under 'Group Customer' main menu is not shown. The user is able to 'Save' an existing Group Customer contents.

'Prototype Analyst' Security user:

1. No New or Save - 'Policy' Plan Page:

Security is removed for both 'New' and 'Save' buttons in the 'Policy' Plan Page of the Dynamic Prototype Plan which is in the path as mention below:

Admin Explorer > Security > Application Security > Security Groups > Prototype Analyst > Plan Security > Prototype Individual Child company > Dynamic Prototype Plan > Plan Pages > Policy

Result: In OIPA, when the user associated to 'Prototype Analyst' security is logged in, the 'New' sub menu option under 'Policy' main menu is not shown. The 'Save' button is not available for the existing Dynamic Prototype Plan Policies.

2. No New or Save - 'Client' Company Page:

Security is removed for both 'New' and 'Save' buttons in the 'Client' Company page which is in the path as mention below

Admin Explorer > Security > Application Security > Security Groups > Prototype Analyst > Company Security > Company Pages > Prototype > Client

Result: In OIPA, when the user associated to 'Prototype Analyst' security is logged in, the 'New' sub menu option under 'Client' main menu is not shown. The 'Save' button is not available for an existing Client.

3. No New or Save - 'Group Customer' Company Page:

Security is removed for both 'New' and 'Save' buttons in the 'GroupCustomer' Company page which is in the as mention below

Admin Explorer > Security > Application Security > Security Groups > Prototype Analyst > Company Security > Company Pages > Prototype > Group Customer.

Result: In OIPA, when the user associated to 'Prototype Analyst' security is logged in, the 'New' sub menu option under 'Group Customer' main menu is not shown. The 'Save' button is not available for an existing Group Customer.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Security > Security Needed by Inquiry Screen

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Security Needed by Inquiry Screen

OIPA supports 3 different levels of Inquiry Screen rules accessed off of the Main Menu, Policy screens or Client/Group Customer screens. Security is granted separately for each level/InquiryScreen name

Security - Main Menu Level

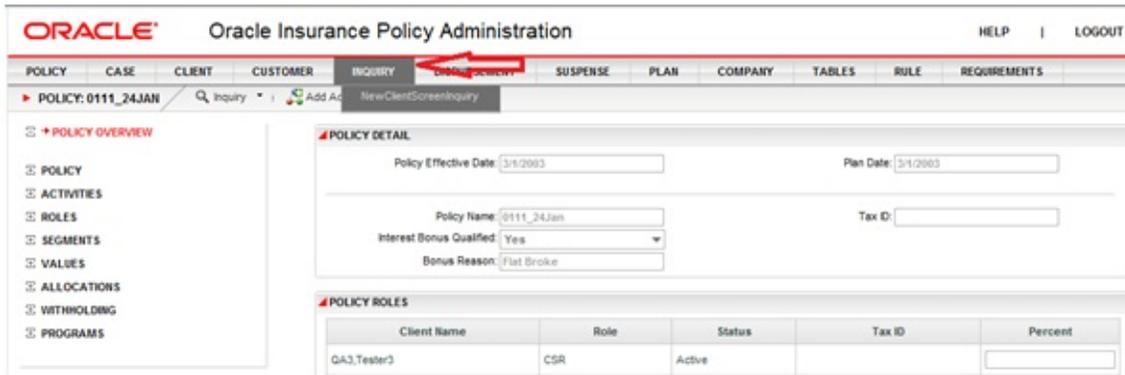
The 'Inquiry' main menu button is visible if the Security Group has authorization for one or more Main Menu level InquiryScreen rules for the applicable Primary Company.

Security is granted for Main Menu level Inquiry Screens by granting access following the path as mention below

Admin Explorer > Security > Application Security > Security Groups > 'SecurityGroup' > Inquiry Security > Main Menu > 'InquiryScreenName'

When the user hovers over the Inquiry button, the list of Main Menu level InquiryScreen Names/Rules the Security Group has access to is displayed.

If the Security Group does not have access to at least one Main Menu level InquiryScreen rule for the applicable Primary Company the 'Inquiry' button will not be visible on the Main Menu.



Security - Policy Level

The 'Inquiry' button is visible from any 'Policy' page as long as the Security Group has access to at least one 'Policy' level InquiryScreen name/rule for the policy's plan, product(s) or company.

Security is granted for Policy level Inquiry Screens by granting access following the path as mention below

Security set at Parent Product level:

Admin Explorer > Security > Application Security > Security Groups > 'SecurityGroup' > Inquiry Security > Policy Level > 'Company' > 'Product' > Inquiry Screens > 'InquiryScreenName'

Security set at Child Product level:

Admin Explorer > Security > Application Security > Security Groups > 'SecurityGroup' > Inquiry Security > Policy Level > 'Company' > 'Product' > Child Products > 'Child Product' > Inquiry Screens > 'InquiryScreenName'

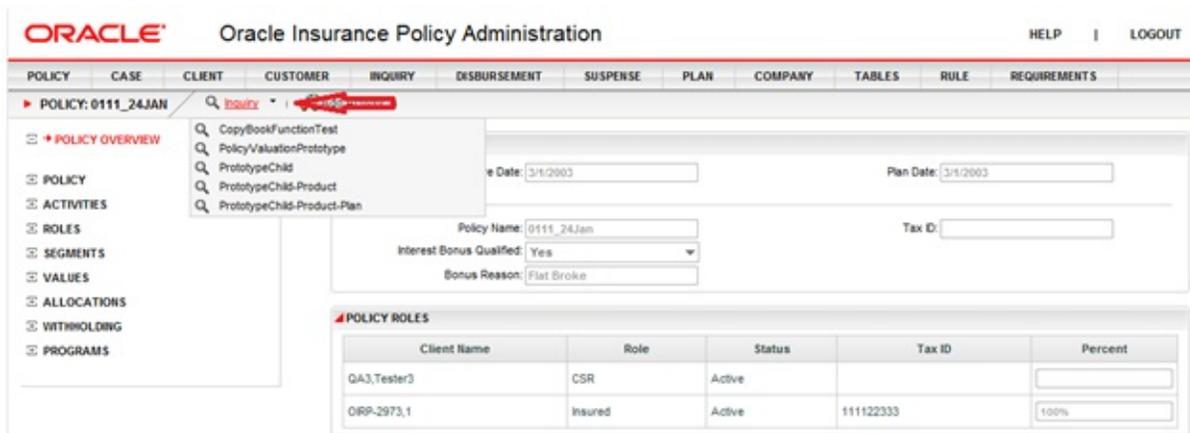
Security set at Plan level:

Admin Explorer > Security > Application Security > Security Groups > 'SecurityGroup' > Inquiry Security > Policy Level > 'Company' > 'Product'

> Plans > 'Plan' > 'InquiryScreenName'

When the user hovers over the Inquiry button from any Policy page, the InquiryScreen Names/Rules the Security Group has access to will be displayed.)

If the Security Group does not have access to any Policy Level InquiryScreen name/rule then the Inquiry button will not appear on any 'Policy' screens.



Security - Client Level

The 'Inquiry' button is visible from any 'Client' or 'Group Customer' page as long as the Security Group has been granted access to at least one 'Client' level InquiryScreen Name/Rule for the applicable Primary Company.

Security is granted for Client level Inquiry Screens by granting access following the path as mention below

Admin Explorer > Security > Application Security > Security Groups > 'SecurityGroup' > Inquiry Security > Client Level > 'InquiryScreenName'

When the user hovers over the Inquiry button from any Client or Group Customer page, the InquiryScreen Names/Rules the Security Group has

access to will be displayed.

The screenshot displays the Oracle Insurance Policy Administration web application. The top navigation bar includes the Oracle logo, the title "Oracle Insurance Policy Administration", and links for "HELP" and "LOGOUT". Below this is a menu with tabs for "POLICY", "CASE", "CLIENT", "CUSTOMER", "INQUIRY", "DISBURSEMENT", "SUSPENSE", "PLAN", "COMPANY", "TABLES", "RULE", and "REQUIREMENTS". The "CLIENT" tab is active, showing "CLIENT: TOMAGENT". A search bar contains "Inquiry" and "Face Amounts Inquiry". A red arrow points to the "Client Activity" link in the search bar. On the left, a sidebar menu lists "CLIENT" (expanded) with sub-items: "Client Detail", "ADDRESS", "GROUP", "ROLES", "HISTORY", "ENROLLMENT", "ACTIVITIES", "WITHHOLDING", "COMMENTS", and "RELATIONSHIPS". The main content area shows the "CLIENT" details form for "Tom Agent". Fields include: Type (Individual), Legal (United States Of America), Prefix, Middle Name, Gender (Male, Female, Unisex), Marital Status (Unknown), Date of Birth (1/1/1950), Date of Death, Birth Country (Canada), Email Address, First Name (Tom), Last Name (Agent), Tax ID (112345677), Citizenship Country (Canada), and Text Field (00). Below this is the "CLIENT DETAIL" section with "TaxID Source" (Application) and "Date to Send Correspondence" (empty), followed by a "Save" button.

Configuration Details

This change does not require any Rule configuration changes.Details.

You are here: Security > Security for New Companies, Plans and Transactions

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Security for New Companies, Plans and Transactions

When a [new transaction](#), plan or company is created, it must have security applied to it or it will not be visible in OIPA. Configurors will be able to see the new item in the Rules Palette and can edit it if necessary. OIPA users will not be able to view the new item until the Security Manager assigns security to it.



Warning Message to Update Security



The Security Manager must be notified each time a new item is created so that security can be added.

Steps to Assign Security to a New Item

1. Open the [Admin Explorer](#) tab in the Rules Palette.
2. Open **Security | Application Security | Security Groups**.
3. Right-click on Security Groups node and select **Refresh**. This will update the folders with the new item that was added. Make sure any security changes previously made are saved before performing refresh.
4. Open the folder for the security group that should receive access to the new item.
5. Open the folder that corresponds to the type of item added. If it is a new company, open the [Company Security](#) folder. If it is a new plan, open the [Plan Security](#) folder. Likewise, if it is a new product, open the [Plan Security](#) folder, as transaction security is defined for each plan within the product. If it is a new transaction, open the [Transaction Security](#) folder.
6. Locate the new item in the folder structure. Double-click on the name to open it in the Configuration Area.
7. Click the check box at the top to grant security to all pages, buttons and fields or expand the individual sections to add security to individual buttons or fields.
8. If assigning security to a transaction, and if masking was added to a text field in the transaction, expand the Masks section and select a security level for the mask.
9. If assigning security to a transaction, and if overridable errors are configured in a ValidateExpressions or PostAssignmentValidateExpressions business rule attached to the transaction, set the overridability of each error by selecting **Override Allowed** or **Override Not Allowed** from each Security drop-down box. By default, all errors will be set to Override Allowed for all security groups.
10. Repeat steps 3 through 9 for all security groups that need access to the new item.
11. Check in all security group files to make sure the changes

are saved to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

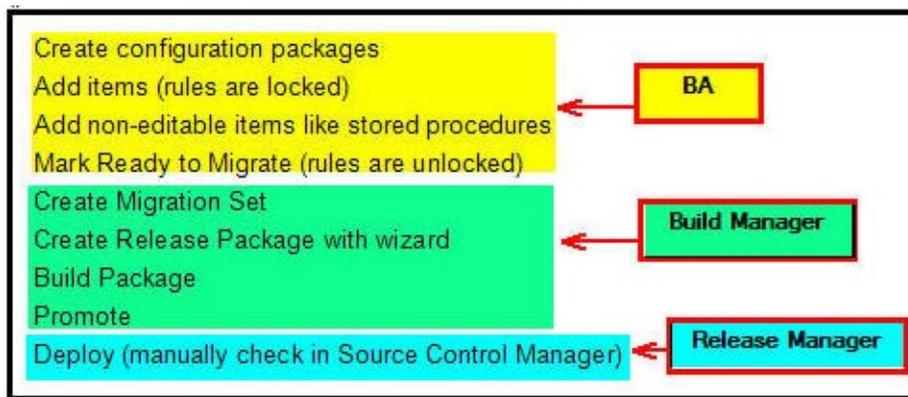
You are here: Release Management > Release Management Security

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Release Management Security

Security plays an important role in the Release Management process. The Security Manager will need to set security levels for configurators and managers who will be involved in the process. The image below shows a typical security breakdown but it can be modified to meet specific business needs. The Release Management folders will not be visible to users unless they have received the proper security role.

Refer to the [Rules Palette Security](#) section for details on creating Release Management Security Roles.



Suggested Security Breakdown for Release Management

Version History

There is also a level of security applied to the data to protect the integrity of the environment where configuration packages are created. When a rule or transaction is checked-out, a snapshot of the item is taken and stored in the Version History folder. The item then becomes locked and can only be edited by the user who checked it out for the configuration package. If another user tries to check-out that item, a message will appear indicating that the item is locked by another user. This prevents rules from being overwritten while they are in the release process.

Once a configuration package is complete, the BA will mark it as Ready to Migrate. At this point, the rules and transactions that were locked will become unlocked and will be available for other users to edit and use.

If a configuration package is placed on hold, the rules and transactions are released and made available to users. This is done to keep the flow of business moving. If the rules and transactions marked for the on-hold configuration package are changed, they will also be changed in the configuration package, since they are no longer locked. When resuming work on an on-hold configuration package, review the attached rules and transactions.

You are here: Security > Comment Security

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Comment Security

Comments are supported at the activity, client, segment, suspense and policy levels in OIPA. Security is applied to each type of comment at several points to provide a granular level of control to the comment functionality.

Each type of comment is listed below, along with the security controls that are in place.

Activity Comments

Comments can be added to activities by clicking the comment icon next to an activity listed on any of the Activity screens in OIPA. The display of the comment icon is controlled by the following security privileges.

From the [Company Security](#) level in the **Company Pages** folder:

- [ClientActivity](#) Company Page has an option for ActivityComments. Check this box to display the comment icon next to activities on the Client Activity screen.
- [CompanyActivity](#) Company Page has an option for ActivityComments. Check this box to display the comment icon next to activities on the Company Activity screen.
- [PlanActivity](#) Company Page has an option for ActivityComments. Check this box to display the comment icon next to activities on the Plan Activity screen.

From the [Plan Security](#) level in the **Plan Pages** folder:

- [PolicyActivity](#) Plan Page has an option for ActivityComments. Check this box to display the comment icon next to activities on the Policy Activity screen.

Client Comments

There are three security privileges that must be applied to grant access to client comments. All three privileges reside at the [Company Security](#) level in the Company Pages folder.

- **Client:** [Client Comments checkbox](#). This controls the display of the Add Comment icon on the Secondary menu of the Client screen. You can only add comment information if the ClientComments privilege is also granted.
- **ClientActivity:** [Activity Comments](#) and [Client Comments](#) must be checked.
- **ClientComments:** this provides you with the ability to add, delete, edit and view comments. Make sure the Comment Buttons on this security page have security applied to the **Allowed** drop down boxes. If a Yes value or a View value is not applied, then you will not be able to do anything with client comments in OIPA.
- **ClientCommentsSearch:** this controls the display of the Comments link on the Left Navigation menu of the Client screen. It also provides access to the Client Search screen.

Policy Comments

There are three security privileges that must be applied to grant access to policy comments. All three privileges reside at the [Plan Security](#) level in the Plan Pages folder.

- **Policy:** [Policy Comments checkbox](#). This controls the display of the Add Comment icon on the Secondary menu of the Policy screen. You can only add comment information if the PolicyComments privilege is also granted.
- [PolicyComments](#): this provides you with the ability to add, delete, edit and view comments. Make sure the Comment Buttons on this security page have security applied to the **Allowed** drop down boxes. If a Yes value or a View value is not applied, then you will not be able to do anything with policy comments in OIPA.
- [PolicyCommentsSearch](#): this controls the display of the Comments link on the Left Navigation menu of the Policy screen. It also provides access to the Policy Search screen. You will be able to also view segment comments from segments attached to the policy.

Segment Comments

There are two security privileges that must be applied to grant access to segment comments. All three privileges reside at the [Plan Security](#) level in the Plan Pages folder.

- [PolicySegment](#): **Policy Comments checkbox**. This controls the display of the Add Comment icon on the Secondary menu of the Policy screen. You can only add comment information if the SegmentComments privilege is also granted. **Segment Comments checkbox** controls the display of the comment icon next to segments on the Segment screen.
- [SegmentComments](#): this provides you with the ability to add, delete, edit and view comments. You must assign at least View privileges from the Comment Button section or the comment icon will not be available on the segment.

Suspense Comments

There are three security privileges that must be applied to grant access to suspense comments. All three privileges reside at the [Company Security](#) level in the Company Pages folder.

- **SuspenseRecord:** [Suspense Comments checkbox](#). This controls the display of the Add Comment icon on the Secondary menu of the Suspense Record screen. You can only add comment information if the SuspenseComments privilege is also granted.
- [SuspenseComments](#): this provides you with the ability to add, delete, edit and view comments. You must assign at least View privileges from the Comment Button section or the comment icon will not be available on the suspense record.
- [SuspenseCommentsSearch](#): this controls the display of the Comments link on the Left Navigation menu of the Suspense screen. It also provides access to the Suspense Search screen.
- **Suspense History:** the SuspenseComments check box must be selected.

Requirement Comments

There are two security privileges that must be applied to grant access to requirement comments. Both privileges reside at the [Plan Security](#) level in the Plan Pages folder.

- [PolicyRequirement](#): this page controls the buttons that display on the Policy Requirement screen. You can only add comment information if the RequirementComments privilege is also granted.
- [RequirementComments](#): this page controls a user's ability to add, delete and view requirement comment information. You must assign at least View privileges from the Comment Button section or the comment icon will not be available on the suspense record.

Impairment Comments

There are two security privileges that must be applied to grant access to Impairment comments. Both privileges reside at the [Plan Security](#) level in the Plan Pages folder.

- [PolicyAssessment](#): this page controls the buttons that display on the Assessment screen. You can only add comment information if the ImpairmentComments privilege is also granted.
- [ImpairmentComments](#): this page controls a user's ability to add, delete and view Impairment comment information. You must assign at least View privileges from the Comment Button section or the comment icon will not be available on the suspense record.

You are here: Admin Explorer > Activity Filters

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

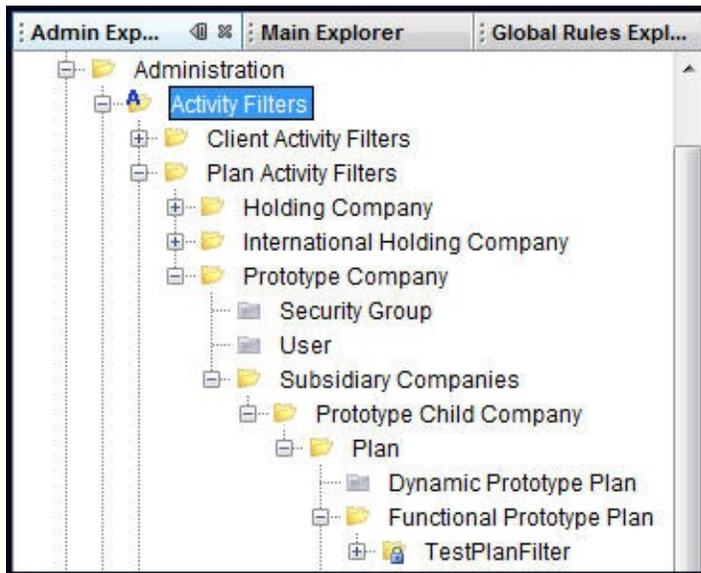
Activity Filters

The Activity Filter editor is located in the Admin Explorer tab's Administration folder. This editor is used to configure filters that control the activities that display on OIPA's various activity screens. Using activity filters will ensure that only relevant activities will display on any given activity screen.

The editor is divided into two sections: **Filter - General** and **Details**. The only field that can be edited in the Filter - General section is the Name field. Once a filter is created, the other fields in the Filter - General section are locked down. The Details section is used to specify which transactions should display on the activity screen being configured. These can be edited at any time.

Types of Activity Filters

Activity filters are based on three criteria: activity screen, company, and filter level. Available activity screens are Client, Plan and Policy. When creating an activity filter, the filter level is specified using the Company, Product (if Products are enabled), Plan, Security Group and/or User fields of the New Activity Filter wizard. The Activity Filter folder hierarchy in the Admin Explorer is based on these same criteria.



Activity Filters Folder Structure

Only one activity filter can exist at a given level (i.e. no two filters can exist for the same combination of activity screen, company and Product/plan/security group/user), and activity filter configuration options differ depending on the activity screen for which the filter is being created:

Client Screen Activity Filters

Creating an activity filter for this screen will control the activities a particular security group or user sees on the Client Activity screen in OIPA. Activity filters can only be created at the primary company level for the Client Activity screen. Additionally, Product- and Plan-level activity filters are not available for the Client Activity screen.

Plan Screen Activity Filters

Creating an activity filter for this screen will control the activities a particular security group or user will see on the Plan Activity screen in OIPA. Product- and Plan-level activity filters can be created for the Plan Activity screen if the filter corresponds to a subsidiary company. Security Group- and User-level filters can be created if the filter corresponds to a primary company.

Policy Screen Activity Filters

Creating an activity filter for this screen will control the activities a particular security group or user will see on the Policy Activity screen in OIPA. Product- and Plan-level activity filters can be created for the Policy Activity screen if the filter corresponds to a subsidiary company. Security Group- and User-level filters can be created if the filter corresponds to a primary company.

Configuration Requirements

In addition to the Activity Filters node in the Admin Explorer, there is an AsCode table created for ActivityFilters: AsCodeFilterType. This table holds code values for each filter type, and can be modified by navigating to **Admin Explorer | Administration | Code Names | AsCodeFilterType**.

Steps to Create a New Activity Filter



Only one filter can exist at a given level. No two filters can exist for the same combination of activity screen, company and Product/plan/security group/user.

1. Navigate to the **Admin Explorer**.
2. Open the **Administration** folder and locate the **Activity Filters** folder.
3. Right-click on the **Activity Filters** folder and select **Add New Activity Filter**. The New Activity Filter wizard will open.
4. Select the activity screen to which the filter applies from the **Activity Screen** drop-down box. This is a required field. When an activity screen is selected, the Company field becomes enabled. If Client is chosen, then the Security Group and User fields are enabled, while the Product (if applicable) and Plan fields are disabled.
5. Select a company from the **Company** drop-down box. This is a required field. Activity filters can be created for primary companies or subsidiary companies--however, if Client was chosen in the Activity Screen field, only primary companies will be available for selection.
6. Enter a name for the filter in the **Name** field. This is a required field. Activity filter names can be identical, but only if the filters are for different activity screens.



The system will return a warning message when the filter is checked in if a duplicate name exists, in order to verify that the user is aware of the shared names. If the user tries to check in an activity filter with an identical name to another filter for the same activity screen, then the system will return an error message.

7. If Products are enabled in the environment, then select one from the **Product** drop-down box. This is an optional field. This field will only be enabled if Plan or Policy was chosen from the Activity Screen field.
8. Select a plan from the **Plan** drop-down box. This is an optional field. This field will only be enabled if Plan or Policy was chosen from the Activity Screen field. Additionally, if a Product was selected, then only the plans corresponding to that Product will be available for selection.
9. Select a security group from the **Security Group** drop-down box. This is an optional field. This field will only be enabled if Client was chosen in the Activity Screen field, or if a primary company was chosen in the Company field. If the security group is selected, then the User field will become disabled.
10. If needed, select a user from the **User** drop-down box. This field will only be enabled if Client was chosen in the Activity Screen field, or if a primary company was chosen in the Company field. This is an optional field. If a user is selected in the User field, then the Security Group field will become disabled.

The New Activity Filter Wizard

Steps to Edit or Delete an Activity Filter

1. Navigate to the **Admin Explorer**.
2. Open the **Administration** folder.
3. Open the **Activity Filters** folder and locate the filter to be edited or deleted.
4. Right-click on the filter's XML file. To edit the filter, select **Check out**. To delete the filter, select **Delete**.
5. If editing the activity filter, make the needed changes. In the Filter - General section of the Configuration Area, only the Name field can be edited. In the Details section, use the >> and << buttons to move transactions between the **Included Transactions** and **Excluded Transactions** boxes. Only the transactions listed in the Included Transactions box will be displayed in OIPA.



If checking in an activity filter with a name already held by another filter for the same activity screen, the system will return an error message, and the check in will be stopped. If checking in an activity filter with a name held by another filter

for a different activity screen, the system will display a warning message, but check in will proceed.



An activity filter must have at least one transaction in the Excluded Transactions box to be checked in.

6. Save any changes to the database by right-clicking on the activity filter's XML file and selecting **Check in**.

You are here: Admin Explorer > Admin Explorer Tab

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

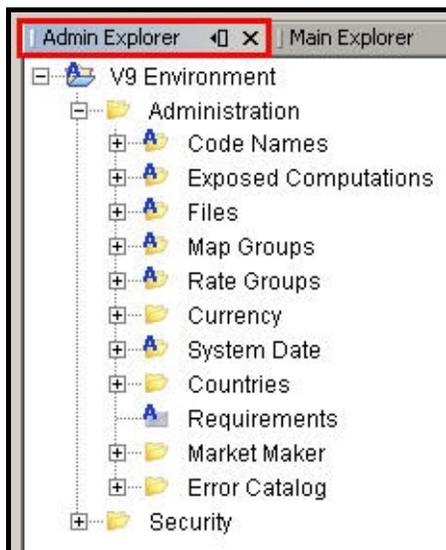
Admin Explorer Tab

The Rules Palette administration tab, **Admin Explorer**, consists of sub-folders for each administration task that may be performed. These tasks include functions such as [defining security roles and privileges](#), managing available [type codes](#), uploading [rates](#), [migrating rules](#) as well as managing various other metadata required for the system to function in accordance with an individual client's requirements.



[Security](#) resides in the **Admin Explorer** tab, but it is only visible to users who have been assigned the appropriate security privileges.

A [Release Management](#) folder will be available in the Admin Explorer tab if the functionality is turned on in the Web Application Utility and the user has release management privileges.



Admin Explorer Folders

You are here: Admin Explorer > Administration Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Administration Overview

As part of Oracle's commitment to improve customer experience and satisfaction with our products, much of the system administrative functionality for Oracle Insurance Policy Administration (OIPA) system has been migrated to the Rules Palette. Administrative functions that are now performed in the Rules Palette include:

- [Activity Filters](#)
- [Chart of Accounts](#)
- [Allocation Models](#)
- [Enrollment Transactions](#)
- [Code Names](#)
- [Comment Templates](#)
- [Cycle Sequence](#)
- [Exposed Computation](#)
- [Files](#)
- [Funds](#)
- [Fund Asset Classes](#)
- [Intake Profile Definition](#)
- [Map Groups](#)
- [Masks](#)
- [Rate Groups](#)
- [Currency](#)
- [State Approvals](#)
- [System Date](#)
- [Countries](#)
- [Requirements](#)
- [Market Maker](#)
- [Error Catalog](#)
- [Web Services](#)

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Admin Explorer > Chart of Accounts > Chart of Accounts Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Chart of Accounts Overview

Chart of Accounts (CoA) is used to record transactions in a company's general ledger. As part of the accounting cycle, the CoA is used in the journaling process (i.e., performing journal entries) and also serves as the title for each ledger. In general, all accounts are filed under one of five categories: assets, liabilities, owner's equity, revenue and expenses.

Each company may use its own CoA numbering system and may assign a number for identification purposes. A block of numbers may be assigned to one of those five categories and then be further divided into sub-categories. Gaps may be left between sub-category listings to allow for the addition of new accounts.

ACME Accounting Number Category	ACME Accounting Number Sub Category	ACME Account Numbers	
1000-1099 Assets	1000-1009 Cash	1001 Cash – Regular Checking	
		1002 Cash – Payroll Checking	
		1003 Cash, Change or Petty Cash Fund	
	1010-1019 Receivables	1011 - Accounts Receivables	
		1012 - Policy Fee	
		1013 - Late Charge	
		1014 - Cancellation Charge	
		1015 - Premium Payment	
	2000-2099 Loan	2000-2009 Current Loans	
	3000-3099 Liabilities	3000-3009 Current Liabilities	
4000-4099 Equity	4000-4009 Stocks	4001 - Preferred Stock	
		4002 - Common Stock	
		4003 - Retained Earnings	
5000-5099 Expenses	5000-5009 Insurance	5001 - Disability Insurance	
		5002 - Property & Causality Insurance	
	5010-5019 Interest Expense	5010 - Finance Charge	
		5011 - Loan Interest	
		5020-5029 Mortgage	

Sample of Chart of Accounts

 Common practice is to use cycle to generate accounting from

executed transactions, gather that data into a format ready for an interface, send the data to middleware, and generate a configurable report for delivery to interested parties. There is no direct communication to a general ledger system.

Chart of Accounts in OIPA

CoA is set-up at the company level in OIPA. When a financial transaction or suspense is processed and a CoA record(s) is present, an accounting record is generated that associates an account number with each piece of money as defined. The accounting records are generated when an activity is processed/reversed or when the suspense record is added/deleted.

Accounting records are stored in the AsAccountingDetail and AsAccountingDetailField tables. The AsAccountingDetail table stores the information for debit and credit amounts, which result from an activity or suspense add/deletion. It also stores information for account number, processing, reversing, gain/loss, dates and any information needed to initiate activity/suspense. If configured, additional information may be captured according to the client requirements in the AsAccountingDetailField table.

General Accounting

CoA entries that are based on math variables or non-fund specific allocations (general accounting) will use values in the transaction math and will write these values to the database when two conditions are met:

1. GenerateAccounting is set to **True**.
2. The activity processes successfully without any non-overridden business errors.

Separate Accounting

CoA entries that are based on funds (separate accounting) will wait until all AsValuation records for the activity are moved into an active state and there are no non-overridden business rule errors generated from the [PostAssignmentValidateExpressions](#) rule.

If there are no PostAssignmentValidateExpressions validation errors and all of the valuation records are in an active state for either type of accounting, even if the activity is still in an NUV Pending status, the Accounting records for each fund can be written. For each execution of the activity, the accounting will run again but will check to see if a fund

accounting record already exists for the activity. Even though the valuation records may be active before the activity moves into an active state, the accounting will not be written if there are validation errors in `PostAssignmentValidateExpressions`.

Account to Transaction Relationship

An account may be associated with one or more financial transactions and/or suspense actions. This is a one-to-many relationship: account to transactions.

Example:

Account 1001 is associated to receive premium via money types from two different transactions:

1. InitialPremium transaction
2. AdditionalPayment transaction

Transaction to Account Relationship

In CoA, an account is higher than a transaction in the hierarchy because a transaction is associated to accounts. It may be helpful from a configuration context to understand that a transaction can have one or more accounts associated with it depending on money types, plan, math variable, status of account, transaction effective date, reversal indicator, gain/loss indicator and user-defined criteria. This is a one-to-many relationship: transaction to accounts.

Example:

A disbursement transaction may have two different accounts associated with it:

1. Account 9001 for a Variable Annuity plan
2. Account 9002 for a Variable Payout plan



Transactions are associated with accounts at the entry level, which is set-up during step three of the CoA wizard.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Admin Explorer > Chart of Accounts > Before Creating CoA

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Before Creating CoA

Chart of Accounts (CoA) is configured in conjunction with the initial implementation of transactions or suspense for an OIPA plan. The CoA record(s) should be configured after the transaction or suspense configuration is complete.

The CoA may be modified as new plans, transactions, suspense and accounts are added to OIPA. There are two different ways to set-up the CoA:

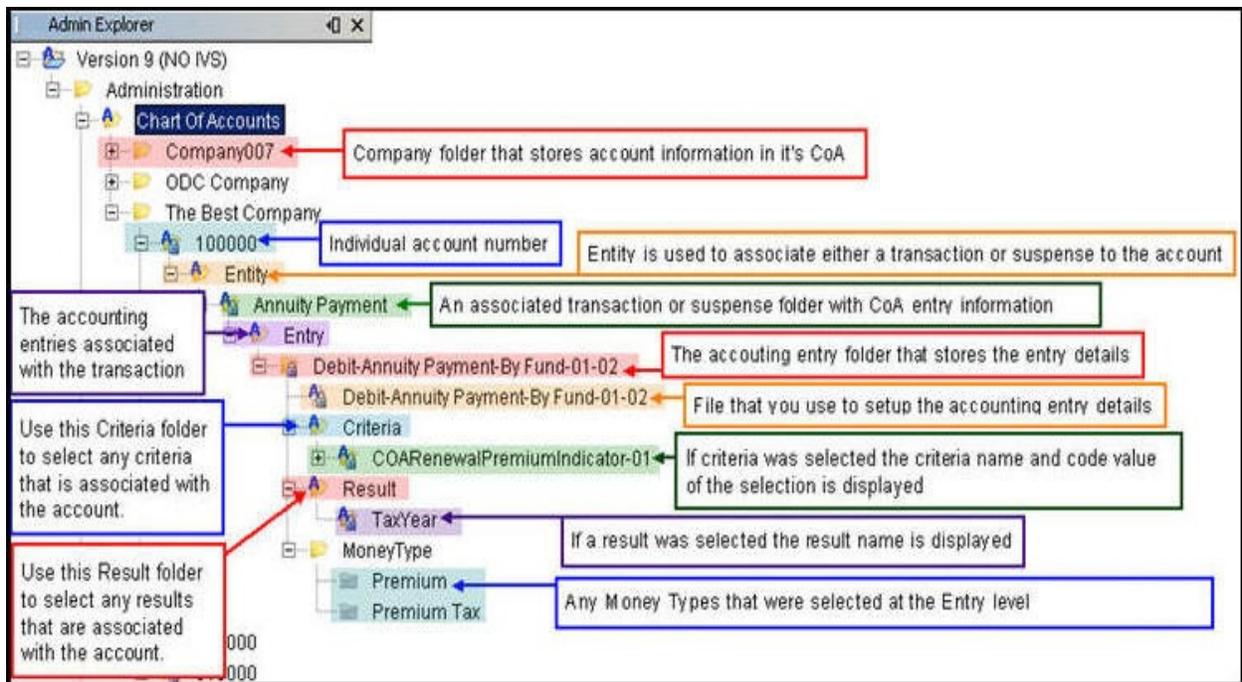
- [through the Rules Palette](#)
- using SQL statements: Initial upload of a CoA may be easier via SQL statements due to the large number of accounts.

Gather Requirements

The CoA is configured in the Rules Palette via the **Admin Explorer** window. To properly set-up CoA the following information is needed:

- a list of accounts and associated numbers
- the account type: debit or credit
- the account association: is it transaction or suspense
- the name of the transaction associated with the account
- required accounting entry details
- additional criteria that distinguishes the account to be used if transactions are associated with multiple accounts
- any other additional information that should be captured in the accounting record for this type of account

The CoA folder stores account information in a hierarchy format with account numbers and entry information in the associated company folder. Review the illustration below for details about the hierarchy structure.



CoA Folder Hierarchy in Admin Explorer

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Admin Explorer > Chart of Accounts > Chart of Account Entry Database Definitions

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Chart of Account Entry Database Definitions

AsChartofAccountsEntry
CHARTOFACCOUNTSENTRYGUID
CHARTOFACCOUNTSENTITYGUID
DEBITCREDITCODE
ENTRYDESCRIPTION
ACCOUNTINGTYPECODE
ACCOUNTINGAMOUNTFIELD
GAINLOSSFLAG
FLIPONNEGATIVEFLAG
EFFECTIVEFROMDATE
EFFECTIVETODATE
DOREVERSALACCOUNTINGFLAG
ORIGINALDISBURSEMENTSTATUSCODE
FUNDTYPECODE
ACCOUNTNUMBERFORMAT
FUNDMOVEMENTDETAILSTATUSCODE
FUNDMOVEMENTNAME
LINKSUSPENSEFLAG

AsChartofAccountsEntry Database Table

- **AsChartofAccountsEntry**: The Debit/Credit entry for an account that must be tied to transaction or suspense.
- **ChartOfAccountsEntryGUID**: Unique identifier for this table.
- **ChartOfAccountsEntityGUID**: GUID to the AsChartofAccountsEntity table.
- **DebitCreditCode**: Indicates 01 for debit or 02 for a credit.
- **EntryDescription**: Description of the debit or credit.
- **AccountingTypeCode**: Stores the code for the way the amount for the accounting will be obtained. 01 for Total of Funds, 02 for ByFund, 03 for MathVariable, 04 for suspense, 05 for disbursement.
- **AccountingAmountField**: Name of the MathVariable to use as the amount if AccountingTypeCode is set as a type MathVariable. For disbursements, this is automatically set to **DisbursementAmount**.
- **GainLossFlag**: Stores 1 (Yes) or 0 (No) indicating if the debit/credit amount will be calculated based on gain/loss values.
- **FlipOnNegativeFlag**: Stores 1 (Yes) or 0 (No) indicating if credits should be flipped to debits and vice-versa when an amount is

negative.

- **EffectiveFromDate:** Effective Date the entry will be effective from.
- **EffectiveToDate:** Effective Date the entry will be effective to.
- **DoReversalAccountingFlag:** Stores 1 (Yes) or 0 (No) indicating if accounting information should be captured when activity is reversed. The default value is 1 and only applies to Disbursement accounting.
- **OriginalDisbursementStatusCode:** When an activity processes, the disbursement status is matched to the appropriate COA entry record based on the value of this code. Pending, Pending Shadow, Recoverable and Recovered are the available status options.
- **AccountNumberFormat:** Holds the format for the account number. This is not implemented in v9 yet and may be left blank.
- **LinkSuspenseFlag:** Stores 1 (Yes) or 0 (No) to indicate whether suspense records should be linked to the activity.

You are here: Admin Explorer > Chart of Accounts > Chart of Accounts and Disbursement

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Chart of Accounts and Disbursements

Disbursements are removals of money from the administration system's accounts into outbound accounts or accounts for downstream systems. The accounting may actually be accomplished as the money is moved out of a policy (non-disbursement activity) or may be delayed until the disbursement is executed.

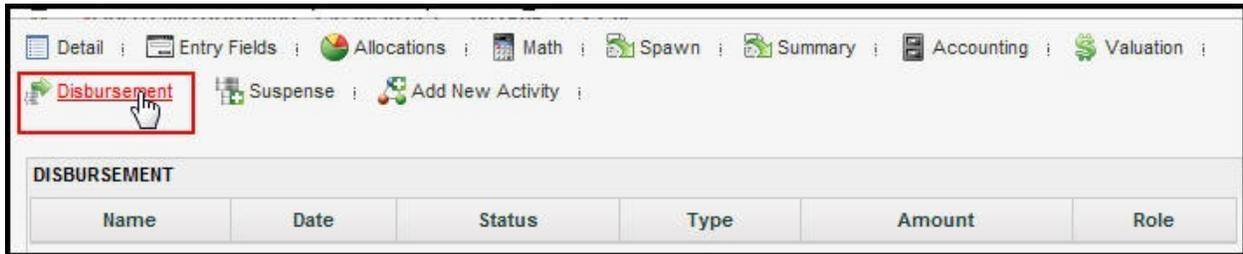
High Level Steps to Configuring Disbursements

Chart of Accounts plays an important role in disbursement processing. Entities and accounts must be set up in order for OIPA to properly track disbursements. There are also several screens and transactions that need to be configured to support disbursements. The main components of disbursement configuration are outlined below.

1. Configure the [DisbursementScreen, DisbursementApprovalScreen and DisbursementSearchScreen](#).
2. Create the necessary policy level transactions to spawn a disbursement transaction that triggers the removal of money from the policy.
3. Configure any necessary client level disbursement transactions to remove money from suspense accounts.
4. Create a separate CoA for each disbursement status that will be supported. [View details for disbursements on the CoA wizard](#). Available statuses are:
 - Active = 01
 - Recoverable = 12
 - Pending = 02
 - PendingShadow = 34
 - Recovered = 27
 - TaxableOffset = 28
 - Unrecovered = 44
5. Create CoA Entities for transactions that initiate a disbursement. This is only needed if the processing of the transaction has accounting implications.
6. Determine if transaction processing will be used as a means to update disbursement status. If the answer is **Yes**, then a transaction must be configured to perform this. The [DisbursementUpdate](#) rule should be configured and attached to this transaction. CoA Entity must also exist to trigger accounting if this option is configured in DisbursementUpdate. [View prototype example](#)

View Disbursement Information in OIPA

When a disbursement activity is processed, the resulting account information can be viewed from the Activity Results window. From the Activity screen, click the Activity Detail icon to the left of the processed activity. This will open the Activity Results window. A **Disbursement** link is provided at the top of the window. All disbursement details can be viewed from that link.



Disbursement Link in Activity Result Window in OIPA

CoA and Disbursement Detail

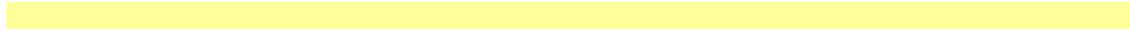
The disbursement status is originally pending. If the disbursement activity was reversed the status would be recovered. In some versions the status may be Pending Shadow.

The Entry Detail section of the CoA creation wizard has specific fields that are associated with setting-up disbursement accounts.

- **Type** - Select disbursement when account is a type of disbursement.
- **Gain Loss/Negative** – If setting up a type of account that is either ByFund or Total Fund, check this box to indicate that accounting will only happen if there is a gain or loss to funds. For example, a FundTransfer activity may use this selection or it could be used in the reversal/recycle of an activity that would result in a difference of the amount because of the fluctuation in market fund value.
- **Flip on Negative** - This checkbox is no longer needed as now all negative accounting is handled in the OIPA code.
- **Original Disbursement Status** - Activities generally have an active and pending status, but for a disbursement activity that sends money out to a client, there is a chance the money may not be received by the client. This field is used to track the status of the money in case an adjustment needs to be made. The status names in this field are pulled from AsCodeDisbursementStatus.
 - Pending – Select for forward accounting. This is the status when an activity is processed for the first time.
 - Recoverable – Select for reverse accounting. For example, if overpayment or recovery of payment, such as broker commission of a policy, that was surrendered. Money is not always recovered.



If this is selected then specific configuration using the disbursement element tag in the transaction is required. The disbursement element tag has an attribute called recoverable that needs to be set to **Yes**.



- Recovered – Select if the insurance company gets the money back. Most commonly used when a disbursement transaction is reversed or recycled.

- **Do Reverse Accounting** – Check if you need to capture the accounting information when reversing the disbursement activity. This is usually always selected, as collecting information for reverse accounting is necessary.

Accounting Detail Reversal

If the Do Reversal Accounting check box is selected, then when a reversal is processed it will be viewable via the Accounting Detail (Reversal) section. This applies when a disbursement activity was reversed or recycled. The reversal accounting status is not viewable via the OIPA application. You will need to query the database. If the status is pending, you will not see the Accounting Detail (Reversal) section at all. It only displays when the status is active.

Chart Of Accounts

Steps

1. New Chart Of Account
2. Chart Of Accounts Entity
- 3. Chart Of Accounts Entry**
4. Chart Of Accounts Criteria
5. Chart Of Accounts Result

Chart Of Accounts Entry

Account Number Format :

Effective From Date :

Effective To Date :

Entry Description :

Credit/Debit :

→ Type :

→ Gain/Loss Flip On Negative

Accounting Amount :

Fund Type :

Transaction Money Types :

→ Original Disbursement Status :

→ Do Reversal Accounting?

Link Suspense

Disbursement Fields on CoA Wizard Step Three

You are here: Admin Explorer > Chart of Accounts > Create Chart of Accounts

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create Chart of Accounts

Chart of Accounts is set-up from the **Admin Explorer** tab. A wizard will walk through the five steps required to set-up an account in a company's CoA.

The five steps are:

1. [Associate account to a company.](#)
2. [Establish the entity.](#)
3. [Enter account processing information.](#)
4. Specify additional [Chart of Accounts criteria](#) (this is optional).
5. Specify [results](#) (this is optional).

You are here: Admin Explorer > Chart of Accounts > CoA Step 1 and 2

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

CoA Creation: Step One & Step Two

Chart of Accounts is set-up from the **Admin Explorer** tab. The Chart of Accounts wizard steps through the process for adding new accounts. After completing the steps the hierarchy folder structure is created for an account. Modify account information by right-clicking on the applicable folder and selecting **Edit Chart of Accounts**.

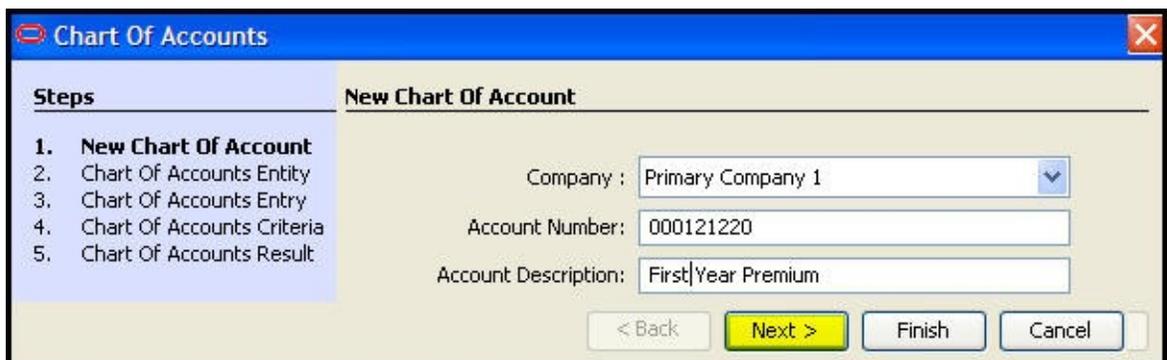
 The CoA Wizard moves through all five steps to set-up an account. Steps four and five can be skipped and added at a later time.

1. Right-click on the Chart of Accounts folder from the Admin Explorer tab and select **New Chart of Accounts**.



Chart of Accounts Right-Click Option

2. Select the **Company** that the account is associated with.
3. Enter the **Account Number**.
4. Enter the **Account Description**.
5. Select **Next**.

A screenshot of the 'Chart Of Accounts' wizard dialog box. The title bar says 'Chart Of Accounts'. On the left, a 'Steps' pane shows five steps, with '1. New Chart Of Account' selected. The main area is titled 'New Chart Of Account' and contains three input fields: 'Company' (a dropdown menu with 'Primary Company 1' selected), 'Account Number' (a text box with '000121220'), and 'Account Description' (a text box with 'First Year Premium'). At the bottom, there are four buttons: '< Back', 'Next >' (highlighted in yellow), 'Finish', and 'Cancel'.

Step One of CoA Wizard

6. Select either an associated **Transaction** or check **Suspense**.
7. Select **Next**.

Refer to the [CoA Step Three](#) section for information on entering the account Entry information.

The screenshot shows a software window titled "Chart Of Accounts" with a "Steps" sidebar on the left. The sidebar lists five steps: 1. New Chart Of Account, 2. **Chart Of Accounts Entity**, 3. Chart Of Accounts Entry, 4. Chart Of Accounts Criteria, and 5. Chart Of Accounts Result. The main area is titled "Chart Of Accounts Entity" and contains a "Transaction:" dropdown menu with "Anniversary" selected and a "Suspense:" checkbox which is unchecked. A red box highlights the "Suspense:" checkbox with the text: "You can only select a transaction or check Suspense. You can't have both." The "Next >" button is highlighted in yellow.

Step Two of CoA Wizard

You are here: Admin Explorer > Chart of Accounts > CoA Step 3

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

CoA Creation: Step Three

The Chart of Accounts entry fields that must be populated will vary depending on the type of account being created. The fields on this screen of the wizard are not configurable since they are needed by the system to perform account processing.

The required fields that must be populated regardless of the account type are:

- Effective From Date
- Entry Description
- Type

The rest of the fields are for additional information about the account, depending on the type of account being created. The values of the fields on this screen of the wizard are saved in **AsChartOfAccountsEntry** table. When a transaction or suspense generates an accounting record in **AsAccountingDetail** it is associated to this table via the **ChartofAccountsEntryGUID**. The **AsChartOfAccountsEntry** is associated to the Entity via the **ChartofAccountsEntityGUID** in the **AsChartOfAccountsEntity** table, which is associated to the account via the **ChartOfAccountsGUID** in the **AsChartOfAccounts** table.

Refer to the [Chart of Accounts Relationship Diagram](#) and the [Chart of Accounts Entry Database Definition](#) for additional information.

Steps for CoA Entry

1. Enter the necessary entry information according to the descriptions given below.
2. Click **Next** to move on to [Step 4 to add Criteria](#) or **Finish** to complete the account set-up.

Entry Field Descriptions

- **Account Number Format:** This is not supported in this release.
- **Effective From Date:** This date signals that the account can start to use the Account Entry detail for accounting. This information is matched to the activity's effective date.



The effective date of the entry must be the same as or prior to the transaction's effective date in order for accounting to be performed.

- **Effective To Date:** The last day accounting can be performed using the AccountEntry detail record of this account. This information is matched to the activity's effective date.
- **Entry Description:** Meaningful description of the account and the entry record details.
- **Debit/Credit:** Select whether the account is a debit or credit. Populated by AsCodeCoAType.



A debit or credit does not mean a decrease or increase. Debits and credits are recorded as positive numbers. The double entry system in a company's general ledger requires each transaction to affect at least two entries, creating a balance within the system. If one account is debited then another must be credited, allowing for a total of all debits and credits to be equal. Each account identified in the CoA has two columns: one for debits and one for credits. The type of account and the column the entry is posted in determines if it is a decrease or an increase. The type of account determines whether a debit or a credit is a loss or gain.

- **Type:** In step two of the wizard, the suspense box was checked or a transaction was associated with the account. If suspense was checked then **Suspense Amount** will be pre-populated in this field by default. Also available to the Suspense accounting is Suspense Field Amount. If a transaction was associated with the account then select either MathVariable, ByFund, TotalOfFunds or Disbursement.
 - **MathVariable:** Select **MathVariable** when the amount value is in a math variable in the transaction's configuration. Enter the exact name of the math variable in the Accounting Amount field. Accounting is performed on the value of this variable. This is used for general accounting purposes only. There is no sub-account accounting generated on this value.
 - **Suspense Field Amount:** Select Suspense Field Amount when the amount value is in a field on the Suspense Screen configuration. Enter the exact name of the field in the Accounting Amount field. Accounting is performed on the value of this field. This is used for general accounting purposes only. There is no sub-account accounting generated on this value.



COA entries that are based on suspense fields, math variables or non-fund specific allocations (general accounting) will use values in the transaction math and these will write with the first successful execution of the activity (when there are no non-overridden business rule errors).

- **ByFund:** Select ByFund to generate sub-account accounting (fund accounting). When selected the **Gain/Loss, Flip On Negative** and **Transaction Money Types** fields are applicable for Fund type entries. The number of accounting records created for this type depends on the number of funds that a specific money type affects in

this activity or save of the suspense record and the value of **Fund Type**.



When GenerateAccounting is set to YES, if the accounting entries are based on fund (separate accounting), then they will wait until all AsValuation records for the activity are moved into an active state and there are no non-overridden business rule errors generated from the PostAssignmentValidateExpressions rule. Even though the valuation records may be active before the activity moves into an active state, the accounting will not be written if there are validation errors in the PostAssignmentValidateExpressions. If there are no PostAssignmentValidateExpressions validation errors and all of the valuation records are in an active state, even if the activity is still in an NUV Pending status, the accounting records for each fund can be written. When there are subsequent executions of the activity, the accounting will run again but will check to see if fund accounting records already exist for the activity.

Example:

Account Detail record shows **Accounting Type** is ByFund and **Fund Type** is Variable. A policy that has five funds, two fixed and three variable, will then have up to three records created for accounting purposes, depending on the variable funds that are affected by the specific money movement of the associated value. If the **Gain/Loss** checkbox is checked, accounting is generated on the gain or loss experienced by the company because of the money movement in the affected variable funds.



The Assignment tag must be present in the transaction if ByFund is used.

- **TotalofFunds**: Select **TotalOfFunds** when all funds will be affected. When selected the **Gain/Loss, Flip On Negative** checkboxes and Transaction Money Types are enabled. Sub-account accounting is not generated.
- **Disbursements**: Select Disbursements when the activity performs a disbursement. When selected the **Original Disbursement Status, Do Reversal Accounting, Forward Accounting Status** and **Reversal Accounting Status** are enabled. The **Original Disbursement Status** could be pending, recovered or recoverable. Depending on the status of the disbursement at the time the activity executes, the accounting is generated using the appropriate Chart of Account entry record.
 - **Scenario 1**: Create three Chart of Account Entries, one for each of the Original Disbursement Statuses. The first time the Disbursement activity is processed, the accounting will be performed using the Chart of Account Entry that has the Original Disbursement Status set as Pending. If the parent activity is reversed or recycled, when the disbursement is in an active status, then the accounting will be done using the COAEntry that has the Original Disbursement Status of **Recovered**.
 - **Scenario 2**: Make sure the <Disbursement> tag of the Disbursement transaction has the attribute RECOVERABLE set to **Yes**. Processing a Disbursement activity will perform the accounting using the Chart of Account Entry record that has the Original Disbursement Status of Recoverable.
- **Gain/Loss**: Check if type is either **ByFund** or **Total Of Funds**. When checked accounting is performed on the Gain/Loss amount for all of the funds.

- **Flip On Negative:** Check if type is **ByFund** or **Total Of Funds**. When checked and the amount that accounting should be performed on is Negative, then Credit/Debit records are flipped.



Only select either **Gain/Loss** or **Flip on Negative**. They are mutually exclusive.

- **Accounting Amount:** Only use when the type is either math variable or disbursement. Enter the name of a math variable that is in the associated transaction on which the accounting should be performed. DisbursementAmount will be auto-populated if it is a disbursement type entry.
- **Fund Type:** Fund type that accounting should be performed for. Populated by AsCodeFundType.
- **Transaction Money Types:** Only enabled when the type is **ByFund** or **Total Of Funds**. More than one money type can be selected. The money type specified is used as a filter to the activity's valuation records. Accounting is generated for that account only when the money type in the valuation record matches with the money type selected for that account. This list box is populated by AsCodeMoneyType.
- **Original Disbursement Status:** Only enabled when type is **disbursement**. This field maps back to the database column AsChartOfAccountsEntry.OriginalDisbursementStatusCode. This value is used to make adjustments to AsDisbursement records that have a matching status and are still applicable (not already adjusted to zero).
- **Do Reversal Accounting:** Check this if accounting information

needs to be captured when reversing the activity. This is usually always selected, as collecting information for reverse accounting is necessary. When checked, the reversal of the activity will generate the exact opposite accounting from the accounting records previously generated by the activity.



If the Do Reversal Accounting check box is selected and a reversal is processed then it will be viewable via Accounting Detail from the results of the active reversal of the activity.

- **LinkSuspense:** Check to link a transaction to a suspense record. Both the transaction (activity in OIPA) and Suspense records will be connected to the AsAccountingDetail record and will be visible in OIPA on the Accounting Detail screen of a Suspense record.

You are here: Admin Explorer > Chart of Accounts > CoA Step 4

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

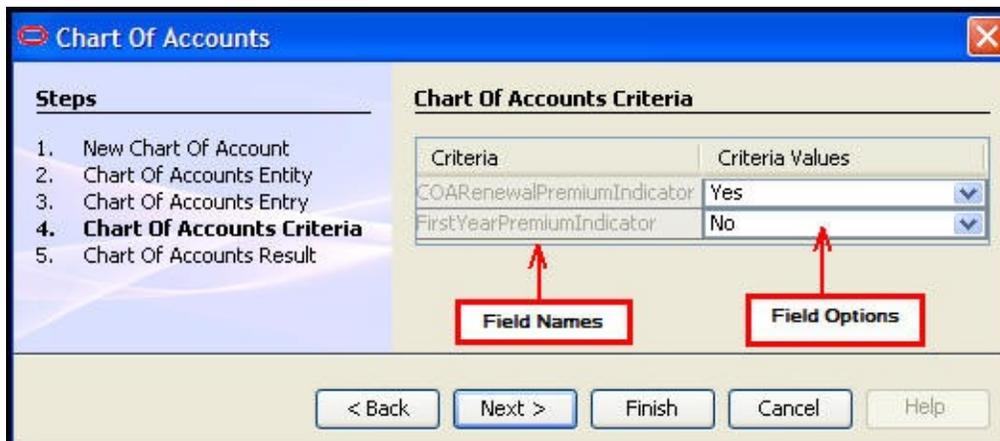
CoA Creation: Step Four

The criteria section is where filters identify values from the policy or the activity that must be matched in order to generate the chart of account entry. This should be used when there are multiple chart of account entries for an account.

This section is configured using two business rules. The first is the ChartOfAccountsScreen business rule, which configures the actual (dynamic) fields displayed on the screen. The second is the ChartOfAccountsSpecifications business rule, which is configured with the criteria logic that if met identifies the account to be used for accounting. Both rules need to be configured prior to saving account entry details that will need to use criteria logic. These two rules are described in the [Steps to setup Chart Of Accounts Criteria](#) section.

Steps to Select Criteria

1. Select the criteria values from the drop down boxes that must match to a value on the policy or in the activity as configured in ChartOfAccountsSpecifications in order for the entry to be used for accounting.
2. Click **Next** to advance to [Step Five of the CoA Creation Wizard](#).

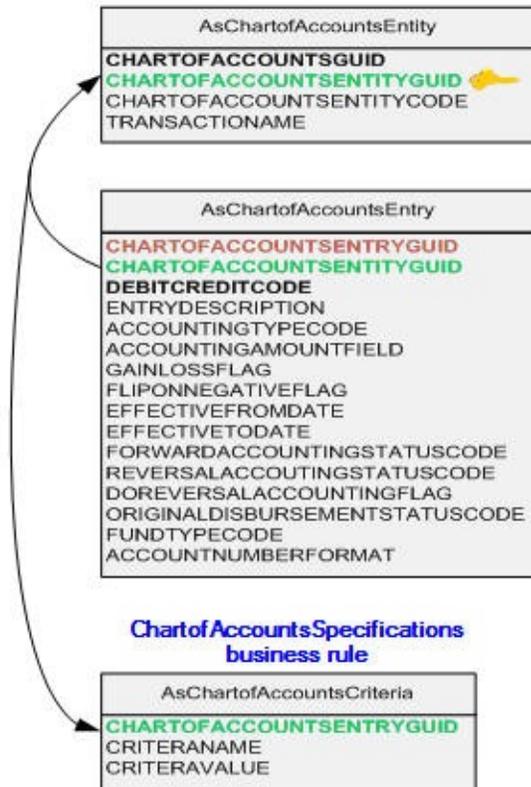


The screenshot shows a window titled "Chart Of Accounts" with a "Steps" sidebar on the left and a "Chart Of Accounts Criteria" main area. The sidebar lists five steps, with the fourth step, "Chart Of Accounts Criteria", highlighted. The main area contains a table with two columns: "Criteria" and "Criteria Values". The table has two rows: "COARenewalPremiumIndicator" with a value of "Yes" and "FirstYearPremiumIndicator" with a value of "No". Red boxes labeled "Field Names" and "Field Options" are positioned below the table, with red arrows pointing to the "Criteria" and "Criteria Values" columns respectively. At the bottom of the window are five buttons: "< Back", "Next >", "Finish", "Cancel", and "Help".

Criteria	Criteria Values
COARenewalPremiumIndicator	Yes
FirstYearPremiumIndicator	No

Step Four of the CoA Wizard

Criteria Database Tables



You are here: Admin Explorer > Chart of Accounts > CoA Step 5

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

CoA Creation: Step Five

The **Results** section of the Chart of Accounts screen is used to capture additional client specific information needed for various accounting purposes. By checking a result when a transaction or suspense is processed the extra result information is written to the **AsAccountingDetailField** table, which is linked to the **AsAccountingDetail** table via the **AccountingDetailGUID**. The value to write comes from the transaction or suspense configuration and must match exactly what is configured for the results section.

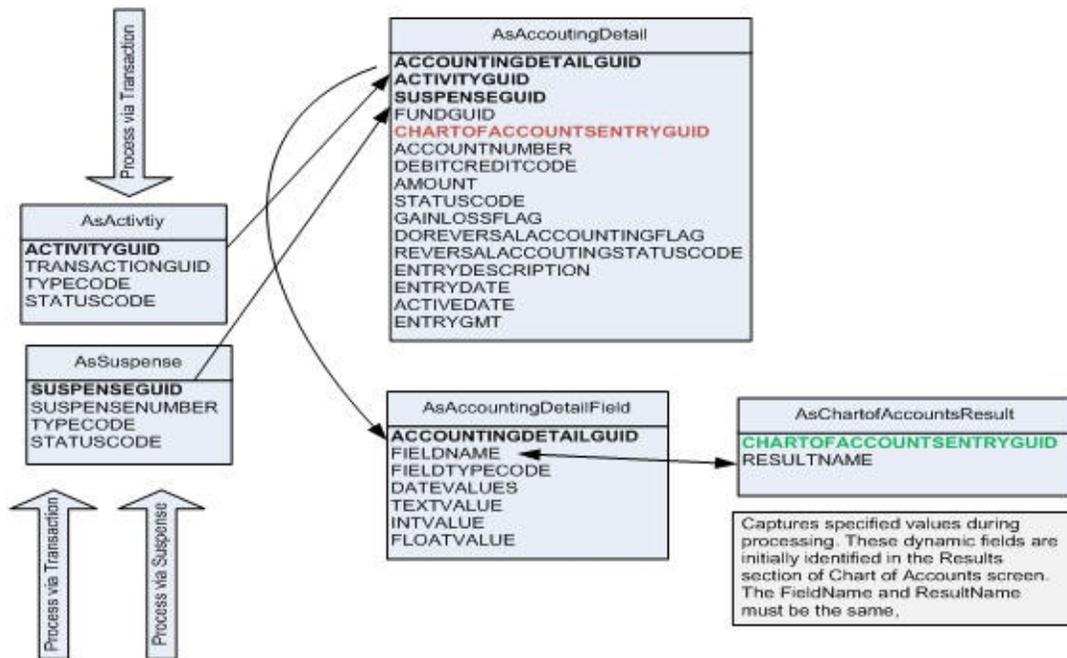
The Results section on the Chart of Accounts screen is configured via the **ChartofAccountsResults** rule. See the [Steps to Set-up Chart Of Accounts Results](#) for details on the configuration of the Results section.

Steps to Select Results

1. Click the checkbox next to any results that should be captured.
2. Click **Finish**.



Step Five of CoA Wizard



CoA Database Tables with Result Information

You are here: Admin Explorer > Chart of Accounts > Steps to Set-up CoA Criteria

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Steps to Set-up CoA Criteria

In order to configure criteria that identifies the correct chart of account entry for suspense or an activity, two rules must be configured. The two rules are as follows:

- **ChartOfAccountsScreen:** This rule is used to set-up the dynamic fields that may be used to select an account entry for accounting generation. It is located in the Global Rules Explorer tab under **Business Rules | Screen Rules**.
- **ChartOfAccountsSpecifications:** This rule is used to identify which of the dynamic field values must match values from the policy or activity math. It is located in the Global Rules Explorer tab under **Business Rules | System Rules**.

ChartOfAccountsScreen Rule

ChartofAccountsScreen business rule can be used to configure dynamic fields and validation for the criteria section in [Step Four of the Chart of Accounts wizard](#). These fields can be configured for both suspense and transaction accounts. The criteria is client specific field information used to identify the correct account entry to which accounting would be written when there are multiple applicable accounts or any conditional generation of an account entry.

! This rule ONLY configures the fields. The ChartofAccountsSpecifications rule is configured to specify the matching logic that will allow the account entry to generate accounting. See the [ChartOfAccountsSpecifications](#) business rule section for further information.

Criteria	Criteria Values
COARenewalPremiumIndicator	Yes
FirstYearPremiumIndicator	No

Step Four of CoA Wizard

ChartofAccountsScreen XML Definition

<ChartOfAccountsScreen> Opening/Closing tag

<TransactionAccounting> Indicates fields for transactions

<Fields> Fields section that will be used as criteria to match a debit/credit on an account

<Field> Standard field tag

<Name> Name of field

<Display> The field's label as it should look on the screen

<DataType> Standard field datatype (this is almost always a combo box)

<Query> Standard query tag for combo boxes and radio buttons

<SuspenseAccounting> Indicates fields for suspense

<Fields> Fields that will be used as criteria to match a debit/credit on an account

<Field> Standard field tag

<Name> Name of field

<Display> Display of field

<DataType> Standard field datatype (this is almost always a combo box)

<Query> Standard query tag for combo boxes and radio buttons

ChartofAccountsSpecifications Rule

This rule is only used in conjunction with the **ChartOfAccountsScreen** rule. The **ChartOfAccountsSpecifications** rule specifies the matching logic between the account entry criteria values and policy or activity math values. The fields that capture the criteria information are configured via the **ChartOfAccountsScreen** rule. If a transaction or suspense account has multiple associated accounts or there is a conditional situation that affects accounting generation, then this rule is configured to identify the criteria match.

ChartofAccountsSpecifications XML Definition

<ChartOfAccountsSpecifications> Opening/Closing tag

<TransactionAccounting> Criteria information for transaction fields

<Criteria> Information about criteria for fields on the ChartOfAccountsScreen rule.

This tag must be configured for each transaction that could use multiple accounts for processing.

- Attribute NAME Exact name of CoA field to match to criteria
- Attribute CATEGORY Specifies where to get a value to match criteria either POLICY or MATH.
 - POLICY looks at AsPolicy and AsPolicyField
 - MATH looks for a MathVariable or field in the transaction.
- Attribute CATEGORYFIELD Name of field or MathVariable to use to match the criteria value. This field or MathVariable must be the exact name found either on the Policy screen or in the transaction's math section.

<SuspenseAccounting> List of criteria information for suspense fields

<Criteria> Information about criteria for a field on the ChartOfAccountsScreen rule

- Attribute NAME Exact name of CoA field to match criteria to
- Attribute CATEGORY Specifies how to get a value to match criteria either SUSPENSE or SQL for either the suspense record or a SQL statement.
- Attribute CATEGORYFIELD Name of field or MathVariable to use to match the criteria value. Only valid if CriteriaCategory is not set to SQL
- Attribute SQL SQL to get value to compare to CoA criteria. Only valid if category is set to SQL.

You are here: Admin Explorer > Chart of Accounts > Steps to Set-up CoA Results

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Steps to Set-up CoA Results

Results are used to store additional accounting information, possibly for sending to a downstream system. The fields and logic for step five of the wizard, which specify results, are configured in the `ChartofAccountsResults` rule. Unlike the [ChartofAccountsSpecifications](#) rule, results only requires the configuration of one rule. Checkboxes are automatically generated on the Chart of Accounts Wizard Step 5 Results screen when configuring a result.

The resulting checkbox indicates when the result value should be captured from the activity math or fields, suspense activity, policy, fund, map group or static value and written to the `AsAccountingDetailField` table. The transaction or suspense must be configured with a matching value if an activity's math or field is captured.

By checking a result when a transaction or suspense is processed, the extra result information is written to the `AsAccountingDetailField` table, which is linked to the `AsAccountingDetail` table via the `AccountingDetailGUID`.

For a complete list of the elements, attributes and values supported in the `ChartofAccountsResults` business rule, refer to the V9 XML Configuration topic in Help. Refer to **Business Rules | System Rules | ChartofAccountsResults**.

```
<ChartOfAccountsResults>¶
<TransactionAccounting>¶
.....<Result-NAME="PremiumType".DISPLAY="Premium-Type".CATEGORY="MATH".
CATEGORYFIELD="PremiumType">¶
.....</Result>¶
</TransactionAccounting>¶
<SuspenseAccounting>¶
.....<Result-NAME="CollectionType".DISPLAY="Premium-Type".CATEGORY="SUSPENSE".
CATEGORYFIELD="CollectionType">¶
.....</Result>¶
</SuspenseAccounting>¶
</ChartOfAccountsResults>¶
```

CoA Results XML Example

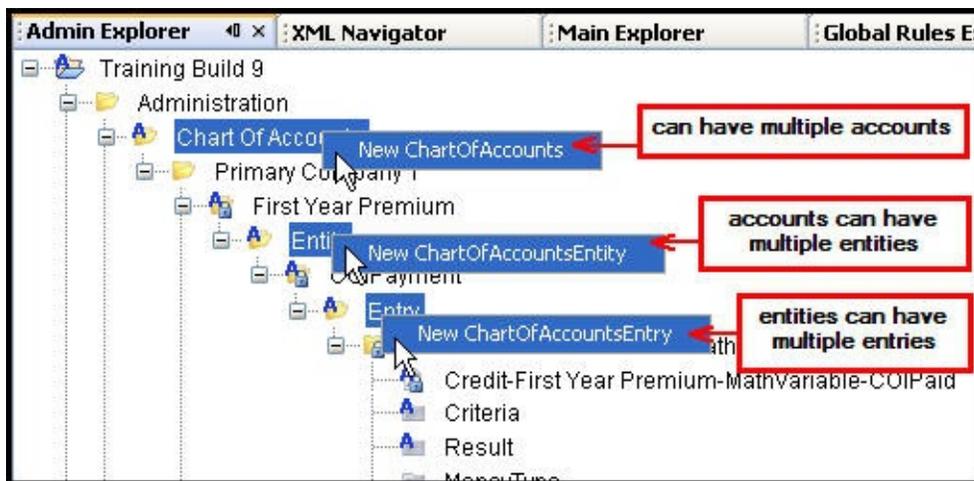
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Admin Explorer > Chart of Accounts > Multiple Entries for an Account

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Multiple Entries for an Account

After the initial account is set-up via the wizard, additional Chart of Accounts entries can be added by right-clicking on the Entry folder and selecting **New ChartofAccountsEntry**. This creates a one to many relationship for a transaction to accounts. Set-up [criteria logic](#) if accounting is only performed when specific criteria are met. Each account that is saved is available in the Entries folder and can be viewed and/or modified.



CoA Accounts and Entries in Admin Explorer

You are here: Admin Explorer > Chart of Accounts > Query Account Records

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Query Account Records

In the database use the ActivityGUID to locate the record written to AsAccountingDetail. Locate the ActivityGUID by running the query listed below.

Query To Locate the ActivityGUID

```
SELECT AsActivity.ActivityGUID
FROM AsActivity
JOIN AsTransaction ON
AsTransaction.TransactionGUID=AsActivity.TransactionGUID
JOIN AsPolicy ON AsPolicy.PolicyGUID=AsActivity.PolicyGUID
WHERE AsPolicy.PolicyNumber='[PolicyNumber]'
AND AsTransaction.TransactionName='[TransactionName]'
AND AsActivity.EffectiveDate='[EffectiveDate]'
AND AsActivity.TypeCode IN ('01','04')
AND AsActivity.StatusCode = '01'
```

View the accounting details for all transactions associated with a policy or activity by using the following SQL statements.

Query to Use ActivityGUID to Locate Record Written to AsAccountingDetail

```
SELECT AsAccountingDetail.*
FROM AsAccountingDetail
JOIN AsActivity ON AsActivity.ActivityGUID =
AsAccountingDetail.ActivityGUID
AND AsActivity.StatusCode = '01'
JOIN AsPolicy ON AsPolicy.PolicyGUID = AsActivity.PolicyGUID
AND AsPolicy.PolicyNumber = '[policy number]'
```

Query to View all CoA Information

```
SELECT * FROM AsChartOfAccounts
JOIN AsChartOfAccountsEntity ON
AsChartOfAccounts.ChartOfAccountsGUID =
AsChartOfAccountsEntity.ChartOfAccountsGUID
```

JOIN AsChartOfAccountsEntry ON
AsChartOfAccountsEntity.ChartOfAccountsEntityGUID =
AsChartOfAccountsEntry.ChartOfAccountsEntityGUID
LEFT JOIN AsChartOfAccountsMoneyType ON
AsChartOfAccountsEntry.ChartOfAccountsEntryGUID =
AsChartOfAccountsMoneyType.ChartOfAccountsEntryGUID
LEFT JOIN AsChartOfAccountsResult ON
AsChartOfAccountsEntry.ChartOfAccountsEntryGUID =
AsChartOfAccountsResult.ChartOfAccountsEntryGUID
LEFT JOIN AsChartOfAccountsCriteria ON
AsChartOfAccountsEntry.ChartOfAccountsEntryGUID =
AsChartOfAccountsCriteria.ChartOfAccountsEntryGUID

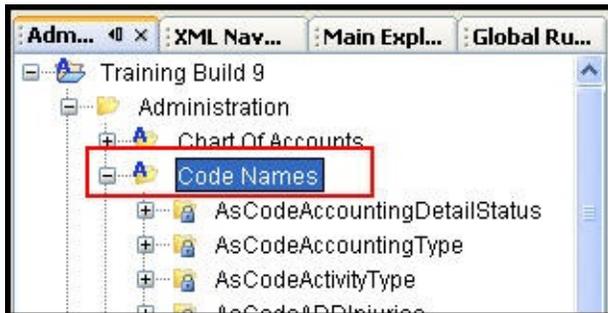
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Admin Explorer > Chart of Accounts > Chart of Account Code Tables

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Chart of Account Code Tables

The following codes are used for Chart of Accounts and can be found in the **Admin Explorer** window in the [Code Names](#) folder.



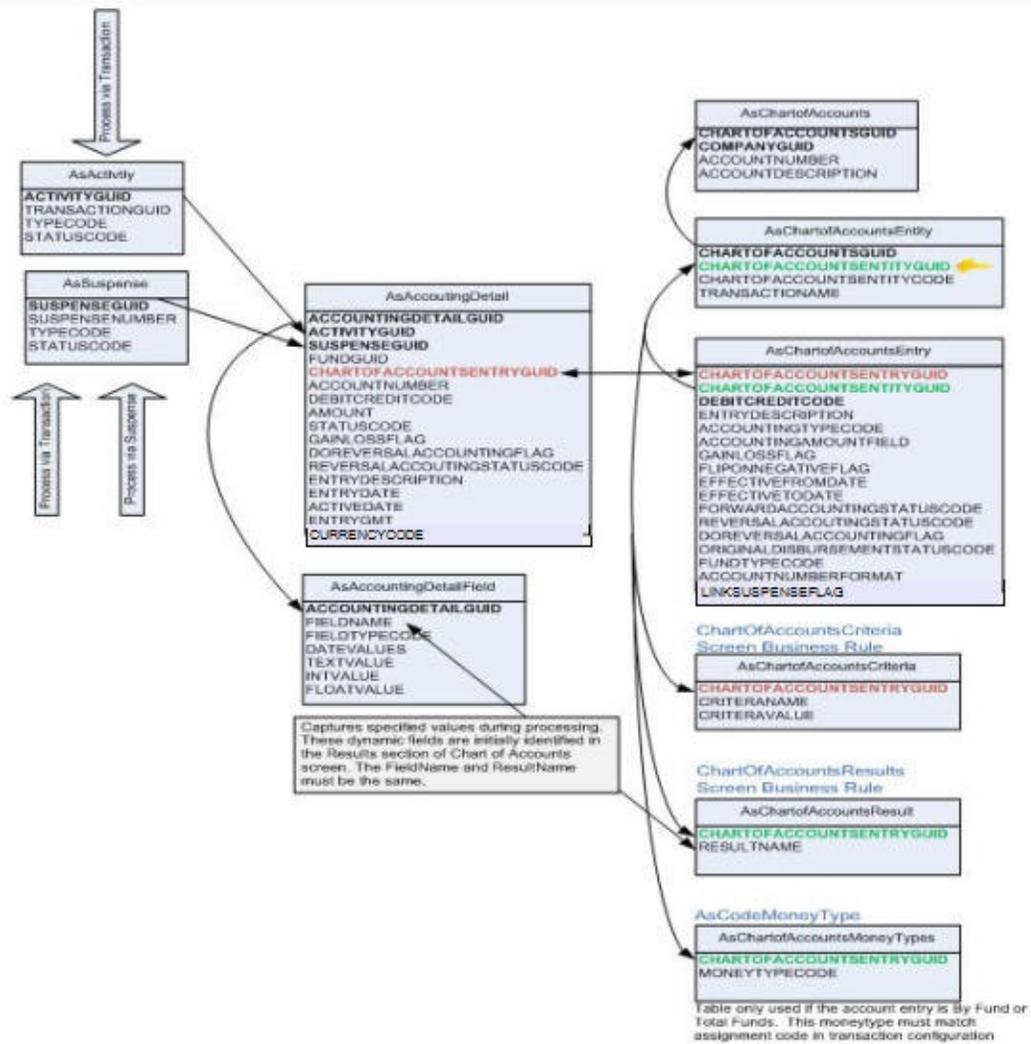
Code Names Folder in Admin Explorer

Code Type	Description
AsCodeDebitCredit	Codes in this table are used on the Account Entry Screen in the Credit/Debit field.
AsCodeMoneyType	Codes in this table are used on the Account Entry Screen in the Transaction Money Types combo field.
AsCodeAccountingDetailStatus	Codes in this table are used on the Account Entry Screen in the Forward Accounting Status and Reversal Accounting Status drop-down box.
AsCodeAccountingType	Codes in this table are used on the Account Entry Screen in the Type drop-down box.
AsCodeFundType	Codes in this table are used on the Account Entry Screen in the Fund Type drop-down box.

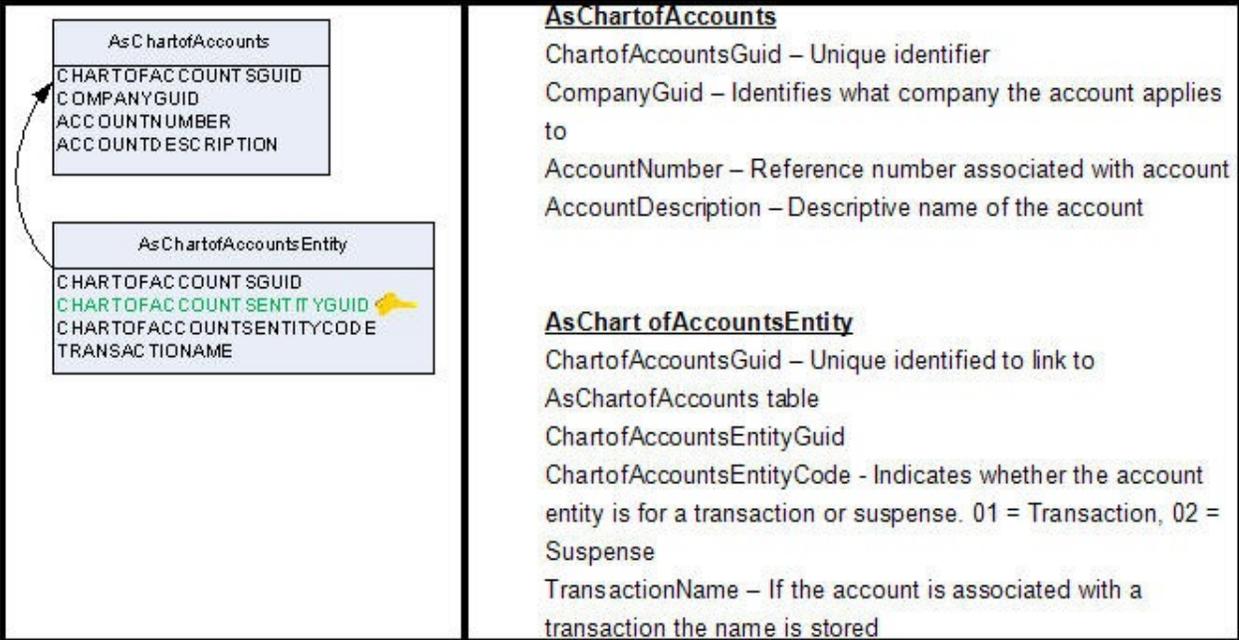
You are here: Admin Explorer > Chart of Accounts > Database Tables

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Chart of Accounts Relationship Diagram



AsChartofAccounts and AsChartofAccountsEntity Tables



AsChartofAccounts

ChartofAccountsGuid – Unique identifier
CompanyGuid – Identifies what company the account applies to
AccountNumber – Reference number associated with account
AccountDescription – Descriptive name of the account

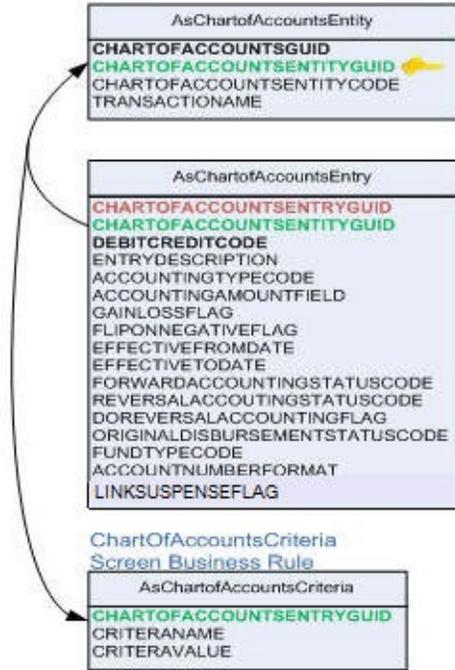
AsChart ofAccountsEntity

ChartofAccountsGuid – Unique identified to link to AsChartofAccounts table
ChartofAccountsEntityGuid
ChartofAccountsEntityCode - Indicates whether the account entity is for a transaction or suspense. 01 = Transaction, 02 = Suspense
TransactionName – If the account is associated with a transaction the name is stored

AsChartofAccountsEntry Table

The values of these fields are saved in the [AsChartOfAccountsEntry table](#) and provide detail for an account. When a transaction or suspense generates an accounting record in AsAccountingDetail it is associated to this table via the ChartofAccountsEntryGUID and provides further information regarding the account. The AsChartOfAccountsEntry is associated to the account via the ChartofAccountsEntityGUID in the AsChartOfAccountsEntity table.

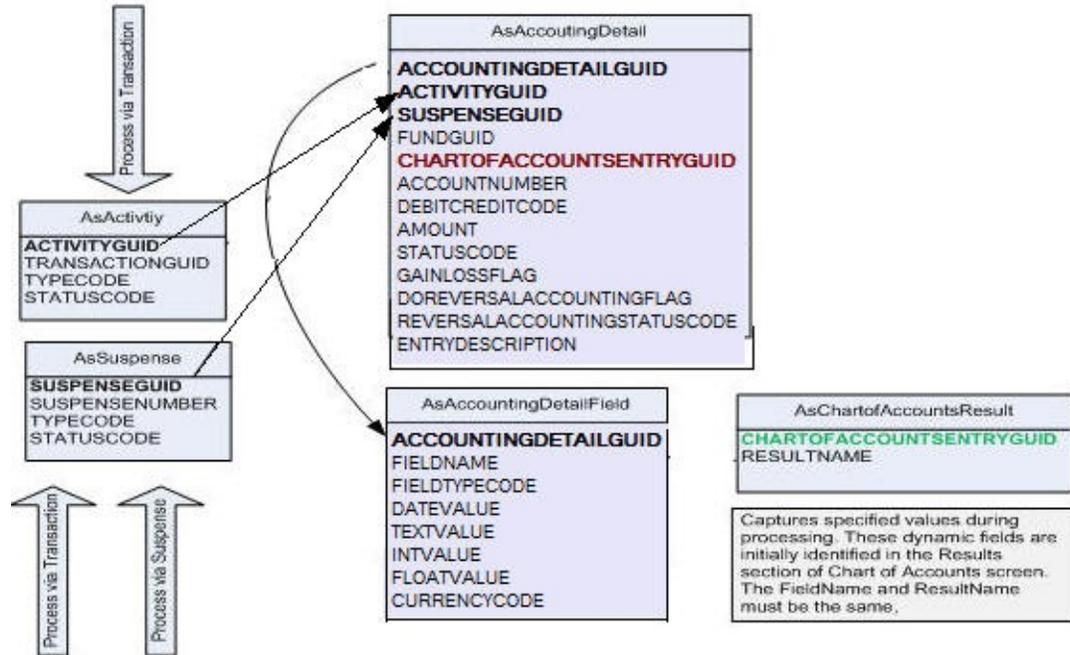
Criteria Database Tables



Disbursement Table

--

Results Database Tables



You are here: Admin Explorer > Chart of Accounts > View Accounting in OIPA

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

View Accounting in OIPA

The Chart of Accounts in OIPA performs accounting when:

- a suspense record is created
- a transaction that has financial implication is processed.

When the transaction is processed, transaction accounting is performed on the associated accounts. View the accounting details by clicking on the icon to the left of the processed activity. This will open the Activity Details window. Select the **Accounting** tab to view the accounting details.

The Accounting Details window displays the various accounts affected by that transaction, along with their Account Numbers, Description, Amount and Debit/Credit details. This information displays in two tables: one with details for forward accounting and one with details for reversal accounting. SQL statements can also be run to view accounting for specific policies, account numbers, etc. Please see [Querying Account Records](#) for additional detail on SQL statements. The below is the example for the tables mentioned for forward and reversal accounting.

ICOIPAYMENT (1/29/2012) - GLPT31035007

Detail : Entry Fields : Allocations : Math : Spawn : Summary : **Accounting** : Class Membership
Valuation : Disbursement : Suspense :

ACCOUNTING DETAIL			
Account Number	Description	Debit/Credit	Amount
First Year Premium	First Year Premium	Credit	3,270.00 USD
Policy Suspense	Policy Suspense	Debit	3,270.00 USD

You are here: Admin Explorer > Code Names

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

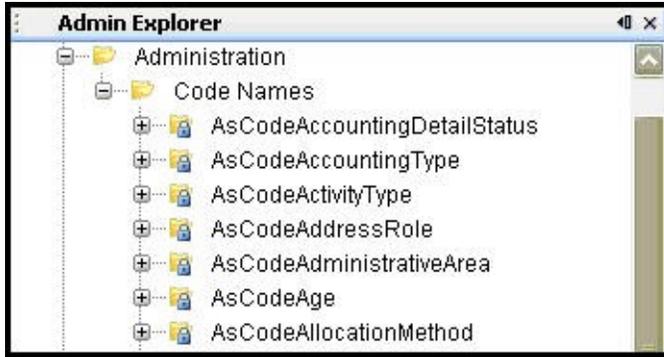
Code Names

Codes are used for consistency, performance and flexibility purposes. Codes are two digit numeric values, which are stored in the **Code Names** folder. The **Code Names** folder has subfolders that store the various codes according to their purpose. For example, all the codes for roles on a policy are stored in the folder called AsCodeRole.

Certain rules have elements that specifically use the codes found in the **Code Names** folder. By using numeric codes instead of names when configuring, updating becomes simple. Change the code name and every instance of that code name will be updated. Using two characters also saves performance time and space in the database.

The Rules Palette provides drop-down boxes labeled TYPE with look-up links that display the codes associated with the task being configured. For example, roles on a policy such as insured, beneficiary, etc., each have a code in the **Code Names** folder under AsCodeRoles. When configuring for an owner, use the associated code of 01. If the company decides to rename the owner role to policy owner, then this will only need to be done in the Codes Name folder instead of everywhere it was used in the configuration.

The Rules Palette provides dynamic code editing capabilities and the ability to [add](#), [edit](#) or [delete](#) the code values that are stored in the AsCode table. This functionality is accessible via the **Admin Explorer** window in the **Administration** folder. Only users with the proper security roles can update the codes and make modifications using the code-editing functionality. Security roles will be determined by the Security Manager. Refer to the section on [Security](#) for more information on assigning security roles.



Code Names Folder in Admin Explorer



When a code is checked-in, none of the code-editing functionality is enabled. [Check-out](#) the code from the Admin Explorer Code Names folder in order to add/edit codes.

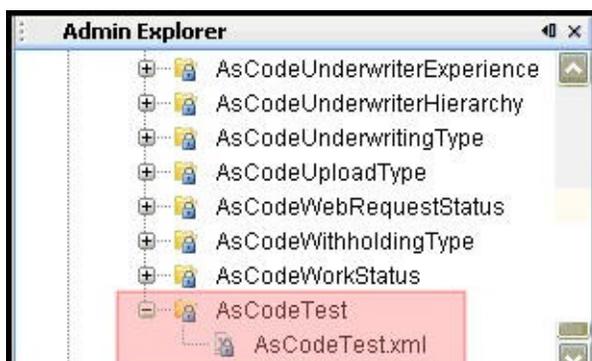
Steps to Add a New Code Name

1. Open **Admin Explorer** and expand the **Administration** folder.
2. Right-click on the **Code Names** subfolder.
3. Select **New Code Name**.
4. Enter the new code name, prefixed with **AsCode**. The system will not save a code without the AsCode prefix in the code name. Once the code is entered correctly, the Finish button becomes enabled.
5. Click **Finish**.



New Code Wizard

By default, the new code name will be checked-in. [Check it out](#) in order to edit it.



New Code Displayed Under Code Names Node

Editing Codes

The following options are available in the Codes pane of any existing code name and will also be available for any new code names added. The functionality will become enabled as soon as the code is checked-out.



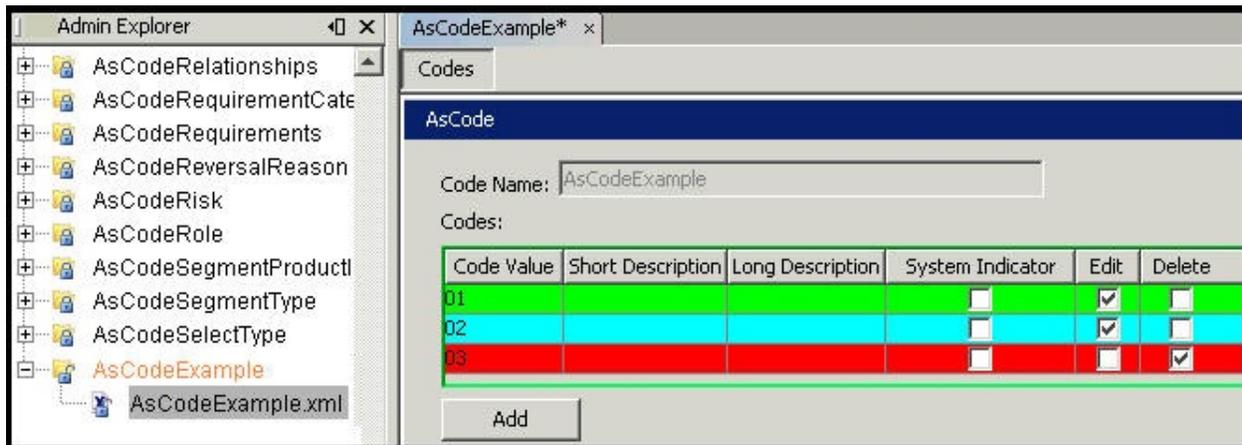
When code names and values are added, a Translation window will display upon Check-in. At least one translation must be added for each code value.

Option	Description
Code Value	The numeric value of the code.
Short Description	A short description of the code.
Long Description	A longer, more detailed description of the code.
System Indicator	This denotes a special type of system code. Default for this option is unchecked.
Edit	Selecting this checkbox allows a code to be edited within a particular code name. When a new code is added, the edit checkbox defaults to checked and the code will appear highlighted in blue. If the Edit checkbox is selected on an existing (not new) code, it will become highlighted in green.
Delete	Selecting this checkbox allows a code to be deleted from the particular code name that is checked out. This code will become highlighted in red when the checkbox is selected.
Add	The Add button allows a new code to be added to the particular code name that is checked-out. When a new code is added, a new code line will appear in the codes pane and it will be highlighted in blue with the Edit checkbox defaulting to checked.

Color Coding

The code-editing functionality uses the following color coding (highlighting) to identify the status of a particular code in a code name.

Code	Task
Green	Edit an existing code (not new).
Blue	Add a new code and edit while new.
Red	Delete a code.



Code Color Scheme

Adding, Editing and Deleting Codes in a Code Name

Once a code name is checked out and the Codes Editor functionality becomes enabled, begin the updates to the codes on the Codes pane. Keep in mind that editing or deleting existing code could adversely affect anything in the system that needs to access the code.

- Clicking **Add** on the Codes pane will add a new blank row (code) and the new code appears highlighted in blue.
- Double-click in the Code Value, Short Description and Long Description fields for the code in order to add information.
- Select the Edit and/or Delete checkboxes for that code as desired. Please see editing codes section above for details.

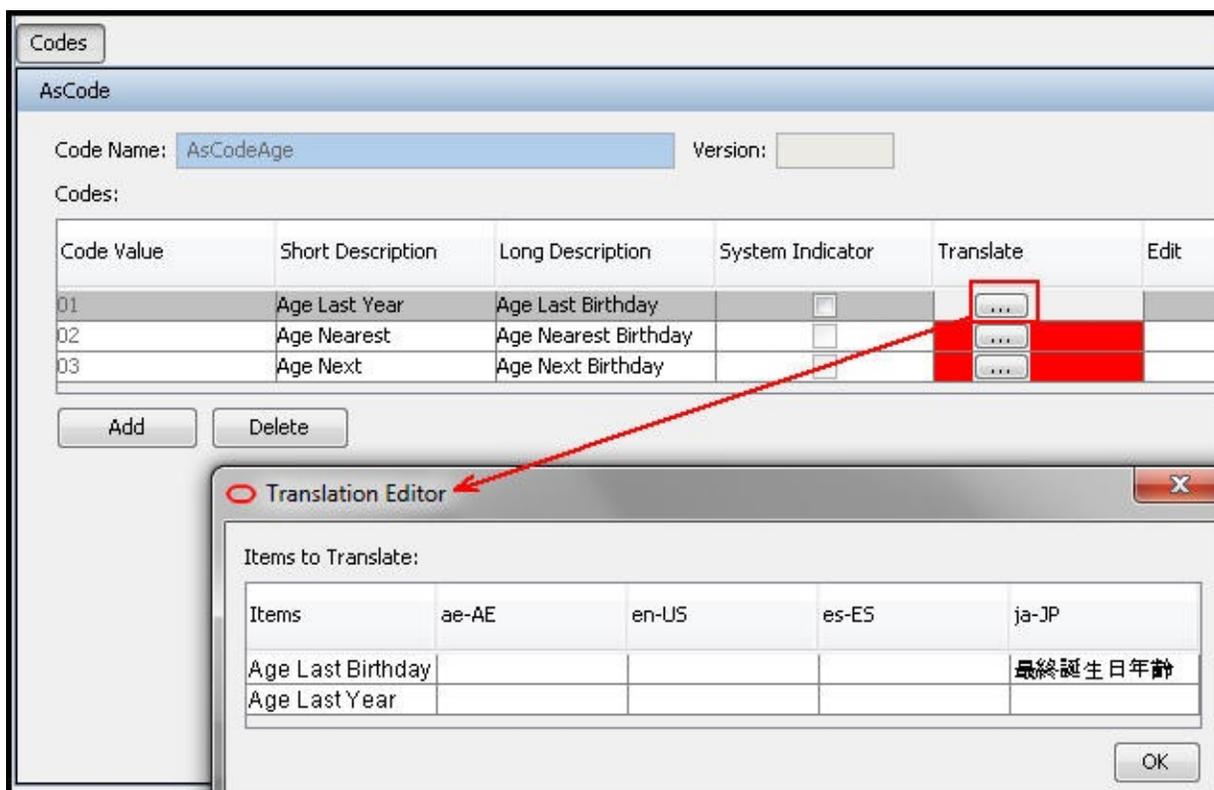
Updating Translation Values when Codes are Edited

A translation button is available on each row to allow easy access to the Translation editor. Click the node in the row to open the translation values for the Short and Long Descriptions of the code. Make any changes and click **OK** to close the window and save the changes.

If changes are checked in for a code in AsCode and the translation node is not clicked, then the Rules Palette will open the Translation Editor window so that translation values can be updated if needed. Update any values and click **OK** to save the changes.



Deleting code values does not remove the translation value.



Translation Button and Translation Editor Window

You are here: Admin Explorer > Comment Templates > Comment Templates

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Comment Templates Overview

Comment templates can be created to define the content of various comments that can then be applied to activities, policies, segments or suspense records in OIPA. Having a template makes it easy for a CSR to add general statements that apply to multiple types of records.

Template Configuration

There are three configuration pieces that are required to support comment templates.

- AsCode must have AsCodeCommentType populated with available types of comments.
- CommentScreen and CommentSearchScreen business rules must be configured.
- [Security privileges](#) must be granted to the user working with templates.

Business Rules

The **CommentScreen** business rule contains a <UseTemplates> element that controls whether comment templates are available in OIPA. This element can be repeated under each of the comment sections to indicate if comment templates are available for that specific type of comment.

There are five types of comments referenced in the CommentScreen business rule.

- <DefaultComments>: configuration in this section will apply to all types of comments, unless specific configuration is present in a comment section below it.
- <PolicyComments>: configuration in this section will apply to policy comments only.
- <ActivityComments>: configuration in this section will apply to activity comments only.
- <SuspenseComments>: configuration in this section will apply to suspense comments only.
- <SegmentComments>: configuration in this section will apply to segment comments only.

Comment templates can be defined at several different levels. Creating an override of the CommentScreen business rule allows you to fine tune

comment templates and control the templates that are available to users in OIPA. The following override levels are available.

- **Plan level:** available for any Policy, Segment or Activity comment under the identified plan.
- **Product level:** available for any Policy, Segment or Activity comments for all plans under the identified Products.
- **Subsidiary Company level:** available for any Policy, Segment or Activity comments for all plans under the identified subsidiary company.
- **Primary Company level:** available for any Policy, Segment, Activity, Suspense or Client record under the identified company.
- **Global level:** available for all comment types regardless of company.

Security

A Rules Palette user must have the [Administration - All Non-Security Administration - CheckIn/CheckOut](#) privilege to work with templates.

Comment Templates Via AsFile

Comment templates can also be defined via AsFile, where AsFile allows inserts into the AsCommentsTemplates. An example of this configuration is provided below.

```
<AsCommentsTemplate>
  <CommentsTemplateGUID>3D8AC8C1-C4F1-4DBF-8068-758A4EA52743</CommentsTemplateGUID>
  <TemplateName>TestTemplate</TemplateName>
  <CompanyGUID>3D8AC8C1-C4F1-4DBF-8068-758A4EA52743</CompanyGUID>
  <PlanGUID>(null)</PlanGUID>
  <TemplateText>This is a test template</TemplateText>
</AsCommentsTemplate>
```

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Admin Explorer > Comment Templates > Create and Edit Comment Template

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

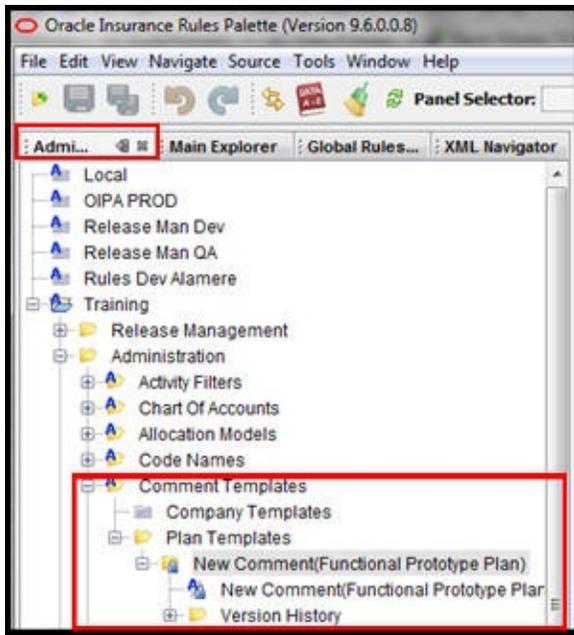
Create and Edit Comment Templates

Comment templates can be created to define the content of various comments that can then be applied to activities, policies, segments or suspense records in OIPA. Having a template makes it easy for a CSR to add general statements that apply to multiple types of records.

Once a comment template is created, it can be edited using the [Template Editor](#).

Steps to Create a Comment Template

1. Navigate to the **Admin Explorer** tab and open **Administration | Comment Templates**.
2. Right-click on **Comment Templates** and select **Add Comments Template**. The New Comments Template wizard will display.
3. Type the name of the template in the **Template Name** field.
4. Select the primary or subsidiary company where the template will be applied.
5. Identify the plan if the comment template can be further defined by plan. This is an optional field.
6. Type the comment message in the Template Text area.
7. Click **Finish**. The new template will appear in the navigation tree in either the Company Template or Plan Template folder.
8. Check-in the template to save it to the database.



Comment Template Navigation in Admin Explorer

Edit Comment Templates

Once a comment template is created, the name and comment text can be edited if needed. The company and plan association cannot be changed. If the template no longer applies to the defined company and plan, it should be deleted.

Steps to Edit a Comment Template

1. Navigate to the Admin Explorer.
2. Open **Administration | Comment Templates**.
3. Open the Company, Product or Plan folder where the template resides.
4. Right-click on the template.xml file and select **Check-out**.
5. Make changes to the template name or the text.
6. Right-click on the template.xml file and select **Check-in**.

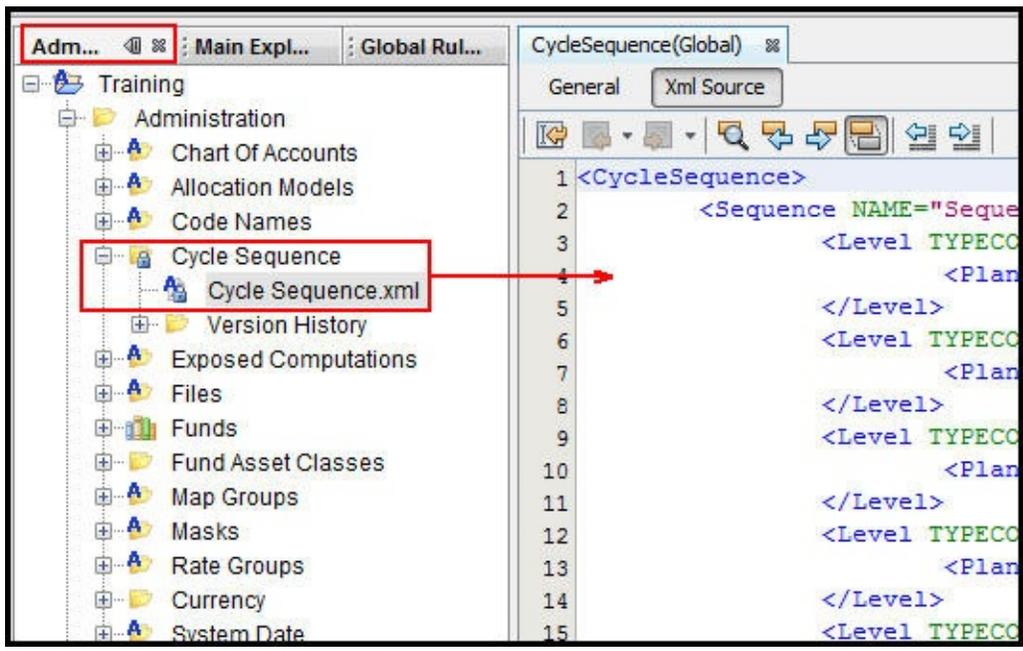
You are here: Admin Explorer > Cycle Sequence

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Cycle Sequence

OIPA supports the ability to run companies and plans through cycle in a predefined order or sequence. Normal day-to-day cycle processing may be defined in one or more of these sequences. Planned business conditions such as month-end or market closures may require an alteration of the normal cycle. Sequences allow the development of alternative execution orders of the plans and/or companies. Occasionally unplanned business conditions exist that require further 'on the fly' alterations of the normal cycle. The client's scheduler can execute individual sequences directly as needs arise and with the individual client's acceptance to such cycle changes.

 In the event of system failure after submitting tasks to a cycle grid, all unprocessed tasks are picked up upon recovery.



Cycle Sequence in Admin Explorer

CycleSequence Business Rule

The Admin Explorer provides an Administration node called **Cycle Sequence**, where cycle sequences are defined. No overrides of this rule are supported and only one version of the Cycle Sequence rule can exist at a time. The TYPECODE attribute for <Level> is defined in AsCodeCycleType. Multiple sequences can be defined in the business rule.

The option to create a new CycleSequence business rule is only available if the rule does not already exist. If the rule exists, it can only be updated or deleted.

Database Table

Cycle sequence information is saved to the database in the AsCycleSequenceProcess table.

Version History

Each time this rule is updated, a version is saved in the Version History folder. Click on a particular version to revert back to that version's configuration. Select two versions at once to use the **Diff** option, which allows differences in the configuration to be easily seen.

Security

Access to this rule is governed by **Administration-All Non-Security Administration-CheckIn/CheckOut** and **Administration-All Non-Security View privileges**. The privileges are assigned to a user in the **Security | Palette Security | Security Role** section of the Admin Explorer. Any user with these security privileges will have access to this business rule. This security should only be assigned to users who are well versed in the area of cycle processing.

Steps to Edit a Cycle Sequence

1. Navigate to **Admin Explorer | Administration | Cycle Sequence**.
2. Open the Cycle Sequence folder, right-click on the file and select **Check out**.
3. Make changes to the sequences. Level TYPECODES are listed in AsCodeCycleType.
4. Save the changes and check in the rule.

You are here: Admin Explorer > Enrollment Transactions

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Enrollment Transactions

Enrollment transactions are specialized transactions used for processing Participant Enrollments, either via Data Intake or OIPA's Enrollment screen. A configurator can use the Enrollment Transactions node in the Admin Explorer window to designate previously-configured transactions for enrollment use—in the Admin Explorer window, navigate to **Administration | Enrollment Transactions**.

Enrollment transactions are created at the Product level, meaning a single enrollment transaction can be shared across multiple plans linked to the same Product.

Creating Enrollment Transactions

Before an enrollment transaction can be created, the following must be true for the environment being used:

- Products are enabled.
- At least one Product is set up.
- At least one Product-level transaction override has been created.

To create a Product enrollment transaction:

1. Navigate to the **Admin Explorer** window.
2. Right-click on the **Enrollment Transaction** node and select **New Enrollment Transaction**. The Enrollment Transaction window will open.
3. The following drop-down fields and their available values will display for selection in the window. Select a value for each field.
 - a. Company—All companies that have been configured
 - b. Product—All Products under the previously selected company
 - c. Plan—All plans under the previously selected Product
 - d. Transaction—All Product-level transactions under the previously selected Product
4. Click **Finish**. The transaction that was just selected will appear under the Enrollment Transactions node, and an enrollment transaction record will be created in the AsEnrollment database table.

Viewing Enrollment Transactions

Right-clicking on a transaction under the Enrollment Transactions node gives two options for viewing an enrollment transaction: **View Details** and **View Transaction**.

- Selecting **View Details** will open the Enrollment Transaction window, which will display the Company, Product, Plan and Transaction that were selected during the Enrollment Transaction's creation.
- Selecting **View Transaction** will open the XML configuration of the transaction.

Deleting Enrollment Transactions

Right-clicking on a transaction under the Enrollment Transactions node also provides the **Delete Enrollment Transaction** option. Selecting **Delete Enrollment Transaction** will open a confirmation window to ensure that the user wishes to delete the transaction. Deleting an enrollment transaction via the Rules Palette will also remove that transaction's record in the AsEnrollment database table.

If a configurator attempts to delete the transaction off of which an existing enrollment transaction is based, an error message will display warning the user that the transaction exists as an enrollment transaction and cannot be deleted. First deleting the enrollment transaction will allow the original transaction to be deleted normally.

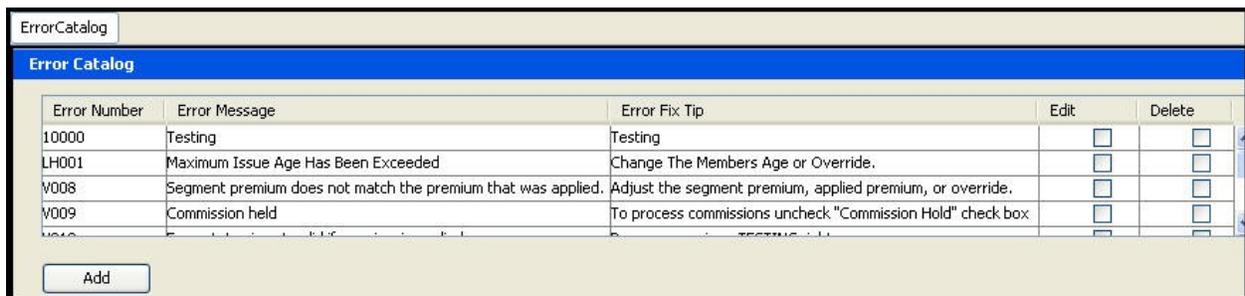
You are here: Admin Explorer > Error Catalog

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Error Catalog

The Error Catalog is a tool for managing error messages in OIPA. Error messages are put into configuration via their error number. The Error Catalog is then used like a dictionary of sorts for quick reference of error messages and their associated error numbers.

The Error Catalog is located in the Admin Explorer tab in the Administration folder. All error messages used in OIPA must be manually entered into the Error Catalog in order to use this administration feature.



Error Number	Error Message	Error Fix Tip	Edit	Delete
10000	Testing	Testing	<input type="checkbox"/>	<input type="checkbox"/>
LH001	Maximum Issue Age Has Been Exceeded	Change The Members Age or Override.	<input type="checkbox"/>	<input type="checkbox"/>
V008	Segment premium does not match the premium that was applied.	Adjust the segment premium, applied premium, or override.	<input type="checkbox"/>	<input type="checkbox"/>
V009	Commission held	To process commissions uncheck "Commission Hold" check box	<input type="checkbox"/>	<input type="checkbox"/>

Add

Error Catalog Editor in Admin Explorer

Steps to Add a New Error Message

1. Navigate to the **Admin Explorer** and open the Administration folder.
2. Scroll down and open the Error Catalog folder.
3. Right-click on the Error catalog file and select **Check-out**.
4. Select **Add** at the bottom of the screen. A blank row highlighted in blue will be added.
5. Enter the Error number. This is the reference used to call the error message.
6. Enter the Error message text. This is the message the user sees when this particular error occurs.
 - a. A place holder will identify the variable name(s) that must be available in the entity where the message will reside. The variable names are surrounded by square brackets ([]). If a square bracket is used as part of the message it must be immediately preceded by a backslash (\[and \]). Where this combination of characters is found, the bracket will be part of the message while the backslash will not (a form of encoding).
 - b. A variable may contain numeric, date or text data. The Rules Palette can retrieve the data type of the variable from the entity in which the error message is being added. If the variable does not exist, the message can only be added without the place holder(s) or canceled. Later when the variable has been added to the entity, the message may be edited with place holders.
7. Enter the Error fix tip. This is the tip the user is given to help fix the error.
8. Right-click on the error catalog file and check it back in.



If the Error Catalog is checked in with empty rows, then a validation error message will display. Delete all empty rows before checking-in the catalog file.

If one of the error messages needs to be deleted, click the Delete box to the right of the error message. Then check-in the Error Catalog.xml file to

save the changes to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Admin Explorer > Exposed Computation > Exposed Computation Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Exposed Computation Overview

The Exposed Computation Web Service is exposed to give an external application access to OIPA's robust math engine. First, an incoming Web Service message calls OIPA and requests a specific computationID. OIPA then queries the database table **AsExposedComputation** to find the requested ID. Once found, the system then queries the **AsBusinessRule** table to find the corresponding Exposed Computation business rule, which contains the XML data for processing. The rule is processed and the results are returned to the external application via an outgoing Web Service message. The computed results are generally not kept within the OIPA database and are managed only in memory as required for the specific calculation.

The Rules Palette provides the means for creating and managing exposed computations. There are two steps required for managing exposed computations through the Rules Palette.

1. [Create a Global Business Rule.](#)
2. [Create an Exposed Computation.](#)



ExposedComputation itself cannot use CopyBooks; however, rules referenced by ExposedComputation can use CopyBooks.

High-level Overview of the Steps Involved in Processing an Exposed Computation Web Service Request

1. Load the [AsExposedComputation record](#) for the ComputationID from the request.
2. Parse the [parameters](#) from the incoming XML document, and set policy context if policy information is supplied.
3. Load [Exposed Computation business rule](#) (from AsBusinessRules).
4. Do [valuation](#) if the Exposed Computation business rule is configured to do so.
5. Execute math from the Input configuration in the Exposed Computation business rule.
6. Build response XML from the Output configuration in the Exposed Computation business rule.

How it Works

A SOAP message is sent by the external application to the ExposedComputation Web Service. The SOAP message includes one OIPA-specific element, which is the ComputationID. The ComputationID identifies the Exposed Computation business rule to use when processing this request. The requested information is calculated and a SOAP message is sent back to the requestor, or caller, along with the result of the request. If the request was successful, then the message will contain the requested data. If the request was not successful, then the message will consist of a SOAP Fault detailing any possible errors.



OIPA adheres to the WS-Security standards for the [authentication of SOAP messages](#).

Defining Parameters

A parameters section may be included to further define the request to the exposed computation.

Inclusion of a **PolicyNumber** parameter tells the exposed computation that it is being processed for a policy. The exposed computation business rule can now be overridden by plan with this parameter, and the math engine will have access to Policy and Plan FIELD variables for that policy when executed. This parameter is also required if the exposed computation is configured to do valuation during the request.

Inclusion of an **EffectiveDate** parameter can be done when an exposed computation executes valuation during the request. This date will be used as the valuation date during valuation. If this parameter is not defined and valuation is still executed, then the valuation date will default to the system date. Below is the expected format for the XML parameter in the request:

```
<Parameters>  
<Parameter  
NAME="PolicyNumber">POL12345</Parameter>  
<Parameter  
NAME="EffectiveDate">01/01/2009</Parameter>  
</Parameters>
```

Valuation

Valuation can be run before executing the math. Use the VALUATION attribute as shown below.

```
<ExposedComputation VALUATION="Yes">  
</ExposedComputation>
```

This makes available all valuation FIELDS (Valuation:Policy:CashValue, Valuation:Fund:FundGUID:CashValue, etc.) to the math configuration. Valuation can only be executed if the request is being processed in the context of a policy.

When processing valuation that may contain variable funds, there is also the ability for using the nearest NUVs for the funds. This is achieved through the following configuration:

```
<ExposedComputation VALUATION="Yes"  
NEARESTNUV="Yes">  
</ExposedComputation>
```

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Admin Explorer > Exposed Computation > Create the Exposed Computation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create Exposed Computation

Once the [ExposedComputation business rule](#) is created, the next step is to create an exposed computation. The exposed computation will associate a ComputationID with the business rule that was created. The ComputationID is the value that the external application will send when requesting information from OIPA.

Steps to Create a New Exposed Computation

1. Select **Admin Explorer**.
2. Open the **Administration** folder.
3. Right-click on Exposed Computation and select **New Exposed Computation**.



Steps	Exposed Computation
1. Exposed Computation	ComputationID: EXP-AUT RuleName: ExposedComputation-AutoAssessment

Exposed Computation Wizard

4. Enter a Computation ID. It is a good practice to begin the Computation IDs with **EXP-** and the abbreviation of the activity it is associated with. For example, EXP-AUT is the Computation ID for the exposed computation that returns the errors generated by the AutoAssessment activity on a given policy.
5. Select the name of the new global business rule that was just created from the drop down box for RuleName.
6. Select **Finish**.
7. Check-in the new exposed computation.

You are here: Admin Explorer > Exposed Computation > Create Exposed Computation Business Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create Exposed Computation Business Rule

The Exposed Computation business rule contains the configuration that tells OIPA how to process the exposed computation. The RuleName from AsExposedComputation for the exposed computation request is used to load an AsBusinessRule record. The record contains an Input and Output element. The Input element contains the math variables' configuration for processing the math engine. The math variables should be configured the same way as any other math section. The Output element contains the mappings for the Input variables to send in the response.

[View a sample XML configuration](#) for additional information about the XML elements.

Steps to Create New Exposed Computation Global Business Rule

1. Navigate to the **Global Rules Explorer** tab.
2. Open the **Business Rules** folder.
3. Right-click on the **Plan Rules** folder and select **New Global Rule for category Plan Rules**.
4. Enter the name for the new rule. It is a good practice to begin the name with **ExposedComputation-** and the name of the activity it is associated with. That way all of the business rules will be grouped together. (Ex: ExposedComputation-AutoAssessment.)
5. Select **Exposed Computation** from the Typecode drop down box. This must be selected or the rule will not display in the filter list later when you set-up the actual exposed computation that is associated with this rule.
6. Enter the additional template information.
7. Select **Finish**. The business rule will display in the Configuration window.
8. Click the **XML Source** tab at the top of the Configuration window.
9. Enter the XML code for the exposed computation. The code can have the following sections.

- **Input:** This section contains the math variables that will calculate the specific information being requested. This is a required element.

```
<Input>
  <MathVariables>
    <MathVariable VARIABLENAME="Variable1"
      TYPE="VALUE"
      DATATYPE="TEXT">TestValue</MathVariable>
  </MathVariables>
</Input>
```

- **Output:** This section contains the mapping information that tells how the requested data should be returned. This is a required element. The data returned in the response XML is built from

the <Output> mappings configured in the exposed computation rule. The root element from the response is the ComputationID from the initial request. Each child element of the root is the Mapping from the output configuration with its math value as the element text. A sample Output section is shown below along with the resulting response.

```
<Output>
  <Mappings>
    <Mapping
      OUTPUTNAME="Result1">Variable1</Mapping>
    <Mapping
      OUTPUTNAME="Result2">Variable2</Mapping>
  </Mappings>
</Output>
```

This is a sample of the response XML:

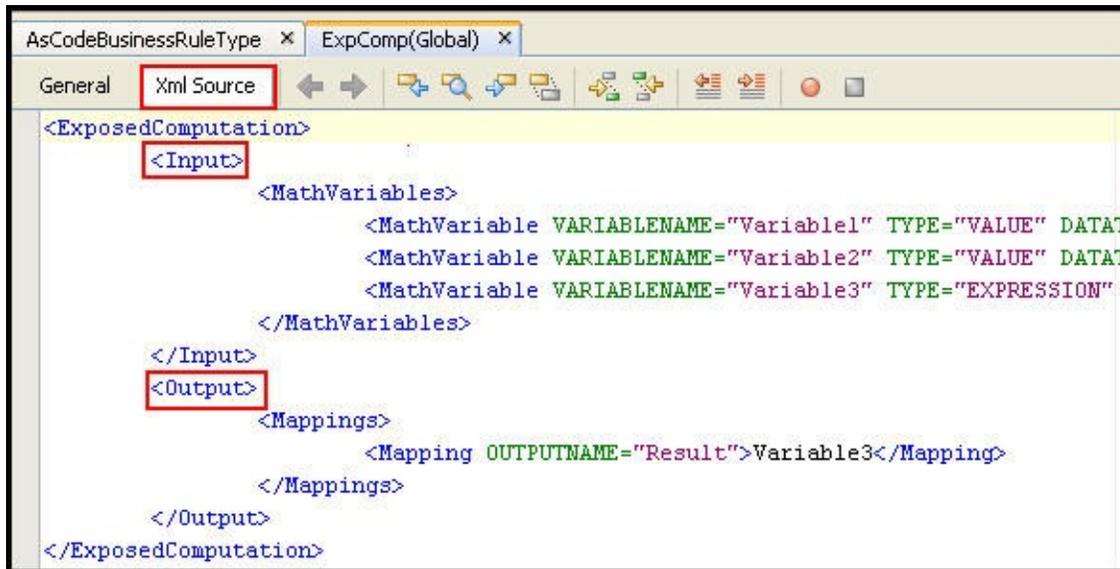
```
<EC_Test>
  <Result1>TestValue1</Result1>
  <Result2>TestValue2</Result2>
</EC_Test>
```

10. Right-click on the XML file and select **Check-in** to save the exposed computation business rule.

Sample Exposed Computation

```
<ExposedComputation>
  <Input>
    <MathVariables>
      <MathVariable
        VARIABLENAME="Variable1"
        TYPE="VALUE"
        DATATYPE="TEXT">TestValue</MathVariable>
    </MathVariables>
  </Input>
  <Output>
```

```
<Mappings>
<Mapping
OUTPUTNAME="Result1">Variable1</Mapping>
</Mappings>
</Output>
</ExposedComputation>
```



Exposed Computation XML Sample

You are here: Admin Explorer > Files > Files Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Files Overview

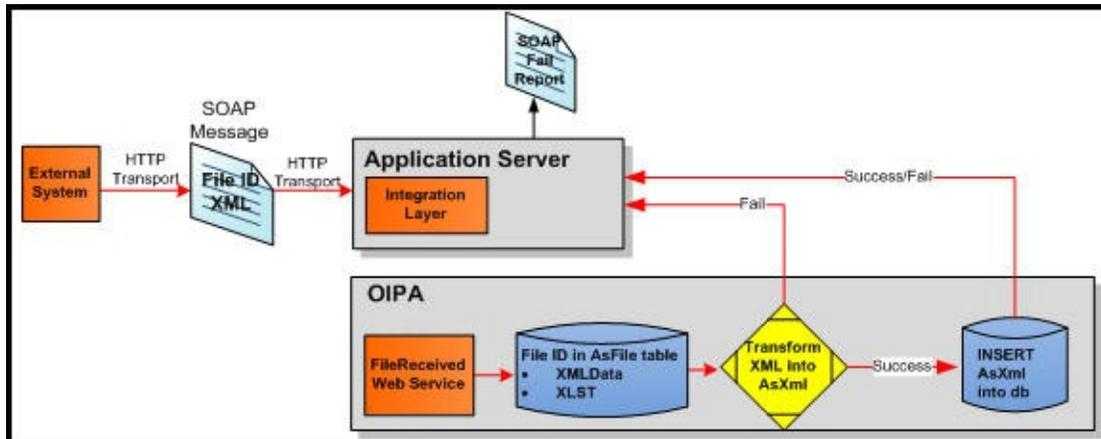
The **Files** administration functionality allows a user to select, transform and upload incoming data sent from an external application and insert it directly into the OIPA database. This functionality can also be leveraged to filter imported data and activities based on the state of a particular entity or request.

You can create a **File** for each required translation via a caller application. Each [file's configuration](#) consists of two sections: the [XMLData](#) and [XSLT](#). The XMLData processes the incoming request by assigning values to variables that will be used for processing. The XSLT directs how the data should be transformed and formatted.

[State approval](#) records can be added to the database through the File process.

High Level Overview of Process

The steps outlined below show the path the files administration functionality takes in order to process a request from an external application.



Overview of Files Functionality

1. External Application sends FileReceived Web Service a request via a SOAP message.
2. The file is looked up using the FileID specified by the request. The [FileID](#) is created in the Rules Palette when a new file is created.
3. The XMLData's AssignAttributes XML is processed.
4. The XSLT maps the request XML to AsXml.
5. The transformed AsXml is mapped to data objects.
6. PreInsert operations are performed on the objects.
7. The objects are inserted into the database.
8. PostInsert operations are performed on the objects.
9. The transformed AsXml is returned to the caller via a SOAP message.

How it Works

A SOAP message is sent by the external application to the FileReceived Web Service. The SOAP message includes two OIPA-specific elements: the FileID and the XML element. The FileID element identifies the configuration for OIPA to use when transforming the inbound XML into OIPA's [AsXml](#). The XML element includes the data to be written to the OIPA database.

Once the SOAP message is received by the FileReceived Web Service, the attributes needed to complete the request are assigned values as configured in XMLData. The data in the XML element is then transformed into AsXml using the configured XSLT. The result of this transformation is then written to the database.

A SOAP message is sent back to the requestor, or caller, including the result of the request. If the request was successful, then the message will consist of the transformed AsXml. If the request was not successful, then the message will consist of a SOAP Fault detailing any possible errors.



OIPA adheres to the WS-Security standards for the [authentication of SOAP messages](#). For additional information on SOAP messages and the FileReceived Web Service, refer to the File Received Web Service document located on the Oracle Technology Network (OTN).

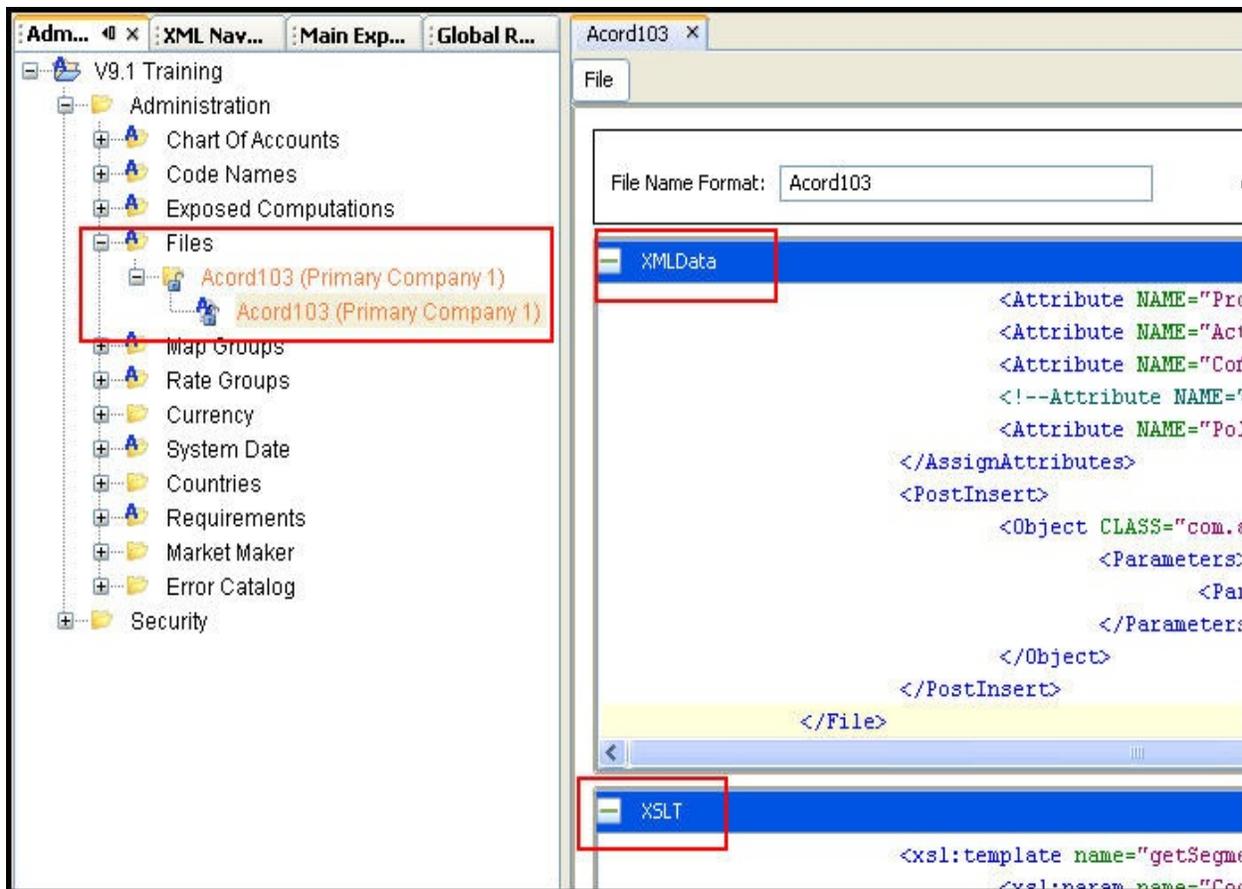
AsFile Database Table

The AsFile database table stores the user-configurable portion of the FileReceived Web Service. The table contains a separate record for each type of file OIPA has been configured to receive. The table includes six columns:

- FileGUID
- CompanyGUID
- FileNameFormat: stores the descriptive name of the file format type
- FileID: stores a unique three-character ID used to describe the file format
- XSLT: stores the XSLT used to transform the inbound XML into AsXml
- XMLData: stores the XML configuration for each specific File business rule



An inbound SOAP message will include a <FileID> element specifying the format of the file being sent.



File Sections Displayed in Configuration Area

You are here: Admin Explorer > Files > Create and Configure a File

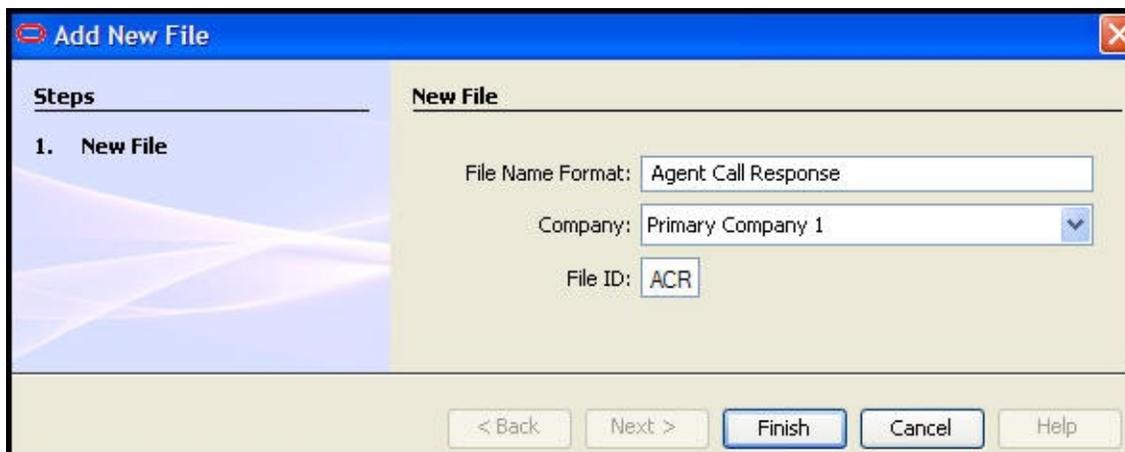
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create and Configure a File

When working with File, there are two steps to perform in the Rules Palette. First, create a new file through the Rules Palette Admin Explorer tab. Then, check-out the new file and [configure the XMLData and XSLT sections](#).

Steps to Create a New File

1. Click the **Admin Explorer** tab.
2. Double-click the **Administration** folder.
3. Right-click the **Files** folder.
4. Select **New File**.
5. Enter a **File Name Format**. This is the name of the new interface.
6. Select the **Company** from the drop down box.
7. Enter a **File ID**, which allows the caller to specify the file to use to process the request. A File ID is limited to three characters and should be an acronym or abbreviation for the calling or process.
8. Click **Finish**.

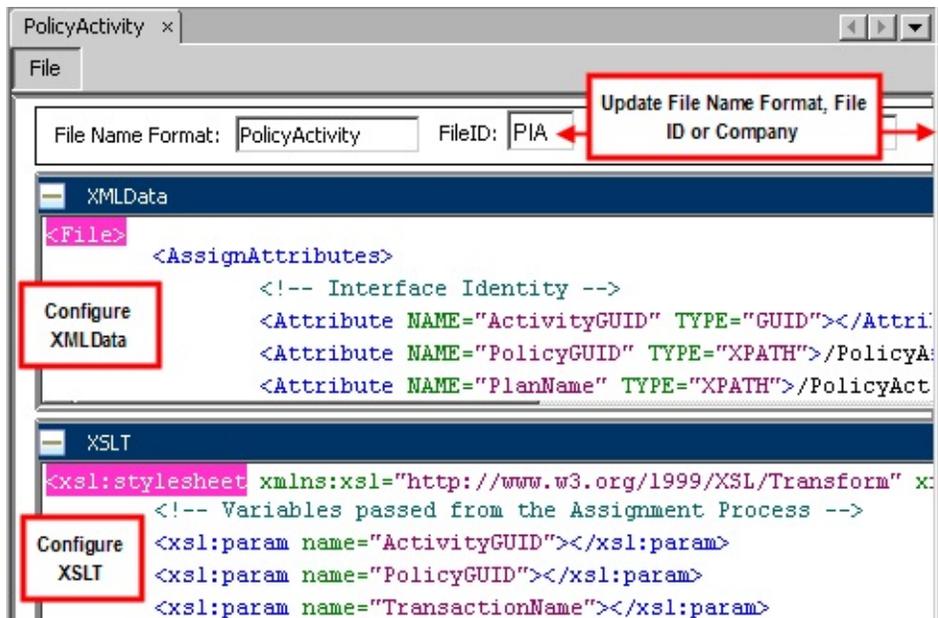


New File Wizard

A folder with the file's XML is created. Use the XML file to configure the file accordingly. Checking-in the file inserts/updates the record in the **AsFile** table in the database. When the file is checked-out, edit the **File Name** and **File ID** using the fields at the top of the Configuration area. Refer to the [Check-Out and Check-in Rules](#) section for further details.

Steps to Configure File

1. Open the **Admin Explorer** window.
2. Double-click the **Administration** folder.
3. Expand the **Files** folder.
4. Open the folder containing the file to configure.
5. Right-click the XML file.
6. Select **Check-out**.
7. Configure in the [XMLData](#) section.
 - a. The <File> element must be the first tag in the XML section and its end tag last.
 - b. Configure the <RequestType> element, if required, for Illustration type AsFile configuration.
 - c. Configure all necessary <Attributes> tags in-between the <AssignAttributes> tag.
 - d. Configure the two required <Attributes> elements, using its NAME and TYPE attributes. The value of <Attributes> is only populated if it is a requirement of the associated TYPE attribute.
 - e. After the closing </AssignAttributes> element, configure pre- and post- insert operations if there is a need to call other functionality in the application. Use the <PreInsert> and <PostInsert> elements to do so.
8. Configure in the [XSLT](#) section
 - a. Configure the XLS stylesheet.
 - b. Perform validations that occur prior to business processing. Validations can also be performed on requests that will not insert an activity record. In addition, validation can be performed before OIPA inserts any records into the database, using XSLT syntax.
9. Check-in the file, using the file's right-click menu.



File XML Example

You are here: Admin Explorer > Files > File XMLData

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

File XMLData

The File business rule allows for values to be assigned to various attributes before the inbound XML undergoes the transformation process into AsXml. This is configured by using the <Attribute> element inside of the <AssignAttribute> parent element to assign values. By assigning values at this stage of the process, pre-existing data from OIPA can be used to populate the AsXml.

Explanation of Elements

<RequestType>

The <RequestType> element is used for defining Illustration processing related AsFile configuration.

<AssignAttributes>

The <AssignAttributes> element is the parent element that contains <Attribute> elements.

<Attribute>

Any processing of data that needs to take place prior to the transformation process is done using the <Attribute> element. The element has two attributes, NAME and TYPE. NAME specifies the name of the attribute, while TYPE defines how the expression will be evaluated. Attributes are evaluated from the top down, so attributes listed first can be used in expressions below them.

For example, the GUID type will set the attribute to a newly generated GUID, and the XPATH attribute will parse data from the inbound XML so that it may be manipulated prior to transformation.



All Attributes used in the XSLT stylesheet must also be defined in the XSLT stylesheet. This is explained in the [XSLT](#) section.

The following table lists the available TYPES:

TYPE	Description
GUID	Sets the attribute to a newly generated GUID.
VALUE	Sets the attribute to the specified value.
SYSTEMDATE	Sets the attribute to the current system date.
SEQUENCE	Sets the attribute by calling asc_NextSequenceInteger and passing the NAME as a parameter.
XPATH	Sets the attribute to the result of the specified XPATH expression.
XPATHSTRINGLIST	Sets the attribute to a comma delimited list containing the resulting values of the XPATH.
XPATHNUMBERLIST	Sets the attribute to a comma delimited list containing the resulting values of the XPATH.
SQL	Sets the attribute to the result of the specified SQL statement.

SQLMAP	Sets the attribute to a key-value-pair type collection of the resulting values of the SQL statement.
--------	--

Examples:

```
GUID: <Attribute Name="PolicyGUID" TYPE="GUID">
</Attribute>
```

```
XPATH: <Attribute Name="Field" TYPE="XPATH">
/Request/PolicyName</Attribute>
```

<PreInsert> and <PostInsert>

<PreInsert> and <Post Insert> are optional elements that allow for other system functionality to be called before or after the data is inserted into the database. This is done by calling specific types of Java classes, which are used for these operations. The architecture of the Pre and Post Insert functionality allows for these classes to be dynamically instantiated at runtime.

Pre and Post Insert operations are specified in the XMLData portion of a File's configuration, after the closing of the AssignAttributes element. The CLASS attribute of both elements sets the name of the Java class to be called.

The following example will invoke the AsFile Post Insert Activity Processor after the records are inserted into the database.

```
<PostInsert>
<Object
  CLASS="com.adminserver.pas.webservice.bll.AsFilePostInsertIn
  dividualActivityProcessorBll">
</Object>
```

Post Insert Example

[View Transformation Example](#)

You are here: Admin Explorer > Files > File XSLT

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

File XSLT

XSLT (Extensible Stylesheet Language) is a XML-based language used for the transformation of XML documents into other XML documents. Using XSLT, OIPA transforms the inbound payload of the SOAP message into AsXml, which can then be processed by the system.

OIPA adheres to XSLT Version 2 specifications, which allows for very flexible configuration of the transformation process. Standard XSLT elements can be used to transform the inbound message into AsXml based on templates and perform data validation and error handling.

Using Attributes from XMLData

Each Attribute defined in the XMLData section that will be needed in the XSLT stylesheet must first be declared as a parameter in the XSLT.

As an example, if an Attribute named PolicyGuid was set in the XMLData section, and its value needs to be referenced in the XSLT, the following would need to be added in the beginning of the XSLT, directly proceeding the prolog.

```
<xsl:param name="PolicyGuid"/>
```

PolicyGUID Attribute

Functions

Several functions are available for use inside of the XSLT stylesheet. This provides for added functionality such as generating GUIDs and retrieving the current system time.

In order to use these added functions, the XsltFunctionHelper class must be added as a namespace in the XSLT prolog as noted below.

```
xmlns:utl="com.adminserver.webservice.helper.XsltFunctionHelper"
```

Xslt Function Helper Class

The getNextGUID() function will generate a new GUID. As an example, the following code would output a newly generated GUID inside of the <PolicyGuid> element.

```
<xsl:element name="PolicyGuid">  
  → <xsl:value-of select="utl:getNextGUID()" />  
</xsl:element>
```

getNextGUID Function

Functions can also be used to retrieve the current system time and then format it properly for insertion into the database.

```
<xsl:template name="GMT">  
  <xsl:param name="Offset" select="0" as="xs:integer"/>  
  <xsl:value-of select="utl:formatDateTime(  
    utl:addMillis(utl:getGmtTime(), . $Offset) )"/>  
</xsl:template>
```

Functions in Xsl

Validation and Error Handling

File has the ability to perform data validations using the XSLT portion of the configuration. As an example, the value of a variable can be tested to ensure the value is as expected. The following validation syntax can be used anywhere in the XSLT.

```
<xsl:if test="$variable = 'incorrect value' ">
  <xsl:variable name="Error1" select="Error·Message"/>
  <ValidationError·ERRORSTATUSCODE="Err001">
    <xsl:value-of select="$Error1·"/>
  </ValidationError>
</xsl:if>
```

Data Validations

After evaluation, if the `<xsl:if>` expression is true, then the `<ValidationError>` element will be included in the resulting AsXml. If one or more `<ValidationError>` elements exist in the AsXml, then a SOAP fault will be thrown and the text within the element and the `ERRORSTATUSCODE` will be returned to the caller as part of the SOAP fault.

[View Transformation Example](#)

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Admin Explorer > Files > File AsXml

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

File AsXml

AsXml is the result of the XSLT transformation process. This is the format used by OIPA to store data destined for the database. The format is very simple. It is comprised of a parent element for each database table, and a child element for each column of the table. The root element, <AsXml>, must be used to identify the formatting.

Each element must exactly match the name of the table or column in the database for mapping purposes. Each column of the table needs to be populated with data unless the column is set to allow NULL.



To reference the term **GUID** in relation to the database (PolicyGUID or RateGUID), it must be entered as **Guid** (PolicyGuid) in the AsXml.

Schema

```
<AsXml>  
<Table  
<Column>Value</Column>  
</Table>  
</AsXml>
```

[View Transformation Example](#)

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Admin Explorer > Files > Batch Processing Using AsFile

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Batch Processing Using AsFile

Batch processing is used to insert policy level activities and suspense. In OIPA, batch processing can be accomplished using AsFile.

Some additional configuration must be added to AsFile to allow an external application to pass batch information. The following section explains how AsFile is altered to prepare for batch processing.

Prepare for Batch Processing

To prepare for batch processing, you must make sure that the external application can send the FileReceived Web Service the input XML with PolicyNumber and activity information for multiple policies.

The AsFile database table stores the user-configurable portion of the FileReceived Web Service. The table contains a separate record for each type of file OIPA has been configured to receive. One record in this table contains the XML data. A pre-extension can be written at the extension point **FileReceived.StarProcessingFileReceived** to alter the XML to add additional information like policyGuid, TransactionGuid, etc.

AsFile.XSLT can transform the input XML to AsXML with multiple AsActivity and AsActivityField tags to insert the relevant activities. The activities are picked up in cycle for processing.



These new activities cannot be linked to AsBatch and associated tables because those tables do not actually exist.

Resolve a Failure

In the event that a processing failure occurs, such as a network problem, a timeout or data error, all data is rolled back.

Timeout issues can be minimized by manipulating the timeout limit that is set in the application server. A typical default timeout is 180 seconds, but this can be increased if needed.

Data errors can be managed in the XSLT by conditionally adding the `ValidationError` tag to the `AsXML`. Since there is no UI screen for `AsFile`, the error message is sent back in the response XML.

You are here: Admin Explorer > Files > SOAP Security

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

SOAP Security

OIPA adheres to the WS-Security standards for the authentication of SOAP messages. The standards, as developed by the OASIS Open committee, can be referenced at the following two URLs. Paste a URL in a browser to view the standards.

<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>

<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf>

The <wsse:UsernameToken> element is used to contain the authentication information. The username and password are specified inside of the <wsse:Username>, and <wsse:Password> elements, respectively. Use an OIPA user name and password here.

It is suggested that SSL (Secure Socket Layer) is used as a method of encryption for all SOAP messages.

The optional <wsse:Nonce> element allows a nonce to be used as added security. A nonce is a random number, in this case represented in base 64, which is embedded in the security header to aid in preventing old communications from being reused. This number is newly generated for each request on the client side and is returned along with the SOAP response from OIPA. The <wsu:Created> element must contain the timestamp of the creation time of the nonce.

```

<soapenv:Header>
  <wsse:Security
    soapenv:mustUnderstand="1"
    xmlns:wsse="http://docs.oasis-open.org/wss/2004
      /01/oasis-200401-wss-wssecurity-secext-
      1.0.xsd">
    <wsse:UsernameToken
      wsu:Id="UsernameToken-1"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004
        /01/oasis-200401-wss-wssecurity-
        utility-1.0.xsd">
      <wsse:Username>username</wsse:Username>
      <wsse:Password
        Type="http://docs.oasis-open.org/wss/2004
          /01/oasis-200401-wss-username-token-
          profile-1.0#PasswordText">password
        </wsse:Password>

      <wsse:Nonce
        EncodingType="http://docs.oasis-open.org
          /wss/2004/01/oasis-200401-wss-soap-message-
          security-1.0#Base64Binary">
        <wsu:Created>
          OutRdm07dLg/v+0DI04/DA==</wsse:Nonce>
          2009-09-28T17:43:02.546Z</wsu:Created>
        </wsse:UsernameToken>
      </wsse:Security>
    </soapenv:Header>

```

SOAP File with Security

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Admin Explorer > Files > A Transformation Example

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

A Transformation Example

XML Portion of SOAP Request:

```
<Request>
```

```
→ <PolicyName>TestPolicy</PolicyName>  
</Request>
```

XSLT Section:

```
<xsl:stylesheet·  
→ xmlns:xsl="http://www.w3.org/1999/XSL/Transform"·  
→ version="2.0">  
→   
→ <xsl:template·match="/Request">  
→ ····<AsXml>  
→ → ·<AsPolicy>  
→ → ···<PolicyName>  
→ → → <xsl:value-of·select="/Request/PolicyName"/>  
→ → ···</PolicyName>→  
→ → ·</AsPolicy>  
→ → </AsXml>  
→ </xsl:template>  
</xsl:stylesheet>
```

Resulting AsXml after transformation:

```
<AsXml>
  → <AsPolicy>
  →   → <PolicyName>
  →   →   → TestPolicy
  →   → </PolicyName>
  → </AsPolicy>
</AsXml>
```

You are here: Admin Explorer > Files > Illustration Processing Using AsFile

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Illustration Processing Using AsFile

Illustrations can be processed by configuring an AsFile with the RequestType element set to "Illustration". The AsFile created to handle Illustrations can be configured to use the Policy-Illustration type transaction.

Steps in Illustration processing

- a. An external system will submit an AsFile request of type “Illustration”
- b. OIPA will extract the submitted policy and illustration data and insert them into the database (similarly to what is done for the Insert requests)
- c. An activity will be created based on the data submitted inside the AsActivityQuote tags and executed in the Quote mode (similar to processing of the Quote requests)
- d. An output XML is created (as usually done for requests of the Quote type)
- e. All data inserted into the database in step b will be deleted from the database before output XML is returned to the caller.

The Illustration requests do not support post-insert processing.

Payload for an Illustration SOAP request from external system

The payload of a request of the Illustration type contains the following data:

- a. Data required to populate AsPolicy and related tables necessary to calculate the illustration
- b. Data to populate the new AsIllustration table documented in the FDD
- c. AsActivityQuote data used to create an illustration activity (similar to AsActivityQuote data used in AsFile requests of the Quote type)

Illustration related Data Tables

The following are new data tables created to handle Illustrations:

- a. AsIllustrationRequest
- b. AsIllustrationBasis
- c. AsIllustrationBasisField
- d. AsIllustrationReport
- e. AsIllustrationTransaction
- f. AsIllustrationVector

You are here: Admin Explorer > Intake Profile Definition

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Intake Profile Definition

The Intake Profile Definition page is used to configure a Data Intake Profile. The page consists of three tabs containing text editors similar to the XML Source pane used to configure business rules, transactions and other entities elsewhere in the Rules Palette.

Tabs

IntakeProfileDefinition

This tab is used to configure a Data Intake Record Schema, which is used to define all of the fields contained in an incoming record, and is configured to define all of the field names that can be received in a record. For information on configuring a Data Intake Record Schema, see the XML Configuration Guide—navigate to **Integration | Data Intake | RecordSchema**.

File

This tab is used to configure a Data Intake File Definition, which allows for the creation of dynamic fields for a Data Intake File. The processing of files through Data Intake supports the writing of data back to these fields. For information on configuring a Data Intake File Definition, see the XML Configuration Guide—navigate to **Integration | Data Intake | FileDefinition**.

Record

This tab is used to configure a Data Intake Record Definition, which allows for the creation of dynamic fields for a Data Intake Record. The processing of files through Data Intake supports the writing of data back to these fields. For information on configuring a Data Intake File Definition, see the XML Configuration Guide—navigate to **Integration | Data Intake | RecordDefinition**.

Creating and Editing Intake Profile Definitions

Creating an Intake Profile Definition

1. Navigate to the **Admin Explorer** window.
2. Expand the **Administration** folder.
3. Right-click on the **Intake Profile Definition** folder and select **New Intake Profile Definition**.
4. Enter a name for the Intake Profile Definition in the **Profile Definition Name** field.
5. Select the company to which the Intake Profile Definition should belong from the **Company** drop-down.
6. Select a TypeCode for the Intake Profile Definition from the **Definition Type Code** drop-down.
7. Click **Finish**.

Editing an Intake Profile Definition

1. Navigate to the **Admin Explorer** window.
2. Expand the **Administration** folder.
3. Expand the **Intake Profile Definition** folder.
4. Expand the folder for the Intake Profile Definition that requires editing.
5. Right-click on the Intake Profile Definition's XML file and select **Check out**.
6. Make the necessary changes.
7. Right-click on the Intake Profile Definition's XML file and select **Check in**.

On editing the Record Schema of any Intake profile definition, there will be an option available in PAS to update the Record Schema for their respective Intake profiles (Created via PAS before the definition was edited)

A Merge button will be provided on the Intake profile Screen in OIPA for all the profile's for which the record schema of the definition has been updated

On Click of 'Merge' button all the 'New' Fields or Entities added to the definition will be copied over from the definition (ASINTAKEPROFILEDEFINITION) to the Record Schema of the Profiles created via PAS (ASINTAKEPROFILE) .

Please note that only new fields and entities will be copied and no changes would be made to an existing field (even though there was a modification done in Profile definition). If required all such changes needs to be manually done on those specific Fields or Entities via Record details tab on the Intake profile screen.

The availability of 'Merge' button in PAS will be controlled via security given to the 'Merge' button in 'Intake Profile' page under Company pages security

You are here: Admin Explorer > Map Groups > Map Groups Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Map Groups Overview

Map groups are used when constant values for a [plan](#) are based on multiple criteria. For example, a plan might have different free look periods depending on the state where the policy is issued. The state where a policy is issued determines which value is chosen from the map group. Using SQL statements configurors can retrieve values according to criteria for rule configuration. The value is then assigned to a math variable and used for calculations or processing.

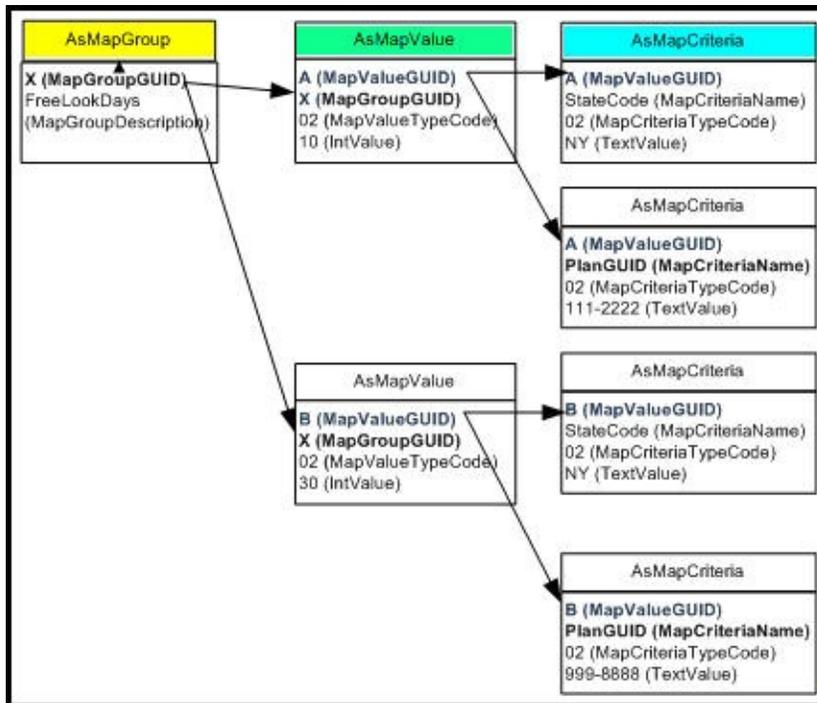
There are three tables used to create map groups.

- **AsMapGroup** is the initial table. This is where the map group is defined and named. The MapGroupDescription is used to identify the map group. The name should be understandable and reusable across plans and segments. The unique MapGroupGUID is dynamically created and links to AsMapValue.
- **AsMapValue** holds all the possible values for a map group and is linked to the AsMapCriteria, which associates the criteria to the values from AsMapValue.
- **AsMapCriteria** is used to create the set of criteria. Each set of criteria has the same MapValueGUID.

[Create map groups](#) and [configure the criteria and values](#) from the Map Groups link on the **Admin Explorer** tab.

Free Look Map Table Example

Insurance plans can have different free look periods depending on the state where the policy is issued along with other factors. New York (NY) requires a 10 day minimum for the free look period and 30 days for policies that were offered through the mail. Based on this requirement, a map group is created and the MapGroupDescription is named FreeLookDays. The values of 10 and 30 days are entered in AsMapValue and linked to AsMapGroup by the MapGroupGUID. The state and plan are the criteria and will be used in AsMapCriteria. In AsMapCriteria, StateCode is used to identify the state and PlanGUID is used to identify the plan. NY will be used as the StateCode and the actual PlanGUIDs will be used to identify which plan is offered through the mail and which one is not.



Relationship of Map Group Tables

There should only be one Map Group for each MapGroupDescription. There may be many AsMapValues associated with one MapGroupGUID. Each MapValueGUID must have the same set (number) as AsMapCriteria criteria.



The use of an asterisk (*) as a criteria value is useful for any other situations where you do not want to list all possible variations (such as not listing all 50 states). Use the * as the criteria value in the TextValue column.

You are here: Admin Explorer > Map Groups > Create Map Group

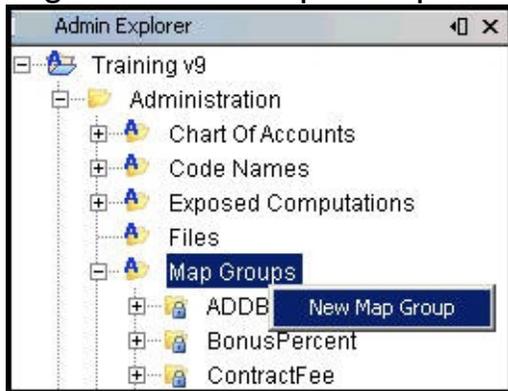
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create Map Groups

The following steps describe the process for creating map groups.

Steps to Create a New Map Group

1. Navigate to the **Admin Explorer**.
2. Expand the **Administration** folder.
3. Right-click on Map Groups and select **New Map Group**.



Map Groups Right-Click Option

4. Enter the name of your new Map Group.
5. Select **Finish**. Your new Map Group will be listed in the Map Group folder. Now you need to [configure the criteria and values](#).

You are here: Admin Explorer > Map Groups > Configure Map Group Criteria and Values

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Configure Map Group Criteria and Values

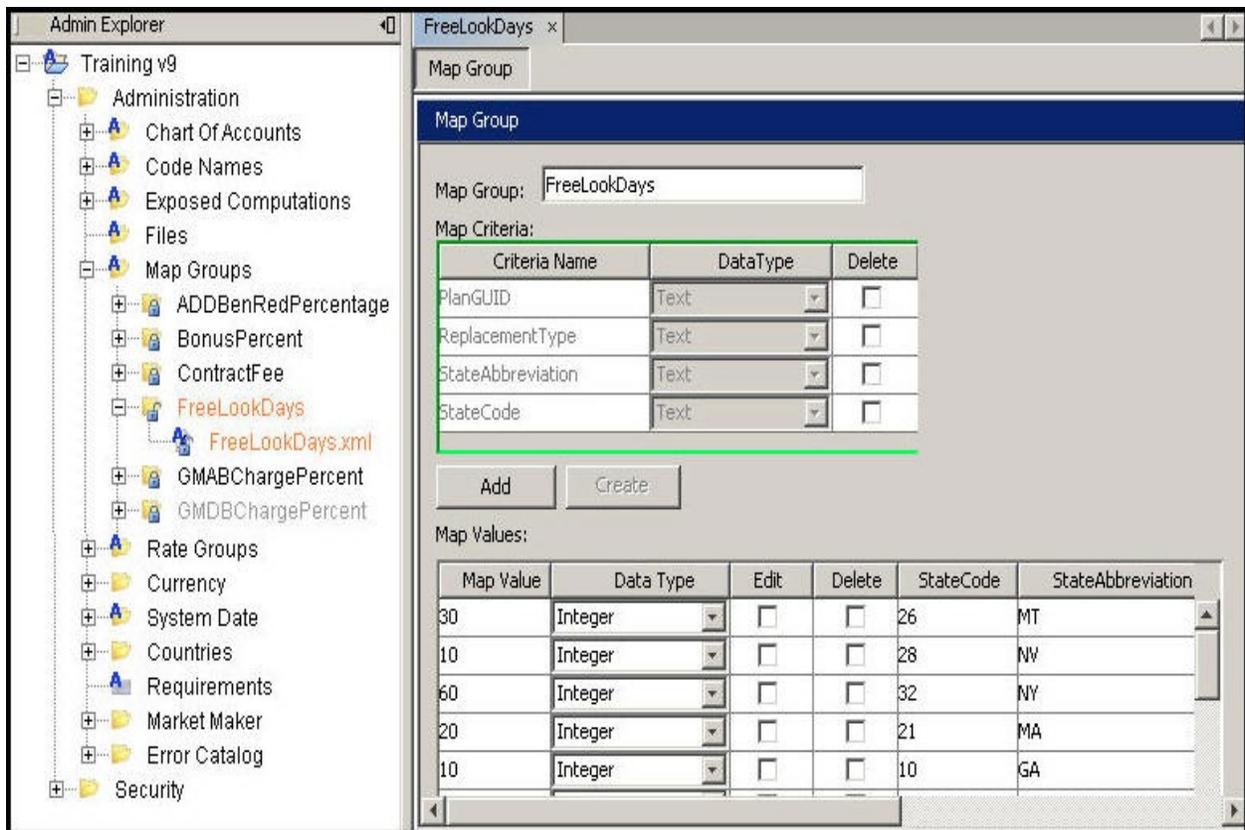
The following steps describe the process for configuring map group criteria and values. You can also [edit existing map groups' criteria and values.](#)

Steps to Configure New Map Group Criteria and Values

1. Right-click on the new map group and select **Check-Out**. The Map Group will open in the Configuration Area.
2. Double-click in the Criteria Name column and enter the name of the criterion to add.
3. Select the down arrow next to the Data Type field and select the type of data to enter. The choices are Date, Decimal, Integer or Text.
4. Select **Create** to save the criteria.
5. Move down to the bottom section of the screen and select **Add** directly under the Values section. A row of fields will appear for the Map Value information. The criterion just entered should be listed across the top in the field headings.
6. Double-click in the **Data Type** field first and select the type of data to enter for the Map Value. If the Map Value is entered first, it will be erased when the Data Type is selected.
7. Double-click in the Map Value field and enter the Map Value. If entering a date, click on the down arrow in the field and a calendar will appear for the date selection.
8. Enter the values for the criteria columns.
9. Select **Add** to enter additional Map Values. Another row of fields will appear.
10. When finished entering Map Value information, right-click the XML file from Admin Explorer and select **Check-In**. The information entered will be saved.



If you decide to delete any of the Map Criteria or Map Values, check **Delete** in the row that contains the data you want to delete. The information will be removed when the XML file is checked in.



FreeLook Map Group Sample

Steps to Edit Map Group Criteria

1. Right-click on the Map Group XML file to edit and select **Check-out**.
2. Select **Add** to create additional criteria. A field will appear for the criteria values.
3. Select **Create** after all the criteria values are entered.
4. Check **Delete** to delete a criterion.
5. Right-click on the XML file and select **Check-in** to save the changes.



If you make changes to the Criteria, the Map Values Delete box will be disabled. You will not be able to delete any Map Values until you check-in the file and save the Criteria changes.

Steps to Edit Map Group Values

1. Right-click on the Map Group XML file to edit and select **Check-out**.
2. Double-click in the field that contains the value to edit. The row will be highlighted in green.
3. Enter the new value and then click once on another field to close the field that was being edited.
4. Select **Add** to create another Map Value row. The new row will be highlighted in dark blue.
5. When the values have been updated, right-click the XML file and select **Check-in**.



If you want to delete a Map Value, check **Delete** in the row that contains the Map Value you want to delete. The row will be highlighted in red. The Map Value will be removed once you check-in the XML file.

Updating Translation Values when Map Criteria and Values are Edited

A translation button is available on each row to allow easy access to the Translation editor. Click the node in the row to open the translation values for the Short and Long Descriptions of the map value. Make any changes and click OK to close the window and save the changes.

If changes are checked in for a map value or criteria in Map Groups and the translation node is not clicked, then the Rules Palette will open the Translation Editor window so that translation values can be updated if needed. Update any values and click **OK** to save the changes.



Translation Editor Window

You are here: Admin Explorer > Map Groups > Retrieve Map Group Values

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Retrieve Map Group Values

In order to retrieve the rate, a SQL needs to be written that links all three tables. The same rules and standards for SQL apply for the AsMapSeries lookup.

For example, to lookup the GMIB rate, assume that the following variables have been set:

- PlanGUID: E99F8927-B97E-40F4-B6A3-D53C6BA00F82
- SegmentRateActiveDate: 3/5/2006
- IssueState: 32

```
SELECT AsMapValue.IntValue FROM AsMapValue
JOIN AsMapGroup ON AsMapGroup.MapGroupGUID =
AsMapValue.MapGroupGUID
AND AsMapGroup.MapGroupDescription = 'GMIBMaximumAge'
JOIN AsMapCriteria ON AsMapCriteria.MapValueGUID =
AsMapValue.MapValueGUID
AND AsMapCriteria.MapCriteriaName = 'PlanGUID'
AND AsMapCriteria.TextValue = '[Policy:PlanGUID]'
JOIN AsMapCriteria AS R ON R.MapValueGUID =
AsMapValue.MapValueGUID
AND R.MapCriteriaName = 'StateCode' AND ((R.TextValue =
'[IssueState]') OR (R.TextValue = '*'))
ORDER BY R.TextValue ASC,S.TextValue ASC FETCH FIRST ROW
ONLY
```



Without the FETCH FIRST ROW ONLY, the Query would return both rates associated with the GMIBMaximumAge MapGroup. Due to the ordering of R.TextValue, NY will appear prior to *, and therefore the FETCH FIRST ROW ONLY will correctly return an integer value of 65.

The above SQL shows:

- Hardcoded values (AsMapGroup.MapGroupDescription =

'GMIBMaximumAge'),

- Policy fields (AsMapCriteria.TextValue = '[Policy:PlanGUID]')
- A math variable (R.MapCriteriaName = 'StateCode' AND ((R.TextValue = '[IssueState]') OR (R.TextValue = '*')))
- The use of * (as seen in the StateCode criteria).

Free Look SQL Example

```
SELECT TOP 1 AsMapValue.IntValue FROM AsMapValue
JOIN AsMapGroup ON AsMapGroup.MapGroupGUID =
AsMapValue.MapGroupGUID
AND AsMapGroup.MapGroupDescription = 'FreeLookDays'
JOIN AsMapCriteria ON AsMapCriteria.MapValueGUID =
AsMapValue.MapValueGUID
AND AsMapCriteria.MapCriteriaName = 'PlanGUID'
AND (AsMapCriteria.TextValue = '[Policy:PlanGUID]'
OR AsMapCriteria.TextValue = '*')
JOIN AsMapCriteria AS Q ON Q.MapValueGUID =
AsMapValue.MapValueGUID AND Q.MapCriteriaName =
'ReplacementType'
AND (Q.TextValue='[ReplacementCriteria]')
JOIN AsMapCriteria AS R ON R.MapValueGUID =
AsMapValue.MapValueGUID AND R.MapCriteriaName ='StateCode'
AND (R.TextValue = '[Policy:IssueStateCode]' OR R.TextValue='*')
ORDER BY R.TextValue DESC
```

You are here: Admin Explorer > Rates > Rate Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Rate Overview

OIPA supports the management of rates using various methods. Rates can be entered [manually](#) or [uploaded](#) via an excel spreadsheet to the AsRate table. The DateCriteria column in AsRate provides the effective date in which the rate(s) apply. All rates loaded by a single spreadsheet have the same date criteria value. These rates are not accessible via RATEARRAY and RATE variable types. Repricing does not apply to these types of rates. OIPA valuation routines use these rates that contain only one rate group.

Other rates use multiple rate groups because these rates experience changes based on external factors. This information is still housed within the AsRate table, but the dates are maintained in the AsRateGroup table. These are suitable for fees, COI and regulatory rates and any product rate directly referenced in a contract to a customer. These rates are not suitable for OIPA's valuation routines. Generally the rates are uploaded via a spreadsheet as the volume of rates is typically large. In special circumstances, [an activity is used to generate the new rates and populate the database with rate records](#). Rates with multiple rate groups may be accessed via RATEARRAY and RATE variable types.

Uploading Rates

Rates used for valuation are usually entered manually through the Rules Palette, as the volume of rates is low. The rates populate the AsRate table in the database. In order to upload rates, a [rate group](#) that describes the rates must first be created. A rate group bands rates together according to their designated purpose and provides a means of describing and [retrieving rates](#). Usually rates are created for a specific plan.

Once a rate group is created, the actual rates may be uploaded as [aggregate](#), select or a user may [enter rates manually](#). The rates can be provided in an Excel spreadsheet by the business, client or party responsible for supplying rate information for a plan.

Updating Rates Via Transactions

In some cases, insurance rates are updated by applying experience factors or other frequency-based factors to an existing set of rates. These generated updates to rates are applicable for a certain time frame and may be used by existing policies as they age through that time period, or they may be applicable only to new policies issued within that time period. The CreateRates business rule was developed for the purpose of repricing this type of rate.

The [CreateRates](#) rule can be attached to a transaction to create a new AsRateGroup database record and a new set of rates in AsRate. This rule should be placed in the TransactionBusinessRulePacket for the transaction. Refer to the [Rate prototype](#) for additional information on configuring this rule to create new rates.

Adding Rates to an Existing Rate Group

In other cases where rates are tied to and fluctuate based on external factors, new rates may need to be added to an existing rate group without adding a rate group record. This can also be accomplished with a transaction, by using the [CreateAdditionalRates](#) attached rule, which is able to generate a set of new date-based rates within an existing rate table that is associated to a rate group. This rule should be placed in the TransactionBusinessRulePacket for the transaction.

Refer to the [CreateAdditionalRates prototype](#) page in this help system, as well as the CreateAdditionalRates page in the XML Configuration Guide, for additional information on configuring this rule to create new rates. To open the page in the XML Configuration Guide, open the help system from the **Help** menu in the Rules Palette and navigate to **Configuration Overview | Business Rules | Attached Rules | CreateAdditionalRates**.

Database Tables

There are two rate tables that hold rate information: AsRate and AsRateGroup. AsRate holds all the individual rate records, regardless of rate type. Valuation rates use the DateCriteria column in AsRate to determine the rate's effective date. On the rare occasions when rates are created using an activity and the CreateRates business rule, the AsRateGroup table holds the date information for each group of rates.

Security

OIPA users may view information loaded into the Rates tables when the **Tables | Rates** option is selected from the OIPA Main menu. These rates will only be visible to the user if security privileges are granted through the Security folder in the Admin Explorer.

To add security to the Rates tables, click the Admin Explorer tab and navigate to **Security | Application Security | Security Groups | *Name of the Security Group* | Company Security | Company Pages | *Name of the Company***. The Rates option is listed under the Company. Check out the file and check the box to grant access to the Rates tables.

There are also two buttons available to users when the Rates window displays in OIPA: Filter and Find. A checkbox next to those boxes in Security will allow a user to filter rates according to user specified criteria and search for rates.

You are here: Admin Explorer > Rates > Rate Groups

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Rate Groups

Rate groups help organize and associate rates together in the AsRateGroup table. Once a rate group is [created](#) a unique GUID is assigned and used to link the rate group to rates. The rate group stores a description for rates, the rate indexing method and rate criteria(s). These key pieces of information are required for the rate group and drive how the system will use the associated rates for configuration and policy events.

Key information required for rate groups:

- **Rate Description:** A descriptive name that is used to identify the rates. This should be the name provided with the rate table. This is stored in the RateDescription column in AsRateGroup.
- **Integer Criteria:** This is used as a means of identifying the name of the primary method of rate indexing. Rate indexing is a mechanism employed by insurance companies to track the amount of time or change affecting an insurance rate. Main indexes are generally age and policy duration (in years), which are both integer indexes. The actual index values, such as 1, 2, 3, etc., are actually stored in AsRate. The name of the index is stored in IntegerCriteria of the AsRateGroup table.
- **Rate Activation Date:** This date is used in conjunction with a policy's effective date to identify when a set of rates is active for a policy. The policy must be effective on or after the date listed here to use the set of rates. It is this date that is used as a way to distinguish multiple sets of rates tied to a single rate group name. Rates may change due to [re-pricing](#) after the launch of a plan or for various other reasons. While all other factors may remain, by creating an identical rate group with a different rate activation date, you can manage another set of rates for the same product.



Rate Activation Date is stored as EffectiveDate in the

AsRateGroup table.

- **Transaction From Date:** A date used in conjunction with a transaction that requires the use of rates if a transaction occurs on or after this date, but prior to the ActiveToDate. This is stored in the ActiveFromDate column in AsRateGroup.



Transaction From Date cannot change after it has been set, as the system would see this as a need for undo/redo or backdating.

- **Transaction To Date:** A date used in conjunction with transaction that requires the use of rates if a transaction occurs before this date. This is stored in the ActiveToDate column in AsRateGroup.



Transaction To Date cannot change after it has been set, as the system would see this as a need for undo/redo or backdating.

- **Criteria 1 through 10:** In these fields you will enter the criteria descriptions for the Excel spreadsheet with the rates. Criteria names must remain constant for all rate groups in the system. For example, if UWClass is used, then there should not also be entries for Underwriting Class or UWclass. Keep the syntax and spelling the same for criteria with the same meaning. These criteria names will correlate to columns in your rate table. The system is based on XML and is case sensitive so your criteria casing must match. For example, the words Gender and gender are interpreted differently.



In AsRateGroup, the StatusCode, TypeCode and XMLData columns are not used.

You are here: Admin Explorer > Rates > Create New Rate Group

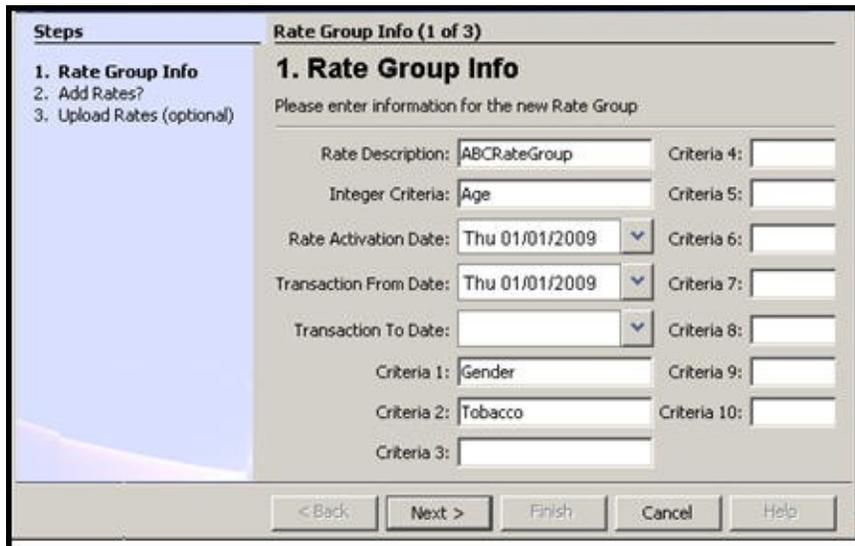
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create New Rate Group

The steps below explain the process for creating a new rate group for rates used for valuation. For the special cases when rates groups for rates fluctuate based on external factors, rates are created using an [activity](#).

Steps to Create a New Rate Group

1. Click the **Admin Explorer** tab.
2. Right-click on the **Rate Groups** folder.
3. Select **New Rate Group**. The Rate Group Wizard will open.



The screenshot shows the 'Rate Group Info (1 of 3)' wizard. On the left, a 'Steps' pane lists: 1. Rate Group Info, 2. Add Rates?, and 3. Upload Rates (optional). The main area is titled '1. Rate Group Info' and contains the following fields:

- Rate Description: ABCRateGroup
- Integer Criteria: Age
- Rate Activation Date: Thu 01/01/2009 (dropdown)
- Transaction From Date: Thu 01/01/2009 (dropdown)
- Transaction To Date: (dropdown)
- Criteria 1: Gender
- Criteria 2: Tobacco
- Criteria 3: (empty)
- Criteria 4: (empty)
- Criteria 5: (empty)
- Criteria 6: (empty)
- Criteria 7: (empty)
- Criteria 8: (empty)
- Criteria 9: (empty)
- Criteria 10: (empty)

At the bottom, there are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Step One of Rate Group Wizard

4. Enter the **Rate Description**. This is a descriptive name that is used to identify the rates. It should be the name provided with the rate table. This name will be stored as the RateDescription in AsRateGroup and AsRate.
5. Enter the **Integer Criteria**. This is used as a means of identifying the name of the primary method of rate indexing. Rate indexing is a mechanism employed by insurance companies to track the amount of time or change affecting an insurance rate. Main indexes are generally age and policy duration (in years), which are both integer indexes. The actual index values, such as 1, 2, 3, etc., are actually stored in AsRate. The name of the index is stored in IntegerCriteria of the AsRateGroup table.
6. Enter the **Rate Activation Date**. This date is used in conjunction with a policy's effective date to identify when a set of rates is active

for a policy. The policy must be effective on or after the date listed here to use the set of rates. It is this date that is used as a way to distinguish multiple sets of rates tied to a single rate group name. Rates may change due to [re-pricing](#) after the launch of a plan or for various other reasons. While all other factors may remain, creating an identical rate group with a different rate activation date allows you to manage another set of rates for the same product.

7. Enter the **Transaction From Date**. This date is used in conjunction with a transaction that requires the use of rates if a transaction occurs on or after this date, but prior to the ActiveToDate. TransactionFromDate is stored as the ActiveFromDate in AsRateGroup.
8. Enter the **Transaction To Date**. This date is used in conjunction with a transaction that requires the use of rates if a transaction occurs before this date. Transaction To Date cannot change after it has been set, as the system would see this as a need for undo/redo or backdating. When loading new rates, this field should be left blank. The TransactionToDate is stored as the ActiveToDate in AsRateGroup.
9. Enter the descriptions for **Criterion 1 through Criterion 10**. The descriptions should be entered as they appear in the Excel spreadsheet. Criteria names must remain constant for all rate groups in the system. These criteria names will correlate to columns in your rate table. Criteria names are case sensitive so make sure the names match exactly. For example, if UWClass is used, then there should not be entries for Underwriting Class or UWclass. Keep the syntax and spelling the same for criteria that have the same meaning.
10. Select **Next**.
11. Determine how to add the rates.
 - a. Select **Yes** to upload an Excel spreadsheet of rate information.

You will be prompted to upload rates.

- b. Select **No** and then select **Finish** if you want to [add the rates manually](#) using the Rate Group Editor.

After the Rate Group has been successfully added, you can upload [aggregate rates](#) or select rates from an excel spreadsheet.

You are here: Admin Explorer > Rates > Upload Rates > Rate Table Formats

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Rate Table Formats

The Rules Palette rate upload process provides a user friendly mechanism to load rates, even multiple sets of rates and multiple rate groups in a single session, via Excel spreadsheets with nominal set up in Palette required. The ability to overwrite rate sets for existing rate groups is also available.

General Information

An insured has an issue age and an attained age. The issue age is based on the person's age at the time the insurance policy goes into force. The insured's actual age, their age at their nearest birthday or their age at their last birthday are typically used to determine their issue age. The attained age is calculated at the policy's anniversary, and is the issue age plus the years in force (based on the number of anniversaries passed). The attained age is usually kept on the policy or segment, although it may be calculated throughout. At time of issue, an insured's issue age equals their attained age.

A number of criteria are used in setting rates. The number of criteria used and the basis for distinction will vary by insurance company. Criteria commonly used include, but are not limited to:

- Gender
- Smoker status
- Underwriting status
- Band (based on premium, face, cash value, or other method)
- Rider/benefit combination
- Multiple policy discount

The table format selected for rate upload will impact the values required in the wizard. Below are brief explanations of the three defined table formats.

Aggregate Rates

Aggregate rates group more individuals to a single rate than select rates, and are typically attained age-based. This means that a person's rate, for a given set of criteria, is determined only by their attained age. For example, a person with Issue Age 40 will have the same rate for a person in duration 5 with an Issue Age of 35, since the attained age in duration 5 would be 40.

To create the rates, then, all the rates for a given criteria combination will be held in a single row or column with the key "attained age." The system will navigate down a given row or column for the duration indicated. Once one row/column is complete, that criteria combination is done, and the system would then move to the next row/column for the next criteria combination.

Select Rates

Select rates are typically issue age- and duration-based, where duration is determined by a maximum age (e.g. 121). This means that a person's rate for a given set of criteria is determined by both the issue age and the policy's current duration. As an example, because of the way select rates are calculated, the insured is typically going to have a lower rate for attained age 40 if the policy was purchased at Issue Age 35 than at Issue Age 40.

Because both issue age and duration are used in setting select rates, each criteria combination will have multiples rows and columns to traverse to complete a single combination. For example, to create the rates, the issue ages are in a row and the duration is in a column. Starting with the initial age, the system will traverse through the column for all applicable durations. Once one issue age is complete, the system must process the next issue age in the same manner. The system would continue in this manner until all issue ages are complete. Only after that is the criteria combination done, and the next criteria combination can begin.

Note: Because select rates are based on issue age and duration to a max age, it is likely that not all issue ages will have the same number of rates populated.

Ultimate Rates

Ultimate rates are a combination of the above two rate types: they are select rates up to a given duration, and then the rates switch to aggregate rates. These rates may be used for Term policies that are guaranteed for X years, where there is an option to renew after X years at aggregate rates. For X duration, the rates are loaded just as for select processing, where there will be a difference for rates between attained ages with different issue ages. For X+ years, the rates for attained age will be the same for the given criteria combination.

For example, to create the rates, the issue ages are in a row, the duration is in a column and the Term period is 10 years, during which time select processing will be used. Starting with the initial age, the system will traverse through the column for 10 durations. After that, it will find the attained age. Once one issue age is complete, the system must process the next issue age in the same manner. The system will continue in this manner until all issue ages are complete. Only after that is the Criteria Combination finished, and the next Criteria combination can begin.

You are here: Admin Explorer > Rates > Upload Rates > Upload Rates for Aggregate

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Upload Rates from Worksheet

As of Version 9.6, the rate upload process for both Aggregate type rates and Select or Ultimate type rates are the same. For each rate table that is uploaded, one workbook is required. Each worksheet in the workbook will be applicable for each rate set. The first sheet should be set up for the rate legend that provides the metadata on the rate group, such as:

- type of rate table
- the required number of rate table records
- the criteria applicable for the rate table.

Each rate set in the workbook needs to provide:

- the unique set of effective date
- active from date
- active to date
- the involved rate criteria 1 to criteria 10.

Depending on the type of rate table that is uploaded, Aggregate or Select/Ultimate, the information on the legend work sheet and the individual rate sets needs to be different. Refer to [Prepare Spreadsheet for Rate Upload](#) for details of the rate workbook set up.

Upload Rates From Workbook in an Excel Spreadsheet

1. Navigate to [Admin Explorer](#).
2. Right-click on the **Rate Groups** folder.
3. Select **Update Rate Group**. The Update Rate Group Wizard will open.
4. Select the radio button at the top of the window to identify either **Manual**, where only the rate group is created, or **Import**, where a spreadsheet of rates is imported.
 - **Manual**: if this is selected, then enter the rate group information in the fields provided. Click Finish to complete the manual update. You do not need to complete the rest of the steps listed below.
 - **Import**: if this is selected, then complete the additional steps listed below.

For Import Only:

5. In the Workbook field, select the Excel spreadsheet containing the rates you want to upload. Enter the full path of the spreadsheet or browse and select it.
6. Click **Validate Rates**.
 - Check the first worksheet for metadata in the required format.
 - Check the remaining worksheets to make sure the rate information is in sync with the metadata in the first worksheet.
 - Ignore any empty worksheets.



If any errors occur during the validation, then a message will appear. If there are multiple messages, then they can be copied into the system clipboard and pasted to a different application. This allows a user to refer to the list of messages while cleaning up the upload worksheet.

8. After completing the validation, click **Upload Rates**. This button is

enabled only when the rates have been successfully validated with no errors.

9. Click **Finish**.

A **Cancel** button is also available in the Update Rate Group Wizard. During the rate load execution, the entire wizard page with the exception of the Cancel button, will be disabled. While the rate load is executing, a window with a progress bar will be presented on the screen as an “in process” advisory to the user. If Cancel is clicked during the rate load, the progress bar window will close, the upload and delete process will halt, and all rate data will be rolled back to its state prior to the Cancel.

If Cancel is clicked during the validation process, the validations will halt and the user will be expected to start the process over

Once the rates have been uploaded, the [Configuration Area](#) will open and display the new rate table. Check-out the XML file if changes need to be made to the rate table information.



Rates can be [manually edited](#) if they were uploaded incorrectly.

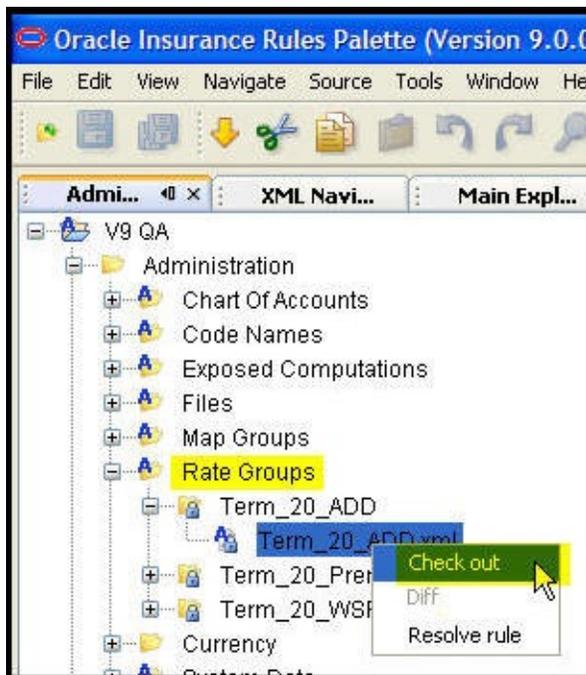
You are here: Admin Explorer > Rates > Manually Enter Rates

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Manually Enter Rates

Once a rate group has been [created](#) rates can be manually entered. This process is typically used for rates used in valuation routines. To edit the data or enter rates manually, right-click on the appropriate **rates.xml** file from the **Rate Groups** folder in **Admin Explorer**. Check-out the XML file.

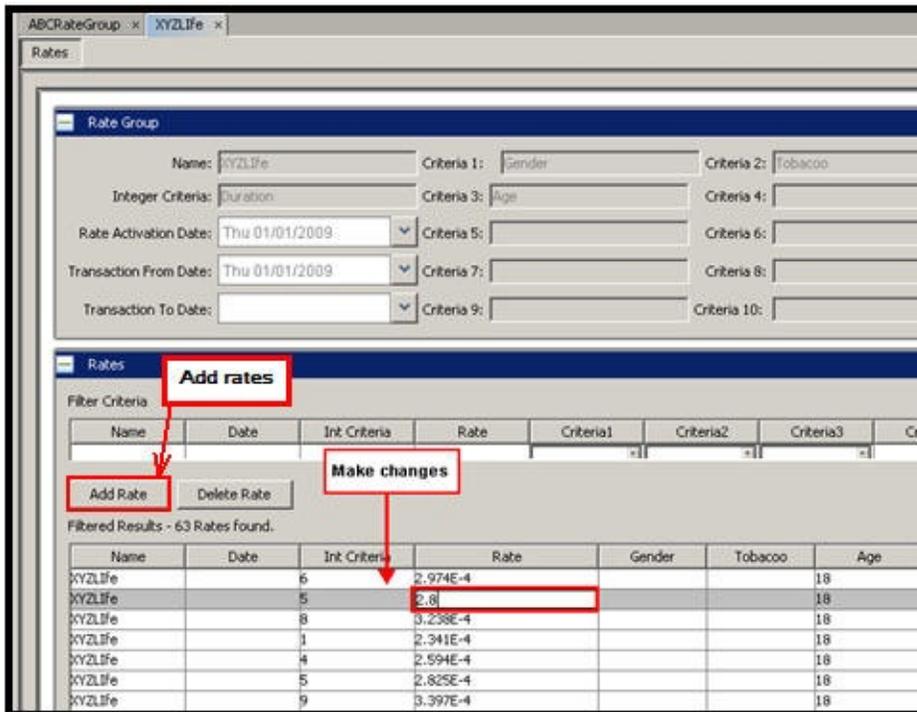
For special occasions when rates will fluctuate based on external factors, rates can be entered using an [activity](#).



Rate Group Right-Click Options

Steps to Manually Enter Rates

1. Check-out a rate group from the **Admin Explorer** in the **Administration | Rate Groups** folder.
2. Click **Add Rates** in the Rates section of the screen.
3. Double-click in the new row and enter the **Rate Group Name, Date, Initial Criteria, Rate** and any other **Criteria**.
4. Check-in the file to save the changes to the database.



Rates Entered Manually in Rules Palette

You are here: Admin Explorer > Rates > Create Fluctuating Rates Using a Transaction

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create Fluctuating Rates Using a Transaction

Some rates fluctuate based on external factors that change regularly and are built into the policy contracts provided to customers. In OIPA, an initial set of rates can be loaded into the system and used as a basis for a new set of rates that are dependent on the original rates and the external factors. The generated rates apply for a certain time frame and are either used by existing and new policies as they age through that time period or are only used by new policies added during the time period. As external factors cause more rates to be generated that apply to policies, OIPA can create a set of production rates (or non-production rates if in a development or test environment). New rates are created for several purposes:

- to mark the effective date for the rates
- to set the criteria
- to add new generated rates
- to maintain and possibly terminate older rate groups.

A transaction is the vehicle used to initiate and control the creation of these new rates. The math section of the transaction uses the RATEARRAY and RATE math variables to obtain the old rates, which are modified by formula, and loaded into a Collection. An attached CreateRates business rule takes the Collection of rates and creates a rate group data table and then populates the table with the rate records. The entire process occurs when the transaction is processed as an activity in OIPA.

System validations within OIPA are used to maintain date relationships. OIPA finds the latest Rate Group by finding the greatest ActiveFromDate. If the new Rate Group's effective date is equal to the effective date from the latest Rate Group and the ActiveToDate is null, then the CreateRates rule updates the ActiveToDate of the latest Rate Group to the date value in the <ActiveFromDate> element. Existing rate groups can be closed to future activities. This will happen automatically but can be controlled by the user by providing a different effective date for the new rate group. The different effective date will not cause the automatic closure of the older

rate group.

Rates With Special Requirements

In the event that rates change based on special circumstances, an activity can be configured to calculate rates and set up the data as defined in an attached [CreateRates](#) business rule. Existing rates can be retrieved using RATEARRAY or RATE math variables. OIPA's math engine calculates rates and loads them into a collection that can be used by CreateRates. The rule creates a rate group data table and populates the table with the newly calculated rates. Typical activity processing adds the rates when an activity is added and processed. Two events typically occur when the activity is processed. First, a new rate group for an existing rate table is created. Then, existing rates groups may be closed as indicated by configuration.

Refer to the [Insurance Rate prototype](#) for a complete explanation of how to configure the activity and attached CreateRates business rule.

You are here: Admin Explorer > Rates > Retrieving Rates

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Retrieving Rates

OIPA's standard rate structure allows rates to be defined with up to 10 text based criteria and 1 numeric (integer) criterion. The criteria provide a business friendly name such as "Risk" class, "Tobacco" class, etc. as needed by any specific rate. It is a configuration standard at Oracle that a business purpose be reserved for the same criteria across all rates where that purpose is applicable. This will aid re-use of configuration wherever possible. For instance, Criteria4 may be reserved to hold "Risk" class values across all rates where risk class is needed to differentiate rates. For a rate where risk class is not a differentiating piece of information, Criteria4 may be re-used for another business purpose if all other criteria have been assigned a business purpose by this and other rates. The 1 numeric criterion (integer criteria) is reserved for a duration or age purpose and is always required in a rate's structure.

OIPA provides historical maintenance through versioning a rate table. Versioning is accomplished by an effective date, a date when the version legally starts, and active from/to dates - a date range that processing must fall within. The active from date, when different than effective date, allows for a window of time when a new version of a rate is accessible prior to its legal start. This may be applicable for billing ahead of a rate change or for illustrative processing into the future. This combination of naming a rate table, providing business names to criteria and versioning the rates via dates is a Rate Group in OIPA.

Following the establishment of a Rate Group, rates may be uploaded to the database. The rates must be associated to a unique combination of criteria values (including the integer criterion, a required criterion) defined by the Rate Group. Once the rates are loaded, they are accessible in Math configuration through the [RATE](#) and [RATEARRAY](#) math variable types.

Retrieval Methods

When OIPA encounters a [RATE](#) or [RATEARRAY](#) math variable type, it executes one of three access methods.

- **OIPA's built-in rate access to the standard rate tables:** this is the default method of rate retrieval. This method will be used unless one of the two rate retrieval system properties is present.
- **Access through a stored procedure:** the property **NamedStoredProceduresRateBll.config.filePath** allows a stored procedure to be used to access rates. Refer to the 10.1.2.0 Documentation library on [OTN](#) for a complete explanation of this property.
- **Access through the extension framework:** the property **rateRetriever.className** allows an extension to be used to access rates. The property identifies the extension by its fully qualified class name. OIPA passes parameter information to the designated extension. The extension may return one rate or multiple rates as a result of the RateGroup's structure and the parameters provided by the RATE or RATEARRAY math variables. If the appropriate result is not returned, an exception will be thrown. Refer to the 10.1.2.0 Documentation library on [OTN](#) for a complete explanation of this property.



Modifications to the System Properties file requires a restart of the application.

Steps to Retrieve Rates

Make sure the System Properties file has the properties for stored procedure or extension framework if one of those methods is used during rate retrieval. Refer to the section above for the specific details on retrieval methods.

1. Define [rate table](#). Rate tables can be user-defined tables stored in the AsRate and AsRateGroup database tables.
2. Determine the method of access.
 1. use direct rate retrieval.
 2. [use a stored procedure](#).
 3. [use the extension framework](#).



RATE and RATEARRAY can be used with stored procedures and/or extension framework. If custom tables are needed, a service request must be created for support from the Oracle Integration team.

The last two access methods require new properties to be added. Modification to the SystemProperties file requires a restart of the application.

The client also has the option of using their own custom tables via stored procedures or web services. However, if the client uses custom rate tables, OIPA's internal features would not be available (i.e. Rate Table viewing in Rules Palette or OIPA.)

Steps to Filter Rates

1. Navigate to the **Admin Explorer**.
2. Open the **Administration** folder.
3. Open the **Rate Groups** folder.
4. Double-click on the folder of the rate group you want to work with.
5. Right-click on the XML file and select **Check-out**.
6. Scroll down to the Rates section of the screen.
7. Select a filter criteria from any column shown in the filter criteria section. For example, if you want to see all male and female non-smokers age 27 with a specific Tobacco class, you would set your filter as shown in the image below.
8. Close the Rate Editor when finished.

The screenshot shows the 'Rate Group' configuration area. The 'Filter Criteria' section contains a table with columns: Name, Date, Int Criteria, Rate, Gender, Tobacco, Risk, and IssueAge. The 'Gender' column is highlighted with a red box, 'Tobacco' with a blue box, and 'IssueAge' with an orange box. Below the table are buttons for 'Add Rate' and 'Delete Rate', and a pagination control showing 'Current Page: 1' and 'Max Results: 10'. The 'Filtered Results' section shows two rows of data for 'Term_20_Premium' with different 'Int Criteria' values (49 and 43) and corresponding 'Rate' values (82.32 and 27.17). The 'Gender', 'Tobacco', and 'IssueAge' columns in the results table are also highlighted with red, blue, and orange boxes respectively.

Name	Date	Int Criteria	Rate	Gender	Tobacco	Risk	IssueAge
Term_20_Premium		49	82.32	01	02	03	27
Term_20_Premium		43	27.17	02	02	03	27

Rate File Open in Configuration Area

You are here: Admin Explorer > Rates > Repricing

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Rate Repricing

Sometimes a re-price may occur after the initial product is launched. The processes and structure of the product may not change, but the re-price could change some aspects of the coverage such as the base coverage rates. Rates used for valuation routines do not fall into this category and would not apply in this situation. Repricing applies to rates that represent a cost to the policy. It is restricted to certain limits by the policy contract and by plan filing with governmental jurisdictions.

The AsRate table holds the detail rates. The AsRateGroup table is used to control when a set of rates in AsRate will be used by a calculation. So in regards to re-pricing, a new complete set of AsRate records is needed for the new rate group.

There are five columns in AsRateGroup that are the most important factors OIPA considers when identifying the correct set of rates that a policy will use during execution. They are the RateDescription, EffectiveDate, ActiveFromDate, ActiveToDate and RateGroupGUID. An explanation of each column is provided below.

AsRateGroup Column Explanation

RateDescription: The rate description for a rate must have the same value across all of the rate group records.

EffectiveDate: The EffectiveDate is compared to a segment's RateActiveDate or a policy's Plan Date or Issue Date; whichever represents the effective date of the policy. The system selects the rate group with the highest EffectiveDate value that is less than or equal to the provided RateActiveDate. At no time can two sets of rates be available for a given RateActiveDate. And there must always be at least one rate set for each RateActiveDate.

Line	RateDescription	EffectiveDate / RateActivationDate
1	X	1/1/2004
2	X	1/1/2009

Effective Date Example for Repricing

Based on the table above, if a RateActivationDate is 12/31/2008, then line 1 represents the rate set that is selected. If a RateActivationDate is 1/1/2009, then line 2 represents the rate set that is selected. Notice that there are two sets of rates with an EffectiveDate less than or equal to 1/1/2009, but the system will only return one set (the set with the highest effective date). If the RateActivationDate is 12/31/2003, then an error will occur within the calculation that tries to use the rates.

Near the end of an older set of rates and at the beginning of a new set of rates, there may be questions as to which rate set the policy should use. Once the set is decided, it is a matter of setting the value of SegmentRateActiveDate appropriately. For example, a policy may be dated on or after 1/1/2009, but the user has the flexibility to select the new rates by providing a value of 1/1/2009 or later in the segment's rate active dates or the user can select the old rates by providing a value of 12/31/2008 or earlier in the segment's rate active dates. These dates for rate access are set by the example and will differ from the dates used by the actual rate data.

ActiveFromDate and ActiveToDate: As the name implies, these two dates define a date range where rates are available based on an activity's effective date. A set of rates will not be available to the activity if the activity's effective date is not within the date range. Being **within the date range** is defined as a date being greater than or equal to ActiveFromDate and less than ActiveToDate. If ActiveToDate is null, then the latter comparison is always true. No two date ranges for the same rate may overlap. For example, in the table below, lines 1 and 2 are valid. Lines 3 and 4 are invalid. Line 5 is valid since it does not overlap with either line 3 and 4. Lines 6 and 7 are valid, but 1 day is missing from the data ranges (12/31/2008). This will pose a problem for activities that use rate description Z and have activity effective dates of 12/31/2008.

Line	RateDescription	ActiveFromDate	ActiveToDate
1	X	1/1/2004	1/1/2009
2	X	1/1/2009	null
3	Y	1/1/2004	1/1/2009
4	Y	1/1/2006	1/1/2009
5	Y	1/1/2009	null
6	Z	1/1/2004	12/31/2008
7	Z	1/1/2009	null

ActiveFrom and ActiveTo Examples

In the table below, a new column of information is added to the table, **EffectiveDate**. The first 7 lines are a copy of the prior table. They are valid and invalid for the same reasons above. Lines 8 and 9 are added to show more invalid dating. The ActiveFromDate and ActiveToDate columns look valid. It is the EffectiveDate in combination with these dates that make the lines invalid. Activities dated between 1/1/2004 and 1/1/2009 where the segment's rate activation date is less than 1/1/2009 will have no problem processing. However, once the activity's effective dates become equal to or greater than 1/1/2009, policies with segment rate activation dates less than 1/1/2009 will not be able to process. Line 8 is not valid for rate access on activities dated 1/1/2009 or later with segment rate activation date less than 1/1/2009. Line 9 has an effective date of 1/1/2009. This tells the system that the segment activation date must be 1/1/2009 or greater, leaving earlier issues without rate access. To become valid, line 9's EffectiveDate should be 1/1/2004 or an

additional record with an EffectiveDate of 1/1/2004 is needed to accommodate the earlier issues or the ActiveToDate on line 8 should be null leaving rate access for these early issues at line 8.

Line	RateDescription	EffectiveDate / RateActivationDate	ActiveFromDate	ActiveToDate
1	X	1/1/2004	1/1/2004	1/1/2009
2	X	1/1/2004	1/1/2009	null
3	Y	1/1/2004	1/1/2004	1/1/2009
4	Y	1/1/2004	1/1/2006	1/1/2009
5	Y	1/1/2004	1/1/2009	null
6	Z	1/1/2004	1/1/2004	12/31/2008
7	Z	1/1/2004	1/1/2009	null
8	A	1/1/2004	1/1/2004	1/1/2009
9	A	1/1/2009	1/1/2009	null

Effective Date Column Added to Example

How Repricing Works

In order for OIPA to use a set of different rates from the original product, you must use a combination of the EffectiveDate, ActiveFromDate and ActiveToDate in the AsRateGroup table. The other data in the AsRateGroup table will remain the same. The date information, along with the Rate Description and criteria for the Rate Group, is used by a configurator to point to the correct stored values in the AsRate table. Regardless of how many entries for a single Rate Description are entered into AsRateGroup, the differentiating rate group information needs to be entered to ensure that only one set of rates will be valid for any policy or segment for any given day.

EffectiveDate for a given RateDescription will determine the RateGroup or RateGroups based on a date data attribute of the Policy or Segment. This is most commonly a PolicyField or a SegmentField.

Once a set of RateGroups is retrieved that have the required EffectiveDate (there may be a few at this point), the list is further narrowed down based on the Activity date, which will use ActiveFromDate and ActiveToDate to determine the applicable rates.

RateGroup selection SQL Pseudo Code that is generated:

```
SELECT * FROM AsRateGroup
WHERE rateDescription = [RateDescription]
AND effectiveDate <= [EffectiveDate]
AND activeFromDate <= [ActivityDate]
AND ( activeToDate >= [ActivityDate] OR activeToDate is null )
```

Repricing for New Policies -vs- Repricing for Existing and New Policies

By using Rate Groups, actuaries are given the opportunity to do one of the following:

- Re-price a product for an existing benefit that sets new rates for a product, but the new rates are used only for new policies after a given date. Existing policies will still use the old set of rates. See **Example One** below.
- Re-price a product where the old rates are no longer used from a certain date forward, and both old and new policies use a set of rates from a given date. [See Example Two below.](#)

In either case, only one set of rates is valid for a policy at any one time.

Example One: Two Sets of Rates are Open

Re-price a product for an existing benefit that sets new rates for a product, but the new rates are used only for new policies after a given date. Existing policies will still use the old set of rates.

- For example, a base segment was priced using rate set A with an original effective date of 1/1/2000 and no backdating. The Rate Group name is BaseRates. It is determined four years later that the rates should be increased to be more profitable with a re-price effective date of 1/1/2004, again with no backdating. Further suppose that it is decided that already existing policies should not be affected and they should continue to use the existing BaseRates rates that have been loaded.
- In this case, BaseRates is the appropriate Rate Description for the rates and by continuing to use the same description, configuration to perform the rate lookup will not need to change as the RateDescription will stay the same.
- In the database, two entries will exist in AsRateGroup. Both are called BaseRates but with different EffectiveDates, which are also known as Rate Activation Dates.
- The second set of rates will follow the same Rate Upload procedure as above. The RateDescription used for the second set of rates will be BaseRates, the same name as the original set of rates, but will be loaded with a different Rate Activation Date and a different ActiveFromDate.
- After entering the Rate Group information, AsRateGroup would hold the following information. For this example, no criteria is used other than IntegerCriteria. If additional criteria were used, all criteria names must match between the old and new rates.

Mapping it Out

Line one is used for RateDescription=BaseRates where policies have an EffectiveDate from 1/1/2000 (inclusive) to 1/1/2004 (exclusive).

Line two is used for RateDescription=BaseRates where policies

have an EffectiveDate from 1/1/2004 (inclusive) forward.

Line	RateDescription	EffectiveDate/ RateActivationDate	ActiveFromDate	ActiveToDate
1	BaseRates	1/1/2000	1/1/2000	
2	BaseRates	1/1/2004	1/1/2004	

Sample of Rate Effective Dates

Backdating

Should backdating of activities be allowed, then the ActiveFromDate would not match the EffectiveDate/RateActivationDate. An example where three months of backdating is allowed would look like this:

Line	RateDescription	EffectiveDate/ RateActivationDate	ActiveFromDate	ActiveToDate
1	BaseRates	1/1/2000	10/1/1999	
2	BaseRates	1/1/2004	10/1/2003	

Sample of Rate Backdating

	RateGroupGUID	RateDescription	IntegerCriteria
1	B2225C5B-8CF0-056F-1184-02BCDAA322B6	BaseRates	Age
2	4DBF1E7F-1777-3928-76EB-F1CDA976A26A	BaseRates	Age

	Criteria1	Criteria2	Criteria3	Criteria4	Criteria5
1	(null)	(null)	(null)	(null)	(null)
2	(null)	(null)	(null)	(null)	(null)

	Criteria6	Criteria7	Criteria8	Criteria9	Criteria10	StatusCode
1	(null)	(null)	(null)	(null)	(null)	(null)
2	(null)	(null)	(null)	(null)	(null)	(null)

	TypeCode	EffectiveDate	ActiveFromDate	ActiveToDate	XMLData
1	(null)	1/1/2000 12:00:00 AM	1/1/2000 12:00:00 AM	(null)	(null)
2	(null)	1/1/2004 12:00:00 AM	1/1/2004 12:00:00 AM	(null)	(null)

Rate Database Samples

- For activities on policies with an EffectiveDate between 1/1/2000 and 12/31/2003, the original rates will always be used.
- For activities on all policies whose EffectiveDate is on or after 1/1/2004, the new re-priced rates will be used.

Example Two: Only One Set of Rates is Open for BaseRates

Re-price a product where the old rates are no longer used from a certain date forward, and both old and new policies use a set of rates from that given date.

- For example, a base segment was priced using rate set A and named Rate Group BaseRates. It is determined four years later that the rates should be increased to be more profitable. Further suppose that it is decided that already existing policies should also have their rates increased and their current rates will no longer be valid as of a given date.
- In this case, BaseRates is the appropriate Rate Description for the rates. Since you are continuing to use the same description, you will configure the rate lookup the same way.
- In the database, there will exist two entries in AsRateGroup, both called BaseRates but with different EffectiveDates. In addition, the rates that will no longer be in affect will have the ActiveToDate filled in.
- After entering the Rate Group information, AsRateGroup will hold the following information. For this example, no criteria is used other than IntegerCriteria. If criteria were used, all criteria names must match between the old and new rates.

Mapping it Out

Line one is used from RateDescription=BaseRates where policies have an EffectiveDate from 1/1/2000 (inclusive) forward but only for activities with effective dates between 1/1/2000 (inclusive) and 12/31/2003 (inclusive).

Line two is used from RateDescription=BaseRates where policies have an EffectiveDate from 1/1/2000 (inclusive) forward but only for activities with effective dates on or after 1/1/2004.

Line	RateDescription	EffectiveDate RateActivationDate	ActiveFromDate	ActiveToDate
1	BaseRates	1/1/2000	1/1/2000	12/31/2003
2	BaseRates	1/1/2000	1/1/2004	

Sample of Rate Backdating

	RateGroupGUID	RateDescription	IntegerCriteria
1	B2225C5B-8CF0-056F-1184-02BCDAA322B6	BaseRates	Age
2	4DBF1E7F-1777-3928-76EB-F1CDA976A26A	BaseRates	Age

	Criteria1	Criteria2	Criteria3	Criteria4	Criteria5
1	(null)	(null)	(null)	(null)	(null)
2	(null)	(null)	(null)	(null)	(null)

	Criteria6	Criteria7	Criteria8	Criteria9	Criteria10	StatusCode
1	(null)	(null)	(null)	(null)	(null)	(null)
2	(null)	(null)	(null)	(null)	(null)	(null)

	TypeCode	EffectiveDate	ActiveFromDate	ActiveToDate	XMLData
1	(null)	1/1/2000 12:00:00 AM	12/1/1999 12:00:00 AM	12/31/2003 12:00:00 AM	(null)
2	(null)	1/1/2000 12:00:00 AM	1/1/2004 12:00:00 AM	(null)	(null)

Rate Database Samples

- Activities on policies between 1/1/2000 and 12/31/2003 whose RateActivationDate is between 1/1/2000 and 12/31/2003 will use the original rates.
- Activities on all policies that occur on or after 1/1/2004 will use the new re-priced rates.

You are here: Admin Explorer > Requirements > Requirements Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Requirements Overview

Requirements can be defined in the Rules Palette and then attached to specific transactions. When those transactions process as activities in OIPA, the requirement conditions must be met in order for the activity to process. If the requirement conditions are not met, then a  requirement icon is generated next to the activity and the activity will remain in a pending status. Once the requirement is met, the activity can be processed.

See the [Requirements in New Business Underwriting](#) page for information specifically regarding requirements in the NBU process.

Set Up Requirements

There are two steps involved in setting up requirements:

1. [Create a requirement definition](#). The AsRequirementDefinition table holds the information entered while creating a new requirement definition.
2. [Configure the requirement](#). There are two additional requirement tables that hold requirement information entered when configuring the requirement. They are AsRequirementCriteria and AsRequirementGroup.

Attach Requirement to a Transaction

After requirements are created and configured, they can be associated with transactions. The requirements will prevent the transaction from processing as an activity in OIPA until all criteria are met.

Requirements are associated with transactions through the [GeneratePendingRequirements and DeliveryRequirements](#) business rules. Attach these two rules to the transaction. Then open the attached rules and configure them as needed.



When adding the requirement business rules to a transaction, **GeneratePendingRequirements** **MUST ALWAYS** appear in the rule list before **DeliveryRequirements**.

You are here: Admin Explorer > Requirements > Create Requirements

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create Requirements Definition

Requirements can be created to set conditions around the processing of an activity in OIPA. When an activity is processed with requirements attached, the activity is held until the requirements are satisfied.

Requirements are created in the Admin Explorer tab of the Rules Palette. This is a two-step process. First, create the requirement definition and then configure the requirement.

Creating a Requirement Definition

1. Open the Admin Explorer tab.
2. Open the Administration folder.
3. Right-click on the Requirements folder and select **New Requirement Definition**. The New Requirement Definition Wizard will open.
4. Select a requirement level for the requirement. The available levels are Activity, Policy and Policy/Client. The code names for these requirement levels are configured in the AsCodeRequirementLevel code name. Refer to the [Codes](#) page for additional information.
5. Type a requirement name.
6. Select a requirement category. The requirement categories are configured in the AsCodeRequirementCategory code name. You can add or edit these options from the Code Names folder in the Admin Explorer. Refer to the [Codes](#) page for additional information.
7. Select a requirement severity. The requirement severity options are configured in the AsCodeRequirementSeverity code name. You can add or edit these options from the Code Names folder in the Admin Explorer. Refer to the [Codes](#) page for additional information.
8. Enter an integer value in the **Results Obsolete Days** field. This value designates for how many days the requirement results should be valid.
9. If needed, select **Yes** or **No** from the **Manual Results** field. This field is optional. If set to Yes, the user will need to manually match results to the requirement. Selecting Yes will also allow the user to be to edit the fields on the Requirement Results screen in OIPA.
10. Type a requirement description. This is what will display in OIPA when the requirement information is displayed.
11. If needed, enter a message in the **Message** field. This field is optional.
12. Click **Finish**. The requirement will be listed under the Requirements folder in blue text. This means it has not been checked-in or saved to the database.
13. [Configure](#) the requirement.

14. Check the file in to save the requirement information to the AsRequirementDefinition database table.

Editing a Requirement Definition

If changes are needed for a requirement definition, they can be made from the Admin Explorer tab. To edit a requirement definition:

1. Open **Administration | Requirements** and expand the relevant requirement node.
2. Expand the relevant requirement node.
3. Expand the relevant override node.
4. Expand the node for the requirement override to be edited.
5. Right-click on the XML file and select **Check out**.
6. If necessary, edit the information on the General pane. The following fields are able to be edited:
 - Requirement Level: Click the field to select a level.
 - Requirement Category: Click the field to select a category.
 - Results Obsolete Days: Double-click the field and enter an integer.
 - Manual Results: Click the field to select a value.
 - Requirement Description: Click in the field to update the description.
 - Message: Double-click in the field to update the message.
7. Click on the **XML Definition** pane to edit the requirement's configuration.
8. Check the file in to save the changes to the database.

Creating a Requirement Override

Requirements can be overridden at the primary company, subsidiary company, Product (if enabled), plan or state. Requirement overrides are created from the Admin Explorer tab. To create a requirement override:

1. Open **Administration | Requirements**.
2. Right-click on the requirement that is to be overridden and select **New [requirement name] Override**.
3. The New Requirement Override wizard will open. Click **Next**.
4. Select all of the relevant override settings and click **Finish**.

You are here: Admin Explorer > Requirements > Configure Requirements

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Configure Requirements

After a requirement definition is created, additional criteria and can be configured. A visual editor is available to assist with requirement configuration.

Requirements Visual Editor

When a requirement is checked out, the visual editor will open in the Configuration Area. There are five panes available to assist with visual editing. Each pane is described below.

General Pane

This pane displays the information entered when creating the requirement definition. Much of the information on this pane cannot be edited, but you can double-click in the other fields to update information. The Requirement Description is what the OIPA user will see when requirement information is displayed.

Fields Pane

This pane is similar to the [Fields](#) pane for transactions, with the exception that multifields are not applicable to requirements. The fields defined here will display on the Requirement Detail window in OIPA. This window opens when the requirement icon next to an activity is clicked. Then right-click on a requirement and select Requirement Detail. The fields are populated with values from the GeneratePendingRequirements business rule.

Events Pane

This pane is similar to the [Events](#) pane for transactions. Fields must be configured if events are used.

Criteria Pane

There are two sections on this pane: Requirement Groups and Requirement Criteria. These sections can be collapsed by clicking the  Collapse icon or expanded by clicking the  Expand icon.

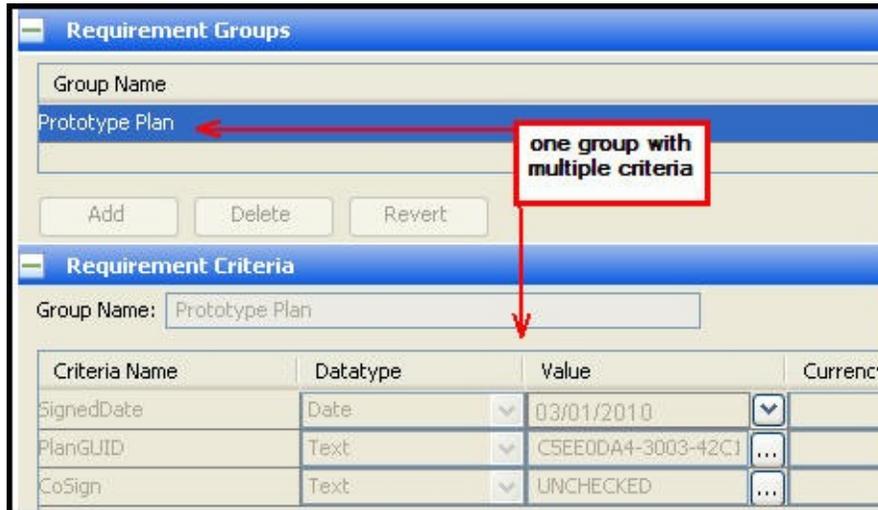
- **Requirement Groups:** provides a method of attaching multiple criteria together. Each group listed will have its own set of criteria defined in the Requirement Criteria section below. You will only see the criteria associated with the group you have selected. Clicking a different group will reveal the specific criteria assigned to that group. You can create a group without any associated criteria. Groups are

saved to the AsRequirementGroup table.

- **Add:** The Add button adds a new row. Double-click in the row to add the group information. Click **Save** on the Tool Bar to save the new group or just check-in the file.
- **Delete:** The Delete button removes the selected group and all associated criteria. A confirmation window will appear to verify the delete. After the confirmation is verified, the Delete button changes to UnDelete. A group is deleted immediately if it was recently added and the file has not been checked-in. If the group existed previously, then the delete happens after the file is checked back in.
- **Revert:** The Revert button reverts all changes made to the Group section.
- **Requirement Criteria:** used as a reference for the system when it is determining whether or not to generate requirements for an activity. These are the fixed criteria the math variables are matched to in the GeneratePendingRequirements business rule in order to determine if the requirement is needed. You can associate multiple criteria with a Requirement Group. Criteria are saved to the AsRequirementCriteria table.
 - **Add:** The Add button adds a new row. Double-click in the row to add the criteria information. Click **Save** on the Tool Bar to save the new criteria or just check-in the file.
 - **Delete:** The Delete button removes the selected criteria. The Delete button changes to UnDelete when clicked. A criterion is deleted immediately if it was recently added and the file has not been checked-in. If the criterion existed previously, then the delete happens after the file is checked back in. When UnDelete is enabled, click it to remove the Delete check mark next to a row. This allows a delete to be reversed before the file is checked-in.
 - **Revert:** The Revert button reverts all changes made to the Criteria section.

Requirement Criteria should be configured to make the set of criteria

within each Requirement Group mutually exclusive with respect to each other Requirement Group's set of Criteria. This would make it impossible for any given activity to satisfy more than one set of criteria. This is often done by using PlanGUID as one of the criteria in each group but it can be done using other methods as well.



Criteria Pane of a Requirement in Admin Explorer

XML Source Pane

This pane displays the XML source code for the Requirement Details screen. There are several important points concerning requirement XML:

- The default XML for a requirement is just an empty `<RequirementDefinition>` parent tag. No further XML configuration is necessary to check in the requirement.
- The requirement cannot be checked in if this tag does not exist. It must be present in the XML Source pane.
- Fields must be configured if events are needed.

Navigate to **Requirements | Requirement Detail** in the XML Configuration Guide for more information on configuring this pane.

XML Definition Pane

This pane contains the necessary configuration for processing a single requirement. There are a few important points concerning requirement definition XML:

- The default XML for a requirement definition is just an empty `<Requirement>` parent tag. No further XML configuration is necessary to check in the requirement.
- The requirement cannot be checked in if the parent tag does not exist.

Navigate to **Requirements | Requirement Definition** in the XML Configuration Guide for more information on configuring in this pane.

XML Results Pane

This pane contains the configuration for the Requirement Results screen. The requirement results on the Requirement Result screen may be formatted as a collection of fields, a text document or a table depending on the type of information contained within the result. There are a few important points concerning requirement result XML:

- The default XML for a requirement result is just an empty `<RequirementResults>` parent tag. No further XML configuration is necessary to check in the requirement.
- The requirement cannot be checked in if the parent tag does not exist.

Navigate to **Requirements | Requirement Results** in the XML Configuration Guide for more information on configuring this pane.

Configuring a Requirement

1. From the Admin Explorer tab open **Administration | Requirements** and expand the relevant requirement node.
2. Expand the node for the relevant override level and check out the requirement.
3. Click on the **Criteria** pane.
4. Click **Add** in the Requirement Groups section and click in the highlighted field.
5. Type the Requirement Group name. This is required.



Criteria do not have to be added initially. The Rules Palette allows this section to remain empty; however, before the requirement can be used in OIPA, criteria must be added.

6. Scroll down and click **Add** in the Criteria section then click inside the highlighted field. The Add button is only enabled if a requirement group was created.
7. Type the Criteria name. Criteria must have unique names within the same group. The Criteria names can be reused if they are used only once in each group.
8. Select a datatype.
9. Enter a value.
10. Select a Currency if the datatype is money.
11. Provide any additional configuration on the **XML Definition** and/or **XML Results** panes.
12. Right-click on the XML file and select **Check in** to save the changes to the database.

You are here: Admin Explorer > Requirements > Requirements Business Rules

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Requirements Business Rules

There are two business rules that must be [attached](#) to any transaction that uses requirements. They are GeneratePendingRequirements and DeliveryRequirements.

OIPA also needs the ActivityRequirementScreen business rule in order to handle requirements properly. This rule will need to be configured as a screen rule. The global rule should only have an empty opening and closing tag. The actual rule is configured as a company level override.

The **XML Configuration Guide** in the Help menu contains an explanation of the elements, attributes and values used in this business rule. Open that topic from the help icon located on the Tool Bar and search for **Business Rules | Screen Rules | ActivityRequirementScreen**. Use the XML Configuration Guide as a resource when creating the ActivityRequirementScreen business rule.

GeneratePending Requirements Business Rule

GeneratePendingRequirements references the criteria identified in the requirement definition (AsRequirementCriteria) and matches it to the math variable identified in the rule. If all conditions identified in the rule are true, then the requirement is generated.



Refer to the **XML Configuration Guide** in the Rules Palette help menu for a detailed explanation of the elements and values available for configuration in this business rule. The business rule can be found in the **Business Rules | Attached Rules** folder.

```
<GeneratePendingRequirements LEVEL="Activity">
  <PendingRequirement>
    <RequirementName>CoSigner</RequirementName>
    <Criteria NAME="SignedDate" OPERATOR="GreaterThan" DATATYPE="DATE">SignedDateMV</Criteria>
    <Criteria NAME="CoSign" DATATYPE="TEXT">CoSignMV</Criteria>
    <Criteria NAME="PlanGUID" DATATYPE="TEXT">PlanGUIDMV</Criteria>
    <Fields>
      <Field>
        <From>SignedDateMV</From>
        <To>SignedDate</To>
      </Field>
      <Field>
        <From>CoSignMV</From>
        <To>CoSign</To>
      </Field>
    </Fields>
  </PendingRequirement>
</GeneratePendingRequirements>
```

Name of Requirement Definition created in Admin Explorer (points to <RequirementName>CoSigner</RequirementName>)

Criteria defined in requirement definition (points to <Criteria NAME="SignedDate" OPERATOR="GreaterThan" DATATYPE="DATE">SignedDateMV</Criteria>)

Situations that must be true in order for requirement to be added to activity (points to the Criteria and Fields sections)

GeneratePendingRequirements Business Rule XML

DeliveryRequirements Business Rule

DeliveryRequirements business rule serves two purposes:

1. It identifies the fields that are required to have values before a requirement should be considered satisfied. The <RequiredDate> tag accomplishes this in the example below.
2. It toggles a setting that determines whether the activity's effective date should be updated after completion of all pending requirements. The PROCESSIONCOMPLETE attribute accomplishes this in the example below. When it is set to No or left blank, the activity will retain its original effective date. If it is set to Yes, the rule will update the effective date of the activity to the date defined in the NEWACTIVITYDATE attribute once all requirements are complete.



Refer to the **XML Configuration Guide** in the Rules Palette help menu for a detailed explanation of the elements and values available for configuration in this business rule. The business rule can be found in the **Business Rules | Attached Rules** folder.

```
<DeliveryRequirements PROCESSIONCOMPLETE="Yes" NEWACTIVITYDATE="SomeDateValue">
  <RequiredDates TYPE="AND">
    <RequiredDate>OpenDate</RequiredDate>
    <RequiredDate>CloseDate</RequiredDate>
  </RequiredDates>
</DeliveryRequirements>
```

DeliveryRequirements Business Rule XML

Attach Requirement Rules to a Transaction

1. Expand  the associated transaction folder in the Main Explorer.
2. Right-click on the transaction's XML file.
3. Select **Edit Attached Rules**. The rule must be checked-in to see this option.
4. Select the GeneratePendingRequirements and DeliveryRequirements rules. The GeneratePendingRequirements business rule **MUST ALWAYS** be listed before the DeliveryRequirements business rule or processing errors will occur.
5. Use the  button to attach the rules.
6. Select **Finish**.
7. Open the attached rules in the Configuration Area and configure the XML as needed.

You are here: Admin Explorer > Requirements > Requirements Error

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Requirements Error

Once requirements are added through one of Policy Requirements wherein the requirement engine processes the same to basis the definition of the requirement. While processing, if there are errors, the error is logged into AsRequirementError table.

Requirement Error display in Policy Requirements Screen

The Error icon  will display on the Policy Requirement Screen next to any requirement which has error. Hovering over the error icon will display a short description of the error. This Error is a short description of the Requirement Error which is extracted from the stack trace.

Requirement Error Search Screen

Also, there is a separate Requirement Error search screen which can be accessed through the "Requirement Error Search" menu item under the Requirement item in the main menu. The Error message will be displayed only to a user with proper security setup.

The Requirement Error screen allows user to filter the requirements through a set of filters. Also, for any requirement error, this screens displays the details of the error and also allows user to reprocess the requirement. The Process icon will be visible and can be selected if the user has the appropriate security.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Requirement Processing Sample

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Requirement Processing Sample

Requirements can be added to transactions to control the situation that must be present in order for the transaction to process as an activity in OIPA. When the transaction is added as an activity, OIPA will check to make sure the requirement conditions are met before allowing the activity to process. If the requirements are not met, a requirement icon will appear next to the activity and it will not process. The activity will then go to Requirement Pending status rather than Active status.

Scenario

CSR selects the ContractSign activity and enters an Activity Effective Date of Feb 15, 2010. The CoSign Checkbox is left Unchecked. The requirements attached to this transaction require that the activity only process when the Co-sign checkbox is checked or the Effective Date is greater than March 1, 2010. A requirement icon appears next to the activity and it does not process.

Configuration Requirements

There are three AsCode tables created for requirements:

- AsCodeRequirementCategory: holds requirement category options.
- AsCodeRequirementType: holds requirement types.
- AsCodeRequirementStatus: holds requirement status options.

There are seven database tables used for requirements:

- AsRequirement: holds information on existing requirement instances.
- AsRequirementActivity: holds the activity GUID and requirement GUID.
- AsRequirementCriteria: holds the requirement criteria entered when configuring a new requirement.
- AsRequirementDefinition: holds the information entered while creating a new requirement definition.
- AsRequirementField: holds requirement field information.
- AsRequirementGroup: holds the requirement group information entered when configuring a new requirement.
- AsRequirementPolicy: holds the policy GUID and requirement GUID.

There are three business rules that need to be configured:

- ActivityRequirementScreen: the global rule should exist, but does not need to be configured. This rule should be overridden at the company level and attached to the transaction. This rule does not need to contain configuration, as the fixed fields are defined by base Java code.
- DeliveryRequirements: this rule should be overridden at the transaction level and attached to the transaction.
- GeneratePendingRequirements: this rule should be attached to a transaction. It should be listed before the DeliveryRequirements rule in the Attached Rules section.

There is one optional business rule:

- CopyToRequirementFields: this rule is attached to copy data to the requirement fields of another transaction's requirement(s).

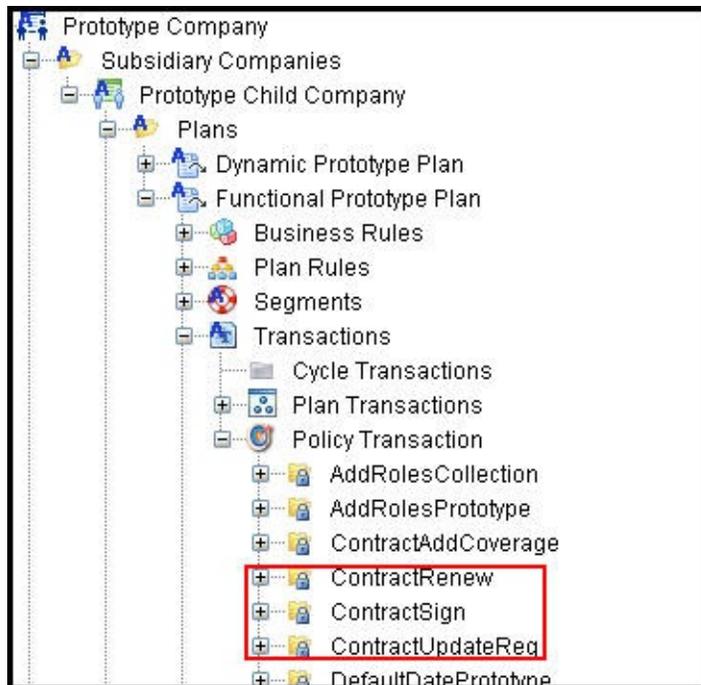
Requirement definitions should be created with specific requirement criteria to satisfy.

A transaction is needed to demonstrate requirements.

Prototype Samples

The following pieces of configuration were added to the Prototype Company to demonstrate requirements processing:

- Three transactions were created. Navigate to Main Explorer and open **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions**.
 - ContractSign: A requirement called CoSign will generate if the activity Effective Date is less than March 1, 2010 and the CoSign checkbox is unchecked. The requirement will not generate if the activity Effective Date is greater than March 1, 2010 and/or the CoSign checkbox is checked. When CoSign is satisfied, ContractSign is configured to update the activity Effective Date to the date when the requirement was satisfied.
 - ContractRenew: This activity always generates a requirement called IRSCheck. The Effective Date remains the same even after the requirement is satisfied. The requirement will generate with a Due Date three months after the Contract Renew Effective Date. The Due Date is not enforced. There is no restriction around adding a Close Date greater than the Due Date.
 - ContractUpdateReq: This activity is used to satisfy the generated requirements for ContractSign and ContractRenew by updating their Close Date. The Effective Date will be forced to the Effective Date of the activity with the requirement that is being updated. This activity will satisfy one requirement: one requirement for one activity.

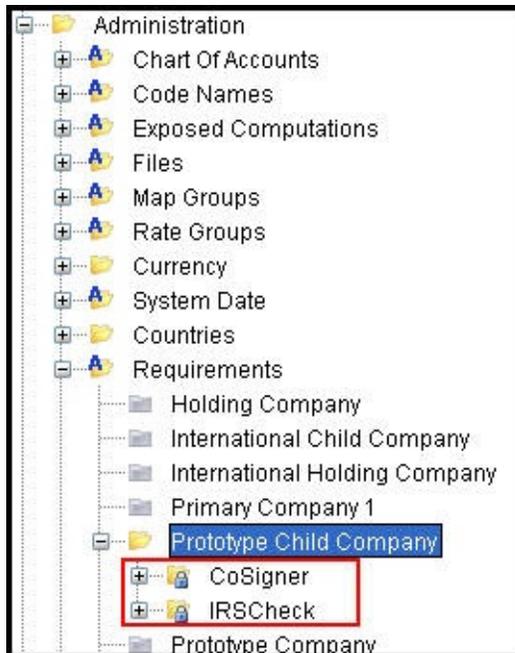


Requirement Processing Prototype Transactions in Main Explorer

- There are two requirement definitions that are used to demonstrate requirement processing. These definitions are referenced from the transactions and represent a requirement that must be filled. The criteria are listed below. Navigate to Admin Explorer and open **Administration | Requirements | Prototype Child Company** to see the requirement definitions.
 - CoSign: this requirement needs three criteria to be met:
 - Date
 - Checkbox: The CoSign requirement will generate if the ContractSign activity Effective Date is less than March 1, 2010 and the CoSign Checkbox is unchecked. The Cosign requirement will not generate if activity Effective Date is greater than March 1, 2010 and/or the CoSign Checkbox is checked. When CoSign is satisfied, ContractSign is configured so that the ContractSign EffectiveDate is updated to the date that the requirement was satisfied.
 - PlanGUID: PlanGuid must match the PlanGUID of the Functional Prototype Plan. This criterion should always be met when processing an activity in the Functional Prototype

Plan.

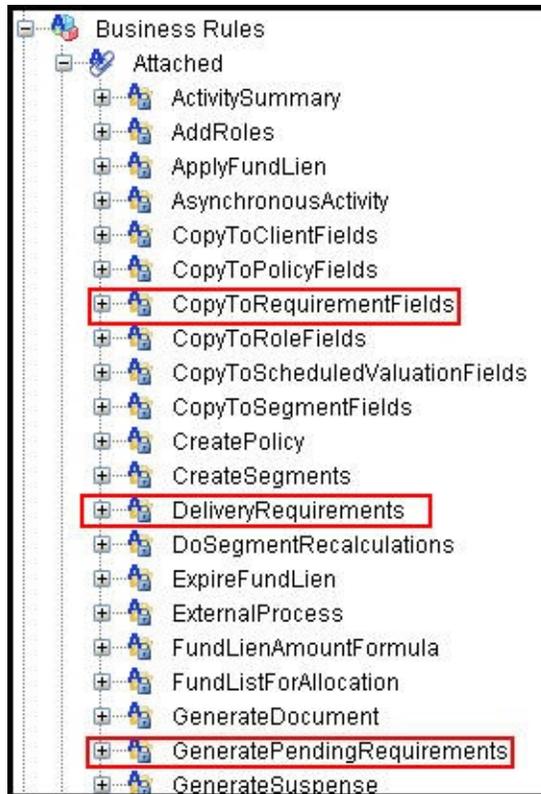
- IRSCheck: this requirement requires only one criteria be met:
 - PlanGUID: PlanGuid must match the PlanGUID of the Functional Prototype Plan. This criterion should always be met when processing an activity in the Functional Prototype Plan.



Requirement Definitions in Admin Explorer

- Three business rules were used to demonstrate requirements processing. Navigate in the Global Rules Explorer to **Business Rules | Attached**. Open the folder for each rule and then open the Transaction Overrides folder.
 - GeneratePendingRequirements: This rule will be attached to both transactions generating requirements. In Contract Renew this rule will be used to add the Due Date. In Contract Sign it will define the conditions under which the CoSign requirement is created. This rule will actually generate the requirement.
 - DeliveryRequirements: This rule will be attached to both Contract Renew and Contract Sign. It will define whether the Effective Date is updated when the requirements are satisfied.
 - CopyToRequirementFields: This rule will be attached to

ContractUpdateReq. This rule will update the Close Date for the requirement that is selected by Contract Update Req to be satisfied.



Requirement Business Rules in Global Rules Explorer

You are here: Admin Explorer > Requirements > Requirements Security

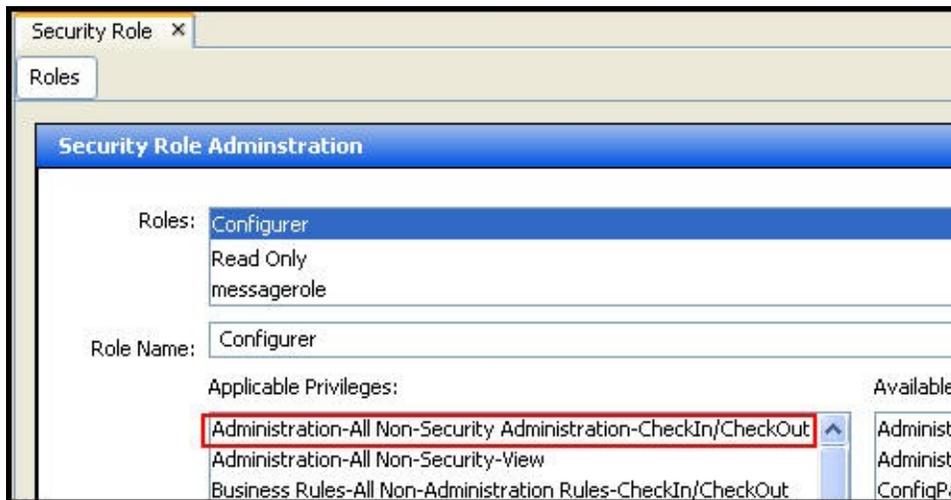
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Requirements Security

Requirement security should be set-up for Palette users and OIPA users. Security is set through the Admin Explorer tab in the Security folder. There is a node for Palette Security and Application Security.

Palette Security

From the Admin Explorer tab, open the Palette Security folder. Double-click the Security Role file to open it in the Configuration Area. Make sure **Administration - All Non-Security Administration - CheckIn/CheckOut** is in the Applicable Privileges column.



Security Privilege in Security Role for Requirements

Application Security

Several new buttons were added to the security group file to specify which users can work with requirements in OIPA . In the Admin Explorer tab, open the **Security | Application Security | Security Groups** folder. Double-click a security group to open it in the Configuration Area. Check the box next to the following buttons to grant access to requirements in OIPA.

- **ActivityRequirement:** grants access to the ActivityRequirementScreen. If this is not checked for any transaction, then the user will not see the requirement icon for that transaction.

requirement icon controlled by security

Requirements Icon in OIPA

- **RequirementDelete:** grants access to the RequirementDelete right-click option on the ActivityRequirementScreen. The Activity Requirement screen is accessed by clicking the requirement icon next to an activity. When the screen opens, right-click over the name of the requirement and the delete option should display unless security privileges are not granted.



Requirement Delete Option in OIPA

- **RequirementDetails:** grants access to the RequirementDetails right-click option on the ActivityRequirementScreen. The Activity Requirement screen is accessed by clicking the requirement icon next to an activity. When the screen opens, right-click over the name of the requirement and the requirement detail option should display

unless security privileges are not granted.



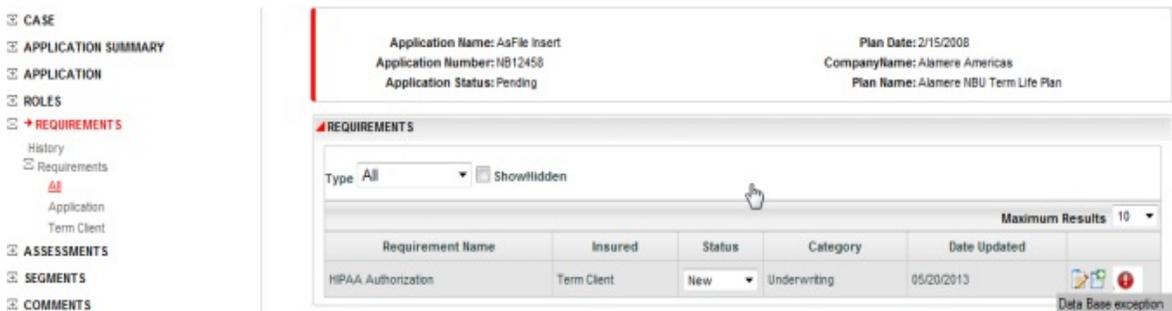
Requirements Detail Option in OIPA

- RequirementOk: grants access to the OK button on the ActivityRequirementScreen. If this is not checked for any transaction, then the user will not have the ability to manually update pending requirements.



Requirements OK Button in OIPA

- **PolicyRequirementScreen plan page:** allows granting access to the Policy Requirement screen page and the associated buttons for policy requirements.



Policy Requirement Screen in OIPA

- Requirement Error Search Screen company page: allows

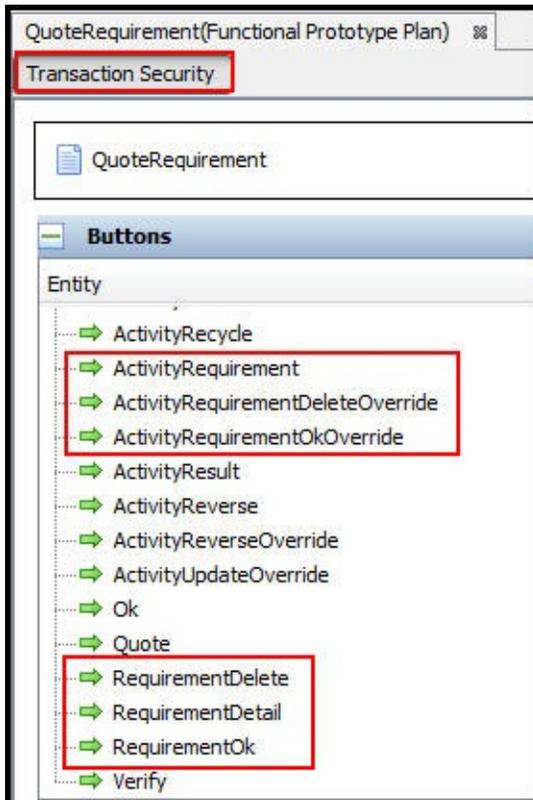
granting access to the Requirement Error Search Screen page under the Requirement main menu item, as well as granting access to the associated buttons for automatic policy requirements.



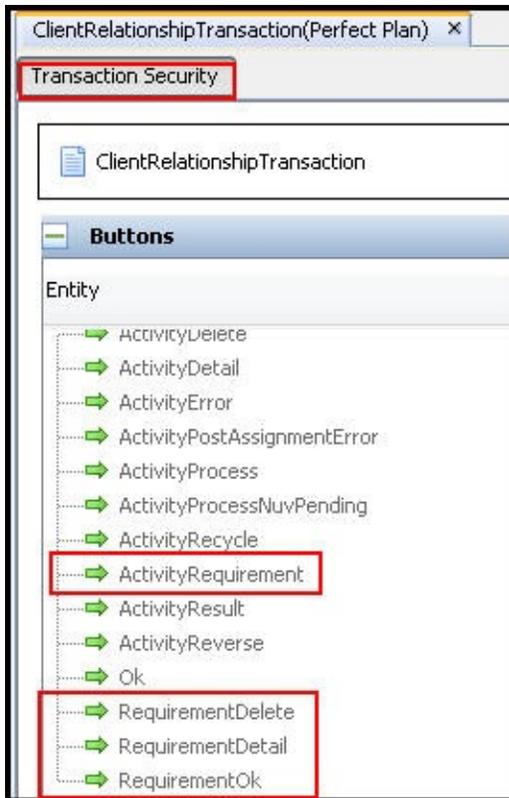
Requirement Error Search Screen in OIPA

Requirement Buttons

A screen shot of the Transaction Security Buttons is shown below. The nodes that control requirement security are outlined in red.



Requirement Security Buttons in Transaction Security



Requirement Transaction Security

You are here: Internationalization > Currency > Currency Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Currency Overview

The Rules Palette manages multiple currencies within the OIPA system. The currency information is displayed according to the international standards outlined in the ISO 4217 Currency Names and Codes Element Table. The Currency Editor is the tool used to define and update currencies and rounding methods, which are stored in the AsCurrency table. Currency codes cannot be edited.



If decimal displays need to be changed for the currency, use the [Localization Editor](#).

Once currency codes are defined in the Currency Editor, they can be associated with activities, plans or companies. This is done by [setting currency displays](#). [Currencies can also be associated with funds](#). Refer to the [Currency Conversion](#) section for more information on how currencies are used in the conversion process.

 Refer to the [Events pane](#) section for additional information on math and currency in Transaction Events.

Currency Rounding

The default method for managing currencies is to truncate the entry based on the value in the **Display Round Places** column of the AsCurrency record for the currency used. If rounding is preferred, the [CompanyCosmetics](#) business rule has an element that allows rounding to take precedence over truncating. This can be set for each company, if needed. OIPA will reference the AsCurrency record again, but instead of truncating will round the currency according to the currency round method and currency round places indicated for the currency in the AsCurrency record.

If the currency code of the field is changed after the value is entered, the value is compared to the new currency's DisplayRoundPlaces column in AsCurrency and adjusted accordingly.

Add Currency Codes

There may be times when additional currency codes are needed. The Currency Editor is used to add new currencies and [edit the existing ones](#). The information displayed in the currency editor is all stored in the AsCurrency table.

Once a new currency code is created, it can be selected as an option in the [Countries Editor](#) under the Taxable Currency Code drop-down box.



The screenshot shows a window titled "Currency" with an "Add" button and a table of currency configurations. The table has columns for Currency Code, Currency Name, Display Round Places, Currency Round Places, Currency Round Method, and Edit. The data rows are as follows:

Currency Code	Currency Name	Display Round Places	Currency Round Places	Currency Round Method	Edit
EUR	EURO	1	3	Round To Even	
GBP	GBP	7	7	Round Down	
INR	Indian Rupee	7	5	Round Down	
TWD	New Taiwan Dollar	2	2	Automatic Round	
USD	UNITED STATES DOLLAR	2	5	Automatic Round	
YEN	Japanese Yen	6	4	Round Down	

Currency File in Admin Explorer

Steps to Add a New Currency

1. Open the **Admin Explorer** window.
2. Double-click the **Currency** folder. It will open in the [Configuration Area](#).
3. Right-click on the currency XML file and select **Check-out**.
4. Click **Add** at the bottom of the Configuration Area window. If the **Add** button is not visible, grab the bottom of the window and pull it down until the Add button is visible.
5. Select a **Currency code** from the drop down menu. The corresponding Currency Name will display.
6. Select the number of decimal places to round the currency for display purposes.
7. Select the number of digits to the right of the decimal place to keep after a currency calculation is performed.
8. Select the rounding method to employ once a calculation is performed.
 - **Arithmetic Round**: Increases the last digit by one if the next digit is five or greater.
 - **Round Down**: Leaves the last digit the same, regardless of the next digit
 - **Round To Even**: Increases the last digit **by one** if the next digit is:
 - six or more
 - a five followed by one or more non-zero digits.
 - the next digit is a five followed by only zeros and the last digit is odd.**Round to Even** will leave the last digit **the same** if:
 - the next digit is four or less
 - the next digit is a five followed by only zeros and the last digit is odd.
9. Check-in the XML file to save the changes to the database.



The currency rounding is applied first based on the currency round places determined in step 7. Then, the result is truncated based on the display rounding places defined in step 6.

Steps to Update Existing Currency Attributes

1. Open the **Admin Explorer** window.
2. Double-click the **Currency** folder.
3. Right-click the currency XML file and select **Check-Out**. Each currency row contains five pieces of information.
 - **Currency Code**: Displays the currency code. Code names cannot be changed.
 - **Currency Name**: Displays the currency name. Display names can be updated.
 - **Display Round Places**: Determines the number of decimal places the currency will be rounded to for display purposes. This also indicates the number of places to keep before the value is truncated.
 - **Currency Round Places**: Indicates the number of digits to the right of the decimal place to keep after a currency calculation is performed. This column is referenced by the CompanyCosmetics business rule when the RoundCurrencyEntry element is set to **Yes**.
 - **Currency Round Method**: Determines the type of rounding method to employ once a calculation is performed. There are three options:
 - **Arithmetic Round**: Fractional remainder values of five (5) or greater are rounded up to the next whole digit. Four (4) or less are rounded down.
 - **Round Down**: All fractional remainders are rounded down.
 - **Round to Even**: Fractional remainders are rounded up or down depending on adjacent positional values. Also known as banker's rounding. Increases the last digit by one if the next digit is six or more, or a five followed by one or more non-zero digits. Also will increase the last digit by one if the next digit is five followed by zeros when the last digit is odd. Leaves the last visible digit the same if the next digit is four (4) or less.
4. Find the name of the currency for update in the Currency Name

column. The currency attributes that can be changed are in the columns to the right and are accompanied by a drop-down list.

5. Click the down arrow next to the attribute to change. The currency's row will appear highlighted in green.
6. Make the necessary currency rounding adjustments.
7. Check-in the file to save the changes to the database.



Currency Code	Currency Name	Display Round Places	Currency Round Places	Currency Round Method	Edit
EUR	EURO	1	3	Round To Even	<input type="checkbox"/>
GBP	GBP	7	7	Round Down	<input type="checkbox"/>
INR	Indian Rupee	6	6	Round Down	<input checked="" type="checkbox"/>
TWD	New Taiwan Dollar	2	2	Automatic Round	<input type="checkbox"/>
USD	UNITED STATES DOLLAR	2	5	Automatic Round	<input type="checkbox"/>
YEN	Japanese Yen	6	4	Round Down	<input type="checkbox"/>

Currency File in Admin Explorer

You are here: Internationalization > Currency > Set Currency Displays

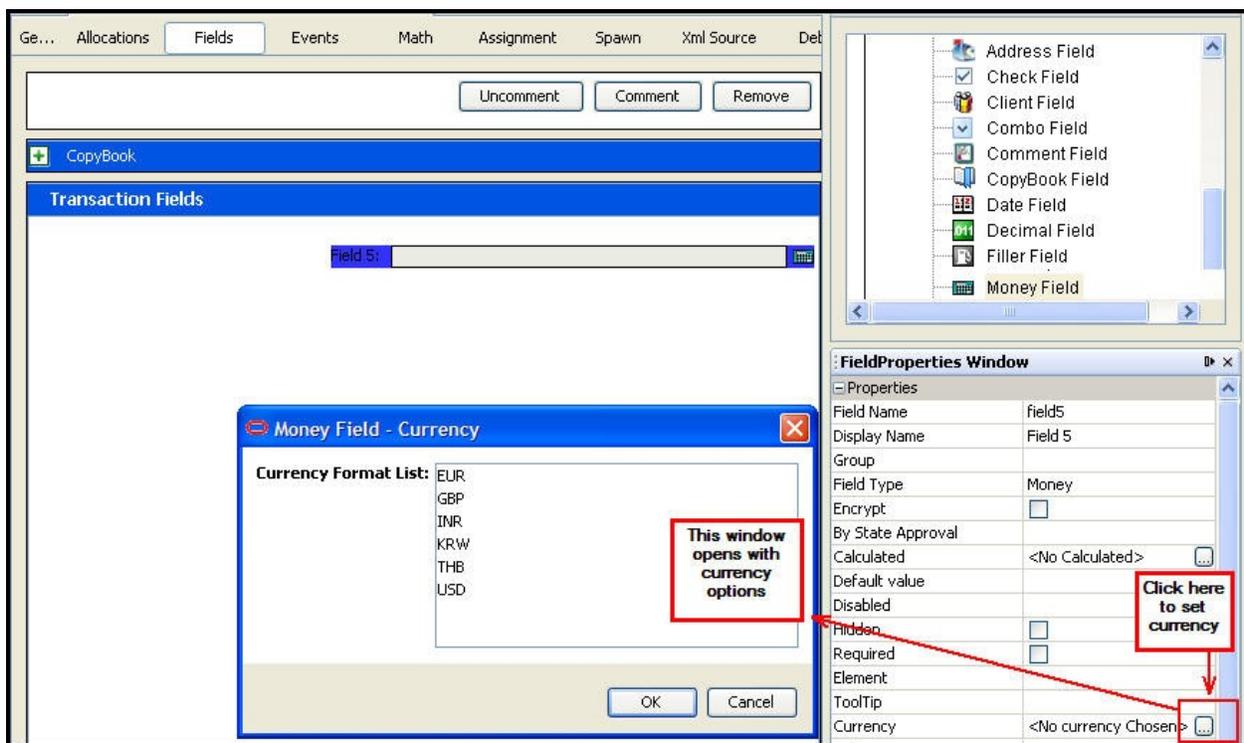
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Set Currency Displays

Currency can be set at various levels within the OIPA application. Currency can be defined at the activity level, the [plan level](#), the [company level](#) or the [system default](#) can be used if no currency is defined. When OIPA reads a money field, it will check at the lowest level (activity level). If no currency is specified, it will move up to the next level until a currency is set.

Activity Level Currency

[Configure a transaction](#) to always use a specific type of currency. Check-out the transaction and click the **Fields** pane. When the money field is clicked, the **Field Properties** window will open. Scroll through the properties to the Currency field. Select the currency from the drop down box. The currency set here will determine the currency that the OIPA user will see when processing this transaction.

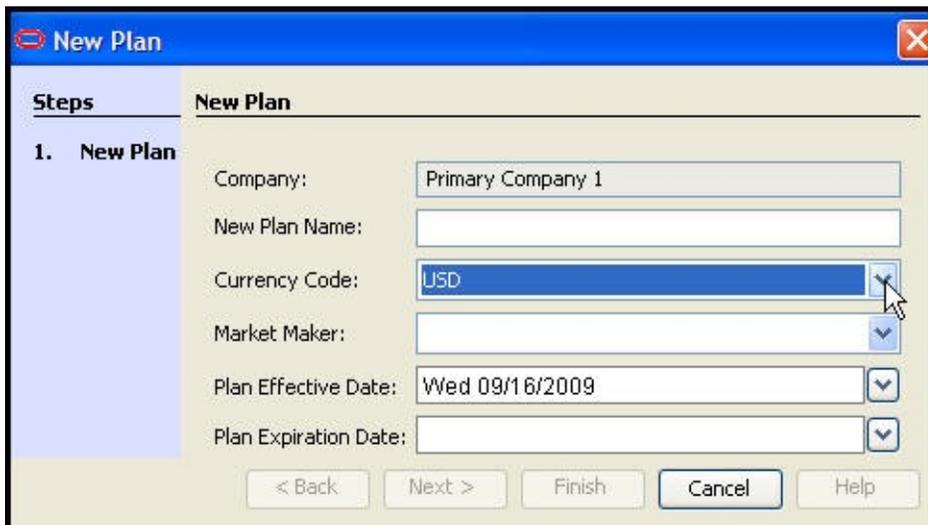


Currency Set at Activity Level

Plan Level Currency

When OIPA reads a money field it will check to see if a currency code has been assigned. OIPA first checks the activity level. If no currency is set, OIPA will check the plan level. Plan level currency codes can be set when a new plan is created. Currency codes assigned to a plan can also be updated.

When a new plan is created, select a currency from the currency code drop down box. This will associate the currency with all policies under that plan.



New Plan Wizard

From the Main Explorer, right-click on a plan and select **Edit Plan**. When the plan information opens in the Configuration Area, change the currency code and save the update.

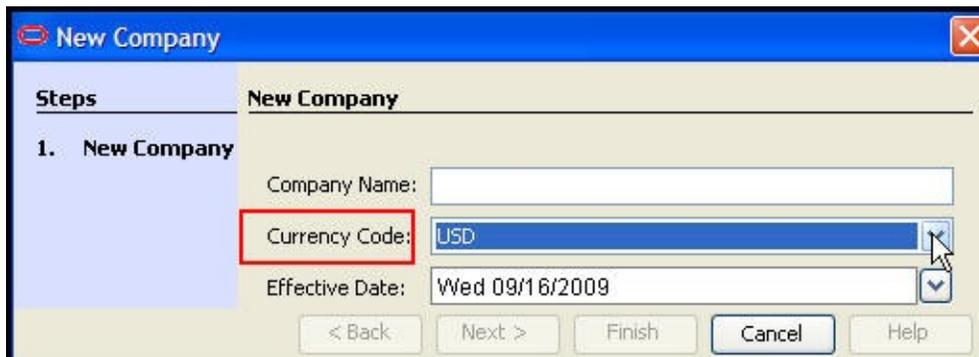


Plan Right-Click Options

Company Level Currency

When OIPA reads a money field it will check to see if a currency code has been assigned. OIPA first checks the activity and plan levels. If no currency code is set, it will check the company level. Company level currency codes can be set when a new company is created. Currency codes assigned to a company can also be updated.

When a new company is created, select a currency from the currency code drop down box. This will associate the currency with all plans and policies under that company.



The screenshot shows a 'New Company' wizard window. On the left, a 'Steps' pane shows '1. New Company'. The main area has a 'New Company' title and three input fields: 'Company Name' (empty), 'Currency Code' (set to 'USD' and highlighted with a red box), and 'Effective Date' (set to 'Wed 09/16/2009'). At the bottom are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

New Company Wizard

From the Main Explorer, right-click on a Company and select **Edit Company**. When the company information opens in the Configuration Area, change the currency code and save the update.



Company Right-Click Option

System Default Currency

Set the system default currency in the PAS.properties file. If the currency is not specified at any other level, the system default will be used.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Internationalization > Currency > Currency Codes for Funds

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Currency Codes for Funds

Funds are located in the Main Explorer tab under the associated plan folder. When a new fund is created, assign a default currency for the fund. The currency can be changed once a fund is created.

To change a default fund currency, open the fund from the Main Explorer. The Fund Details tab has a Currency Code drop down box where the currency can be changed.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Internationalization > Unit Linked Product > Currency Conversion

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Currency Conversion

The Oracle Insurance Policy Administration (OIPA) application can be configured to support multiple currencies and perform currency conversion using pair currencies and cross rates. Whenever a currency is converted, there is a cost incurred in converting the money, which is captured by the system when the conversion takes place.

The system can also be configured to calculate exchanges in either direct or indirect terms. This functionality provides the ability to support global equity trading processes and various aspects of internationalization so products can be developed that require foreign exchanges.

Understanding how the OIPA system processes currency conversion is important when approaching a strategy for configuration. All currencies use a three letter ISO-4217 currency code that is hard coded. [Default currencies that set a base currency for products](#) can be set at the activity, plan, company and system level, which override each other respectively. [Default currencies are also set-up at the fund level](#) and may vary from fund to fund depending on what currency it uses.

When the OIPA system performs assignment configuration in a transaction that is moving money in and out of funds, it checks to see if the default currency codes are different for the plan or company and the fund. If they are, then a currency conversion will be performed according to logic configured. Configure logic that instructs the system to perform the currency calculation in direct or indirect terms. Resulting values will be written in the AsValuation records along with any conversion costs. In buy transactions, the currency exchange is from plan or client currency to fund currency. In sell transactions, the currency exchange is from fund to plan or client currency.

Currency Conversion in Math

OIPA also provides the ability to configure currency conversion directly in rules that contain a math section. Currency conversion in math only supports the ability to convert from one currency to another as a simple formula, and there is no tracking of the currency conversion details. When currency conversion happens in the math, the system will look up the exchange rates for the currencies using rate as of a specified date, or the latest exchange rates that exist. The data is calculated and rounded properly, but no details of the conversion are captured.

Currency conversion in the math is an important part of the math processing and the conversion that takes place is based on foreign exchange rates. The foreign exchange rate can change daily and as a result the system needs to be able to specify the date used to determine the foreign exchange rate. The exchange rate will then be used by the math variable to perform the currency conversion.

The currency ExchangeDate element can be added to math variables with TYPE="CURRENCY" to specify the date used to locate the exchange rate used in conversion exchange calculation. The specified date should allow for past dates and the current date as long as the rate that corresponds to the configured date is present on the AsExchange rate table for the currencies involved.



The **XML Configuration Guide** provides details on conversions in rules that contain a math section. The [CURRENCY element](#) is described in the Transaction Rule | Transaction Elements | Math Elements | CURRENCY section.

High Level Steps Involved in Configuring Products that Require Currency Conversion

If currency conversion needs to be invoked in the system then the following steps must be taken to ensure correct Market Maker information, currency rounding methods, and proper terms (direct or indirect) for conversion calculations.



OIPA assumes all conversions use a single rate per business day and are loaded into OIPA. Unit linking uses the strike method of guaranteeing the next price after the activity date that requires a unit deal. OIPA stores various market calendars and recognizes days that prices can be guaranteed according to that specific market calendar.

1. Create currency codes using the [Currency](#) editor in the Admin Explorer tab.
2. Configure market maker information via [Market Maker](#) in the Admin Explorer tab.
3. Specify the Market Maker for the plan that is associated with the product. This can be done through the [New Plan Wizard](#) if a new plan is being created, or it can be done by [editing an existing plan](#).
4. Configure default currencies at the [activity, plan or company level](#) and the [fund levels](#).
5. Verify how the AsExchangeRate table will be updated. This may be done via a web service like [AsFile](#), direct insert into the database or another method. The following rules apply when loading exchange rates:
 - a. The Bid Price is always the amount that the Market Maker is willing to pay for the currency that needs to be exchanged.
 - b. The Offer Price is always the amount that the Market Maker will sell the currency for.
 - c. The Offer Price must always be greater than the Bid Price for direct exchanges.

- d. The Offer Price must always be less than the Bid Price for indirect exchanges.
6. [Configure transactions](#) that will perform currency conversions. These types of transactions usually move money in and out of funds and are associated with valuation. If your transaction has an assignment section, you must have a [MathStatement](#) function in the Math section to tell Assignment how to retrieve prices and exchange rates.



It is the assignment configuration that invokes OIPA to perform a currency conversion unless invoked directly with specific currency math configuration. The currency math configuration is explained in the XML Configuration Guide in the Transaction Rules>Transaction Elements>Math Elements>CURRENCY section.



Currency needs to be specified on all Money fields. You can list multiple currencies and allow selection during the data entry process.

You are here: Internationalization > Currency > Cross Rate Currency Calculation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Cross Rate Currency Calculation

When stocks are exchanged using different currencies and neither currency is the official currency of the country where the exchange quote is given, a cross rate can be used to facilitate the exchange.

In order to perform a cross rate currency calculation, the following information is required.

- **Originating Currency:** the three letter currency code the amount is currently in.
- **Amount:** a decimal that contains the number of units of originating currency to be converted.
- **Target Currency:** the three letter currency code the amount will be converted to.
- **Market Maker:** the market maker that is associated with the current company or plan.
- **Exchange Date** - allows the user to pass in a reference to a date formatted math variable or field (i.e. "Activity:EffectiveDate") that will be used to locate the exchange rate the system will use in the conversion exchange calculation.

Steps to Perform a Cross Rate Currency Calculation

1. Retrieve the CrossRateCurrencyCode for the current Market Maker from the AsMarketMaker table.
2. Retrieve an exchange rate with the CrossRateCurrencyCode as the BASE currency for both the Originating currency and Target currency:
 - a. Retrieve an exchange rate from the AsExchangeRate table where the TERMS currency is the Originating currency, and the BASE currency is the Market Maker's CrossRateCurrency.
 - b. Retrieve an exchange rate from the AsExchangeRate table where the TERMS currency is the Target currency, and the BASE currency is the Market Maker's CrossRateCurrency.
 - c. If a rate cannot be found for BOTH the Originating and Target currency, generate an Exception.
3. Retrieve the DirectTermIndicator for both the Originating currency and the Target currency. This will determine the cross rate calculation method.
4. Calculate the cross rate exchange rate using the two exchange rates retrieved from step two. The calculation performed is always as follows:

Result = Amount / Denom; where Amount is the amount of Originating currency to be exchanged, and Denom is the number of units of the Originating currency that can be exchanged for one unit of the Target currency.

5. Calculate the Denom in one of four ways, depending on the DirectTermIndicator of the cross rate exchanges:
 - a. Both currencies are direct (have a DirectTermIndicator of T):
 - i. Denom = Target Currency Offer Price / Originating Currency Bid Price
 - ii. Example of Cross Rate Currency Calculation:
AUD/EUR (Australian Dollar, Euro) with USD as the CrossRateCurrency, both in Direct terms

- Originating Currency = EUR
 - Target Currency = AUD
 - Amount = 100000 AUD
 - Exchange Date = 06/01/2010
 - USD/EUR Exchange = 1.2474 Bid; 1.2478 Offer
 - USD/AUD Exchange = 0.7296 Bid; 0.7299 Offer
 - Denom = $.7299 / 1.2474 = 0.585137$
 - Result = $100000 / 0.585137 = 170900.1$ AUD
- b. Both currencies are indirect (have a DirectTermIndicator of F):
- i. Denom = $\text{Originating Currency Bid Price} / \text{Target Currency Offer Price}$
 - ii. Example of Cross Rate Currency Calculation:
HKD/JPY (Hong Kong Dollar, yen) with USD as CrossRateCurrency, both in Indirect terms
 - Originating Currency = HKD
 - Target Currency = JPY
 - Amount = 100000 HKD
 - Exchange Date = 06/01/2010
 - USD/HKD Exchange = 7.2 Bid; 7.1 Offer
 - USD/JPY Exchange = 109.41 Bid; 109.40 Offer
 - Denom = $7.2 / 109.40 = 0.065814$
 - Result = $100000 / 0.065814 = 1519444$ JPY

Note: The Rounding Method here was to round up, and the Rounding Decimal Places was 0.
- c. Originating currency is quoted in direct terms; Target currency is quoted in indirect terms:
- i. Denom = $1 / (\text{Originating Currency Bid Price} * \text{Target Currency Offer Price})$
 - ii. Example of Cross Rate Currency Calculation:
AUD/HKD (Australian Dollar, Hong Kong Dollar) with USD as CrossRateCurrency, AUD in Direct Terms, HKD in Indirect terms

- Originating Currency = AUD
- Target Currency = HKD
- Amount = 100000 AUD
- Exchange Date = 06/01/2010
- USD/AUD Exchange = 0.7296 Bid; 0.7299 Offer
- USD/HKD Exchange = 7.2 Bid; 7.1 Offer
- Denom = $1 / (0.7296 * 7.1) = 0.193044$
- Result = $100000 / 0.193044 = 518016$ HKD

d. Originating currency is quoted in indirect terms, Target currency is quoted in direct terms:

i. Denom = Originating Currency Bid Price * Target Currency Offer Price

ii. Example of Cross Rate Currency Calculation:

JPY/EUR (yen - Euro) with USD as CrossRateCurrency, JPY in Indirect Terms, EUR in Direct terms

- Originating Currency = JPY
- Target Currency = EUR
- Amount = 100000 JPY
- Exchange Date = 06/01/2010
- USD/JPY Exchange = 109.41 Bid; 109.40 Offer
- USD/EUR Exchange = 1.2474 Bid; 1.2478 Offer
- Demo = $109.41 * 1.2478 = 136.5218$
- Result = $100000 / 136.5218 = 732.4838$ EUR

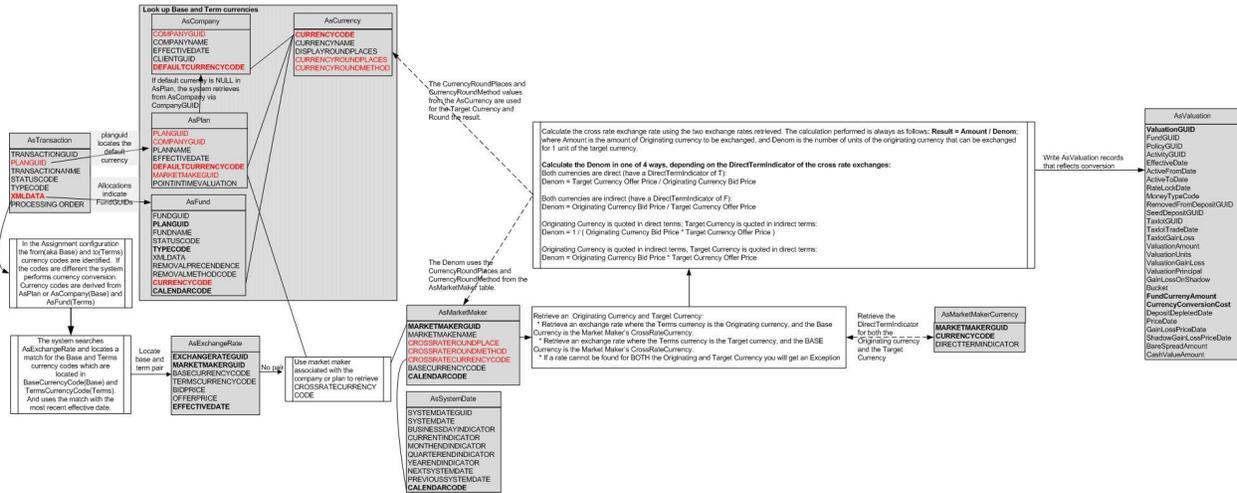
6. Round the Denom using the CurrencyRoundPlaces and CurrencyRoundMethod from the AsMarketMaker table.

Note: This is very important to the proper calculation of the exchange rate.

7. Divide the Denom INTO the amount.

8. Retrieve the CurrencyRoundPlaces and CurrencyRoundMethod values from the AsCurrency table for the Target currency and round the result.

OIPA Currency Conversion
Cross Rate



You are here: Internationalization > Currency > Direct Currency Calculation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Direct Currency Calculation

When stocks are exchanged using different currencies and one currency is the official currency of the country where the exchange quote is given, a direct exchange can be performed.

In order to perform a direct currency calculation, the following four pieces of information are required.

- **Originating Currency:** the three letter currency code the amount is currently in.
- **Amount:** a decimal that contains the number of units of Originating currency to be converted.
- **Target Currency:** the three letter currency code the amount will be converted to.
- **Market Maker:** the market maker that is being used for the current company or plan.
- **Exchange Date:** the date that will be used to locate the exchange rate the system will use in the conversion exchange calculation.

Steps to Perform a Direct Currency Calculation

1. Retrieve the exchange rate for the Originating currency and the Target currency supplied by the current Market Maker.
 - Search the AsExchangeRate table for a single instance that contains the Originating currency as either the BASE or TERMS currency, and the Target currency as either the BASE or TERMS currency, for the current Market Maker. If rows exist, get the row that has the most recent effective date or the date specified in the optional Exchange Date variable. If the Exchange Date is specified, an error is returned if no rate exists for the specified date.
 - If no rows exist in the AsExchangeRate table for Originating and Target currency, a cross rate calculation will need to be performed.
2. Retrieve the DirectTermIndicator for the current Market Maker and whatever currency is the TERMS currency from the Exchange Rate retrieved from the AsMarketMakerCurrency table.
 - If DirectTermIndicator is T, then direct, or American, terms are used in the quote. The quote is the number of BASE currency units needed for one unit of the TERMS currency.
 - If DirectTermsIndicator is F, then indirect, or European, terms are used in the quote. The quote is the number of TERMS currency units needed for one unit of the BASE currency.
3. Perform the currency conversion calculation to get the number of units in Target currency:
 - If the Target currency is the TERMS currency in the exchange rate, then the conversion is done by using the amount of Originating currency to BUY some number of BASE currency from the Market Maker. This is done using the following formula:
 - i. If Indirect: $\text{Amount} * \text{Offer Price} = \text{Number of units of Target Currency}$
 - ii. If Direct: $\text{Amount} / \text{Offer Price} = \text{Number of units of Target Currency}$
 - If the Target currency is the BASE currency in the exchange

rate, then the conversion is done by using the amount of Originating currency and SELLING it for some number of TERMS currency from the Market Maker. This is done using the following formula:

- i. If Indirect: $\text{Amount} / \text{Bid Price} = \text{Number of units of Target currency}$
- ii. If Direct: $\text{Amount} * \text{Bid Price} = \text{Number of units of Target currency}$

4. Retrieve the CurrencyRoundPlaces and CurrencyRoundMethod values from the AsCurrency table for the Target currency and round the result. Rounding is described in detail later.

Examples of Direct Currency Calculation

Exchange Rate Indirect Quote, USD/CAN

- BASE Currency = USD
- TERMS Currency = CAN
- OFFER = 1.0018
- BID = 1.0020

If converting 100000 USD to CAN then:

- Originating currency = USD (the BASE currency in this example)
- Target currency = CAN (the TERMS currency in this example)
- Amount = 100,000
- Currency Conversion = $100000 * 1.0018 = 100180.0000$ CAN

If converting 100000 CAN to USD then:

- Originating currency = CAN (the TERMS currency in this example)
- Target currency = USD (the BASE currency in this example)
- Amount = 100,000
- Currency Conversion = $100000 / 1.0020 = 99800.3992$ USD

Exchange Rate Direct Quote, USD/CAN

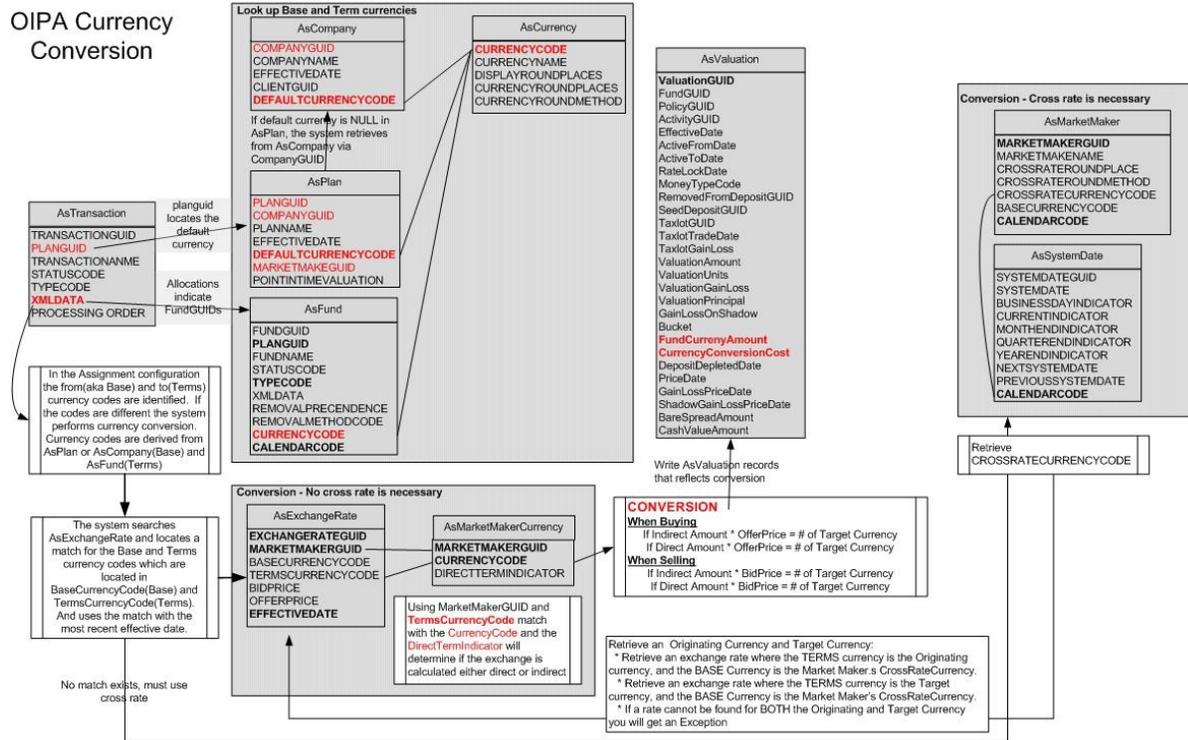
- BASE Currency = USD
- TERMS Currency = CAN
- OFFER = .9985
- BID = .9982

If converting 100000 USD to CAN then:

- Originating currency = USD (the BASE currency in this example)
- Target currency = CAN (the TERMS currency in this example)
- Amount = 100,000
- Currency Conversion = $100000 / .9985 = 100150.2254$ CAN

If converting 100000 CAN to USD then:

- Originating currency = CAN (the TERMS currency in this example)
- Target currency = USD (the BASE currency in this example)
- Amount = 100,000
- Currency Conversion = $100000 * .9982 = 99820$ USD



You are here: Admin Explorer > System Date

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

System Date

The System Date editor is used to enter each calendar day for a particular year and assign those dates specific system information. Each date is identified as a business day, the current day, month-end date, quarter-end date, year-end date or holiday. The system date information will be referenced during cycle processing to determine the day processing should occur. This information is stored in the AsSystemDate table.

The System Date administration tool supports [adding new calendars](#), [populating new calendars with dates using a date generation wizard](#), and manually [adding system dates to an existing calendar](#).



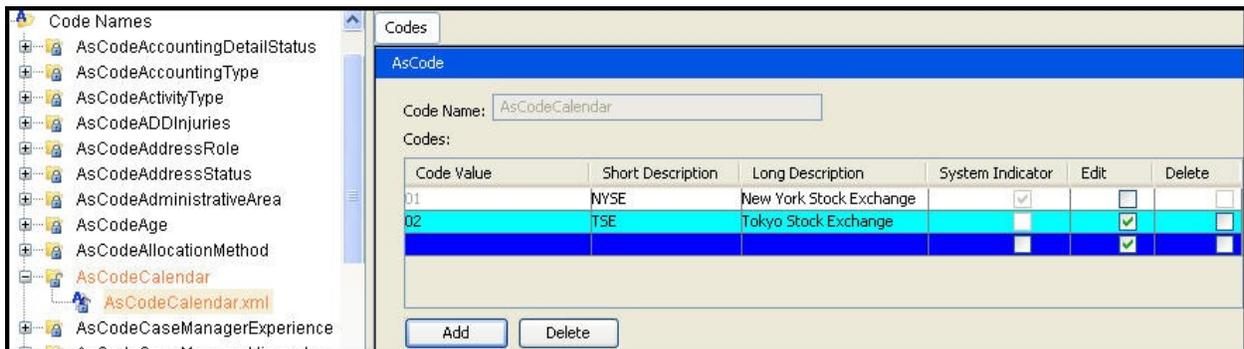
Once a system date is entered, it cannot be deleted.

New Calendars

New calendars are added through the [Code Names](#) section of the Admin Explorer. Once a new calendar is created, it is available in the calendar code field on the System Date.xml file and new system dates can be added.

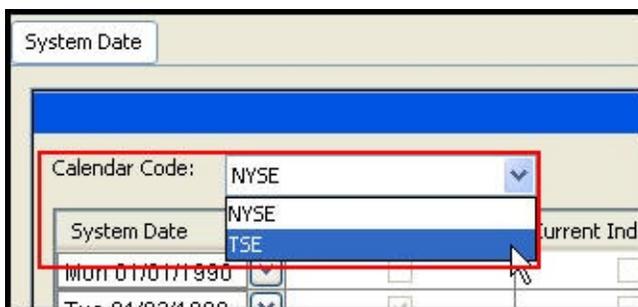
Steps to Add a New Calendar

1. Navigate to the **Admin Explorer**.
2. Open the **Code Names** folder.
3. Open the **AsCodeCalendar** folder.
4. Double-click on the **AsCodeCalendar.xml** file to open it in the Configuration Area.
5. Click **Add** to add a line for the new calendar.
6. Double-click on the Code Value field and enter the code value.
7. Double-click on the Short Description field and enter an abbreviation for your calendar.
8. Double-click on the Long Description field and enter the long name for your calendar.
9. Right-click on the AsCodeCalendar.xml file and select **Check-in**. This will save your changes to the database.



Calendar Codes in Admin Explorer

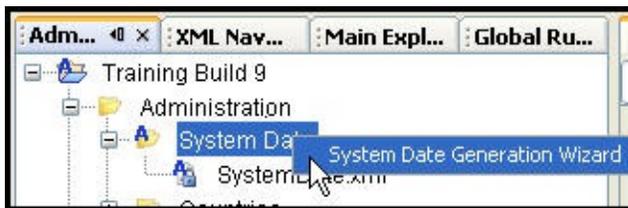
The new calendar will appear in the drop down list in the System Date file's Calendar Code field.



Calendar Codes on System Date File

Steps to Populate a New Calendar Using the Date Generation Wizard

1. Right-click the System Date node and select **System Date Generation Wizard**.



System Date Right-Click Options

2. Select the calendar where you want to add dates from the Calendar Code drop down box.
3. Enter the first date in your date range.
4. Enter the last date in your date range.
5. Click **Finish**.

The System Date Generation Wizard will add all the dates within the specified range to the new calendar. The wizard will also mark the appropriate indicator check box for each date.

Steps to Manually Add a New System Date

1. Select the Admin Explorer tab.
2. Open the Administration folder and then open the System date folder.
3. Right-click on the System date file and select **Check-out**.
4. Select the **Add** button at the bottom of the system date file. A new field will display.
5. Select a date from the System Date field.
6. Check the business day indicator if the day is a business day.



Do not check the **Current Indicator**. That is a system generated check box.

7. Check the month end indicator if the date is a month end date.
8. Check the quarter end indicator if the date is a quarter end date.
9. Check the year end indicator if the date is a year end date.
10. Select the next day's date from the Next System Date field.
11. Select the previous day's date from the Previous System Date field.
12. Check-in your System Date file to save the changes to the database.

Calendar Code: NYSE	Enter a date	Check here if it is a business day	System generated Don't check here	Check here if last day of month	Check here if last day of quarter	Check here if last day of year	Enter date of following day	Enter date of previous day
System Date	Business Day Indicator	Current Indicator	Month End Indicator	Quarter End Indicator	Year End Indicator	Next System Date	Previous System Date	Edit
Mon 01/01/1990	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tue 01/02/1990	Sun 12/31/1989	<input type="checkbox"/>
Tue 01/02/1990	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Wed 01/03/1990	Mon 01/01/1990	<input type="checkbox"/>
Wed 01/03/1990	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Thu 01/04/1990	Tue 01/02/1990	<input type="checkbox"/>
Thu 01/04/1990	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Fri 01/05/1990	Wed 01/03/1990	<input type="checkbox"/>
Fri 01/05/1990	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sat 01/06/1990	Thu 01/04/1990	<input type="checkbox"/>
Sat 01/06/1990	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sun 01/07/1990	Fri 01/05/1990	<input type="checkbox"/>
Sun 01/07/1990	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Mon 01/08/1990	Sat 01/06/1990	<input type="checkbox"/>
Mon 01/08/1990	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tue 01/09/1990	Sun 01/07/1990	<input type="checkbox"/>
Tue 01/09/1990	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Wed 01/10/1990	Mon 01/08/1990	<input type="checkbox"/>
Wed 01/10/1990	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Thu 01/11/1990	Tue 01/09/1990	<input type="checkbox"/>

Explanation of Calendar Code Fields

You are here: Admin Explorer > State Approval Editor

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

State Approval Editor

The State Approval editor is located in the Admin Explorer tab in the Administration folder. This editor allows state approvals to be captured at the plan and segment level according to effective and expiration dates.

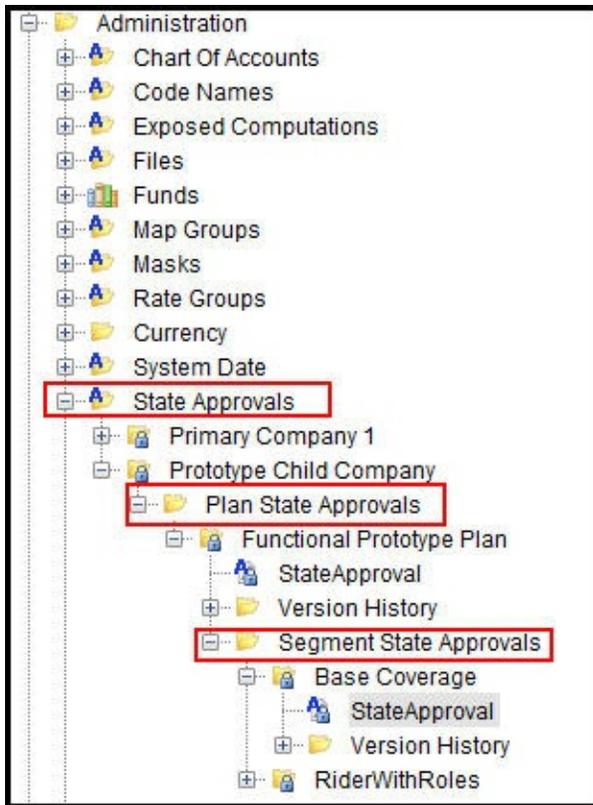
The editor is divided into two sections. The top portion cannot be edited and simply reflects the identifying information such as Company, Plan Group (the [Plan Group field](#) is only visible if the plan group functionality is turned on), Plan, Segment and version number. The bottom portion contains a list of states that can be selected to create an approval record. Each state can be assigned an effective date and an optional expiration date identifying when the approval is valid for the plan or segment. All plan information captured by this editor is saved to the **AsPlanStateApproval** table. All segment information captured by this editor is saved to the **AsSegmentStateApproval** table.

State Approval records can only be added to plans or segments that are set up to support this functionality. Refer to the [Plan Screen business rule General Pane section](#) for more information on setting up plans to support state approval. Refer to the [SegmentScreen business rule](#) for more information on setting up segments to support state approval.

Steps to Add Plan and Segment State Approval Records

1. Navigate to the **Admin Explorer**.
2. Open the **Administration** folder and scroll down to **State Approvals**.
3. Right-click on State Approvals and select **New State Approval**.
4. Select the company where the plan resides from the Company drop down box.
5. Select the plan from the Plan drop down box.
6. Select a segment from the Segment drop down box if a segment is available. Segments will appear in this box when they have been granted state approval privileges through the [Segment screen business rule](#).
7. Click **Finish** when the plan or segment has been selected. The name of the file will appear in the State Approval hierarchy in blue font.
8. Select effective dates for the states that are supported by the plan. Expiration dates are optional.
9. Check-in the new file to save the changes to the database.

New plans will appear under the Plan State Approvals folder, organized by company. New segments will appear in the Segment State Approvals folder, organized by company and plan.



State Approval Hierarchy in Admin Explorer

Delete Plan or Segment State Approvals

Once state approvals have been created, they can be deleted in two ways. The entire state approval for the plan or segment can be deleted by right-clicking on the plan level or segment level state approval editor and selecting **Delete Rule**. All individual state records for that plan or segment will be removed from AsPlanStateApproval or AsSegmentStateApproval.

Individual state approvals can also be deleted for a plan or segment by checking out the appropriate level editor. Select a state by clicking on the row, then click the **Delete** button at the bottom of the screen. A check will appear in the Delete column for that row. The approval will be removed when the editor is checked-in.



Delete Right-Click Option for Plan Level State Approvals

Update Plan or Segment State Approvals

1. Navigate to the **Admin Explorer**
2. Open the **Administration** folder and scroll down to **State Approvals**. Open the appropriate Company and Plan folders.
 - To edit plan state approvals, right-click on the **State Approvals** node under the Plan State Approval folder and select **Check-out**. This will open the plan level State Approval editor.
 - To edit segment state approvals, open the Segment State Approval folder, then open the Segment folder. Right-click on the State Approvals node and select Check-out. This will open the segment level State Approval editor.
3. Find the appropriate state row in the bottom portion of the editor. All states from AsCodeState occupy a row in the grid.
4. To add dates, double-click in the Effective Date column and type a valid date. The date must be in the format **m/d/yyyy**. A date picker tool is also available and can be used by double-clicking the down arrow next to the date field.
5. To remove dates, double-click in the date field, select the date and click the Delete button on the keyboard.
6. Right-click the State Approval file and select **Check-in**. This will save the updates to the database.

You are here: Internationalization > Countries

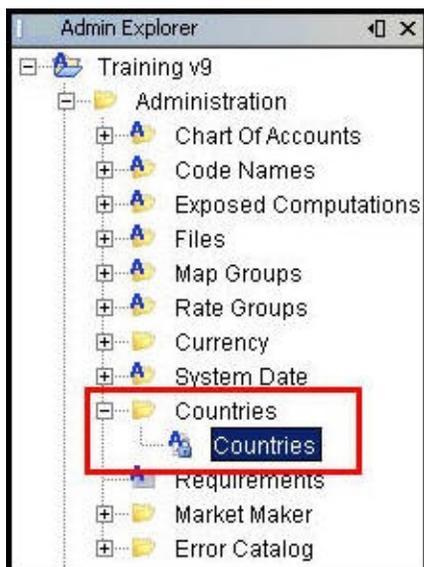
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Countries

The Countries editor is an administration feature used to add and edit country information stored in the AsCountry table. Add and edit the relationship between a supported country and its applicable Taxable Currency value.

If additional currency code options are needed in the Taxable Currency Code drop-down box, they can be added through the [Currency Editor](#). Once new currency codes have been added and saved through the Currency Editor, they appear as options in the Taxable Currency Code drop-down list.

The Country Editor can be accessed from the Admin Explorer tab in the Administration folder.



Country Folder in Admin Explorer

Steps to Add Country Information

1. From **Admin Explorer**, open the **Administration** folder.
2. Double-click the Countries folder.
3. Right-click the **Countries** XML file and select **Check-out**.
4. Click **Add Country**. A new row will appear highlighted in blue.
5. Select a Country Code from the drop down list. The Short Name and Long Name fields will be automatically populated when a Country Code is selected. The codes and descriptions are stored in the AsCountry table.
6. Click the field of the **Taxable Currency Code** and select the correct currency code from the drop-down box. Additional codes can be added through the Currency editor.
7. Check-in the file to save the changes to the database.

Country Code	Short Name	Long Name	Taxable Currency Code	Edit	Delete
GB	UK	United Kingdom	GBP	<input type="checkbox"/>	
US	USA	United States Of America	USD	<input type="checkbox"/>	
CA	Canada	Canada		<input type="checkbox"/>	
ES	Spain	Spain	EUR	<input type="checkbox"/>	
JP	Japan	Japan	JPY	<input type="checkbox"/>	

Add Country Version: 7

Country Editor in Admin Explorer

Edit or Delete Country Information

When the Country file is checked out, changes can be made to the information. Click in a field to change the Short or Long Name or select an option from the drop down box to choose a new country code or taxable currency code. The field will appear highlighted in green and the Edit box for that row will be checked to indicate that the information has changed. Check in the file to save the changes.



If a row exists in the database, then the Country code and Short name cannot be modified. Those fields will be disabled when the Country file is checked out. If a user needs to change the Country code or Short name, then he must delete the row first and then add a new row with the updated information.

If a row needs to be deleted, click the Delete checkbox. The row will appear highlighted in red. When the file is checked in, the row will be removed.

You are here: Admin Explorer > Market Maker

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

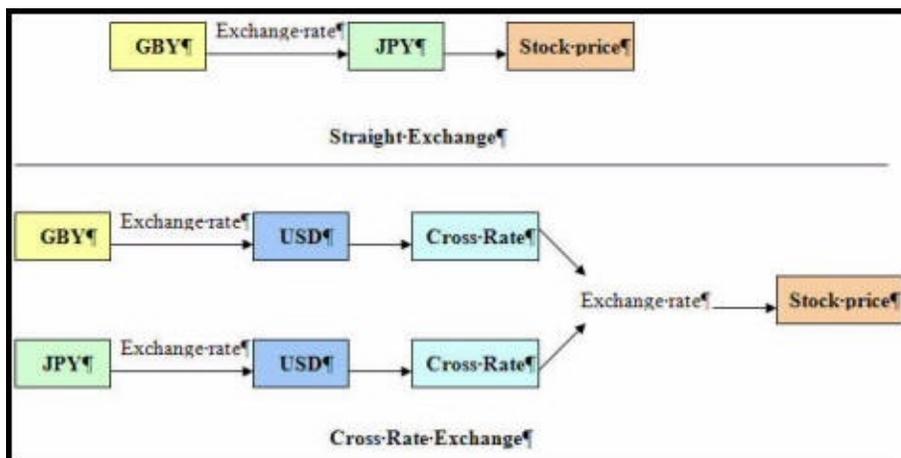
Market Maker

The Rules Palette uses Market Maker to store information about specific currency conversions that are associated with a plan. Market Maker is an administrative feature that can be found on the Admin Explorer tab. Two tables are used to store the currency conversion data. The first table is the [AsMarketMaker](#) table and the second is the [AsMarketMakerCurrency](#) table.

When stocks are exchanged using different currencies the Market Maker (also known as the Custodian Bank) provides the details of currency conversion. In some instances, it is not as simple as going from one currency to another. In such situations, you need an intermediary currency, or [cross rate](#), to facilitate the exchange. For example, suppose your currency pair (GBP/JPY) did not have an exchange rate. However, GBP/USD has an exchange rate and so does USD/JPY. The cross rate currency would be USD. Instead of one conversion, there would be two.



Refer to the [Cross Rate Currency Calculation](#) section for additional information on cross rates.



Explanation of Cross Rate Relationships

Direct vs. Indirect Terms

Once the cross rate currency is identified, there is one final step to consider. You must determine whether the currency rate will be quoted in direct or indirect terms. This is a critical step that will greatly impact the accuracy of the currency calculation.

Selecting direct versus indirect terms simply determines the currency that will be used as the denominator in the conversion equation. For example, if the conversion of USD/JPY was quoted in direct terms, then the domestic currency is the numerator and the foreign currency is the denominator. If the conversion was quoted in indirect terms, then it would be JPY/USD and the foreign currency would be the numerator and the domestic currency would be the denominator.

AsMarketMaker Table

The AsMarketMaker table holds six characteristics of the currency conversion.

1. **Name:** This is the common name for the Market Maker. Market Makers are broker-dealer firms that sell and buy stocks, making money on the spread and providing market liquidity.
2. **CrossRoundPlaces:** Number of places to round to the right of the decimal point. This rounding is applied to the denominator used in the cross rate calculation.
3. **CrossRateRoundMethod:** One of three possible rounding methods: 01-Arithmetic, 02-Round Down or 03-Round to Even. One of these methods will be applied to the denominator used in the cross rate calculation.
4. **CrossRateCurrencyCode:** This is the intermediate currency that must be used when there is no common exchange rate between two currencies.
5. **BaseCurrencyCode:** Default currency used.
6. **CalendarCode:** Identifies the working calendar for the Market Maker and determines if a Market Maker can trade a currency on a given date.

AsMarketMakerCurrency Table

The AsMarketMakerCurrency table holds additional information about the CrossRateCurrencyCode defined in the AsMarketMaker table. It determines whether each cross rate should be calculated in direct or indirect terms.

1. **CurrencyCode:** Identifies the currency code that will be associated with the value in the DirectTermIndicator field.
2. **DirectTermIndicator:** Indicates whether the currency code is quoted using direct or indirect terms. T, or true, indicates direct and F, or false, indicates indirect.

Steps to Add a New Market Maker

1. Navigate to **Admin Explorer**.
2. Expand the Administration folder.
3. Expand the Market Maker folder.
4. Right-click the MarketMaker.xml file and select **Check-out**. The Market Maker file will open in the Configuration Pane. It is divided into two sections. The top table contains the AsMarketMaker table information. The bottom table contains the AsMarketMakerCurrency table information.
5. Select the **Add Market** button at the bottom of the top table. A new row will appear highlighted in green.
6. Double-click in the Name column and enter the name of the new Market Maker. Each Market Maker name must be unique.
7. Double-click in the Cross Round Places column and select the number of places to the right of the decimal point that should be rounded.
8. Double-click in the Cross Rate Round Method column and select the rounding method to use.
9. Double-click in the Cross Rate Currency Code column and select the currency to use for the intermediate currency conversion. This cross rate currency is used when a currency pair does not have a direct exchange rate.



If the currency you want to select is not listed, you need to add the Currency code to the AsCodeCurrency table. The steps necessary to complete this action are listed in the [Currency](#) section.

10. Double-click in the Base Currency Code column and select the default currency the Market Maker uses. If the currency code isn't listed, refer to the Note above.
11. Double-click in the Calendar Code column and select the calendar code the Market Maker uses. The calendar code will determine if a trade can occur on a given date. If the calendar code is not listed,

then add it to the AsCodeCalendar table.

12. Right-click on the XML file and select **Check-in** to save the new Market Maker.

Once a new Market Maker is established, the Currency table should be updated. In this table a cross rate currency code is associated with either direct or indirect terms. A cross rate currency that is marked as direct will calculate the cross rate using the domestic currency as the numerator and the foreign currency as the denominator. A cross rate currency that is marked as indirect will calculate the cross rate using the foreign currency as the numerator and the domestic currency as the denominator.

Steps to Define Cross Rate Terms in the Currency Table

1. Navigate to **Admin Explorer**.
2. Expand the Administration folder.
3. Expand the Market Maker folder.
4. Right-click the MarketMaker.xml file and select **Check-out**. The Market Maker file will open in the Configuration Pane. It is divided into two sections. The top table contains the AsMarketMaker table information. The bottom table contains the AsMarketMakerCurrency table information.
5. Select the **Add Currency** button. A new row will appear highlighted in green.
6. Double-click in the Currency Code column and select the appropriate currency.
7. Double-click in the Direct Term Indicator column and select **T** for True or **F** for False. If **T** is selected, then the currency will be quoted using direct terms. If **F** is selected, then the currency will be quoted using indirect terms.
8. Right-click on the XML file and select **Check-in** to save the currency information.

MarketMaker

Market Maker

Name	Cross Round Places	Cross Rate Round ...	Cross Rate Curren...	Base Currency Code	Calendar Code	Edit	Delete
New MarketMaker	2	Round Down	INR	GBP	NYSE	<input type="checkbox"/>	<input type="checkbox"/>
New MarketMaker	3	Round Down	GBP	EUR	NYSE	<input type="checkbox"/>	<input type="checkbox"/>
New MarketMaker	2	Round Down	INR	USD	NYSE	<input type="checkbox"/>	<input type="checkbox"/>

AsMarketMaker Table

Add Market Remove Market

Currency Table:

Currency Code	Direct Term In...	Edit	Delete
		<input checked="" type="checkbox"/>	<input type="checkbox"/>

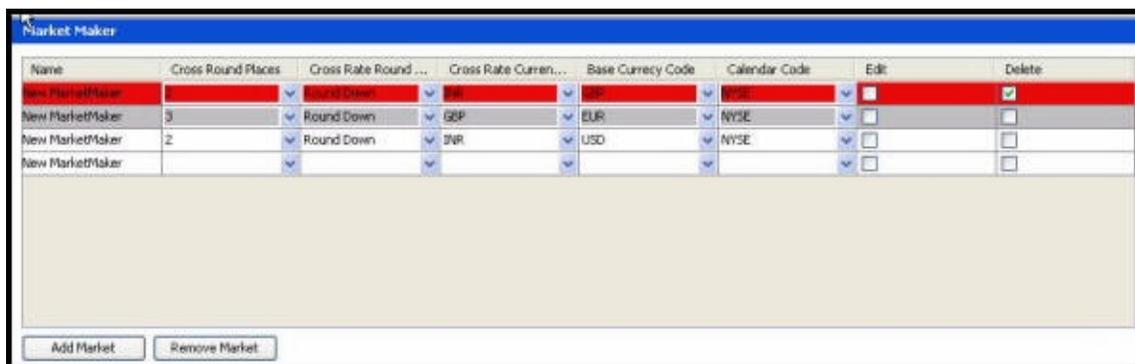
AsMarketMakerCurrency Table

Add Currency Remove Currency

Market Maker Editor in Rules Palette

Steps to Delete a Market Maker or Currency

1. Navigate to **Admin Explorer**.
2. Expand the Administration folder.
3. Expand the Market Maker folder.
4. Right-click the MarketMaker.xml file and select **Check-out**.
5. Highlight the row you want to delete.
6. Select the **Remove Market** button or the **Remove Currency** button.
The deleted row will be highlighted in red.
7. Right-click on the XML file and select **Check-in** to save the deletion.



Row Highlighted for Delete in Market Maker Editor

You are here: Admin Explorer > Masks Editor

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Mask Editor

The Mask editor is located in the Admin Explorer in the Administration folder. It is used to create the input and output values for the mask. A security level can also be assigned to the mask.

Steps to Create a Mask

1. Navigate to **Admin Explorer | Administration | Masks**.
2. Right-click on the Masks folder and click **New Mask Detail**. The Mask Editor will display.
3. Type a mask name.
4. Type an input value. The input must be a regular expression. The regular expression will validate the type of characters and the number of characters entered in OIPA. The regular expression also defines how the input characters are grouped. Refer to the [Mask Formats](#) section for additional information.
5. Type an output value. This tells OIPA how to display the input by the defined groups and supplied characters such as hyphens, parenthesis, etc. Refer to the [Mask Formats](#) section for additional information.
6. Select a security level for the mask. Security levels are defined in `AsCodeMaskSecurityLevel`. Use the [AsCode](#) editor to update these levels.
7. Click **Finish** and check-in the file to save the changes to the database.

Mask Editor



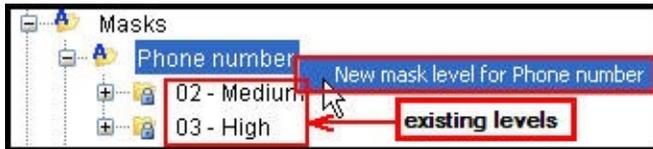
New Mask Detail Wizard



Once a mask has been created, its name cannot be edited.

Create Multiple Security Levels for a Mask

Once a mask is created, multiple levels of security can be assigned. For example, a phone number mask may have several security levels, according to the input/output value. To add additional security levels, right-click on the mask name and select **New mask level for Mask Name**. All mask levels will display below the Mask name in the Mask editor folder.



New Mask Level Right-Click Option

Delete a Mask or Mask Security Level

A mask security level may be deleted when a user opens the security level folder under the Mask name and right-clicks on the security level file and selects **Delete Mask Level**. If all mask levels are deleted, then the Mask name folder will be removed from the parent Mask folder.



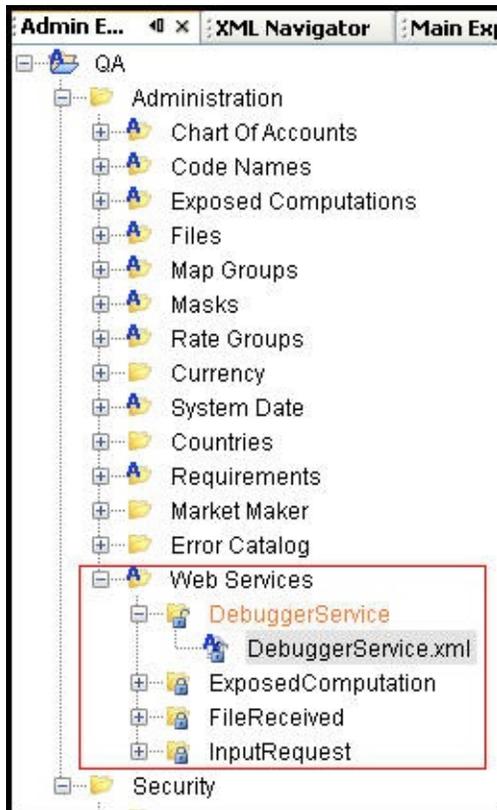
Delete Mask Right-Click Option

You are here: Admin Explorer > Web Services

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Web Services Editor

The Web Services editor allows a user to add Web Service records to the OIPA database. The editor can be accessed from **Admin Explorer | Administration | Web Services**.



Web Services Node in Admin Explorer

Add New Web Service

A new Web Service can be added by right-clicking on the main Web Services node in Admin Explorer. The only step involved in adding a new Web Service is to assign a name to the Web Service. Then the new Web Service will appear in the list of Web Services.

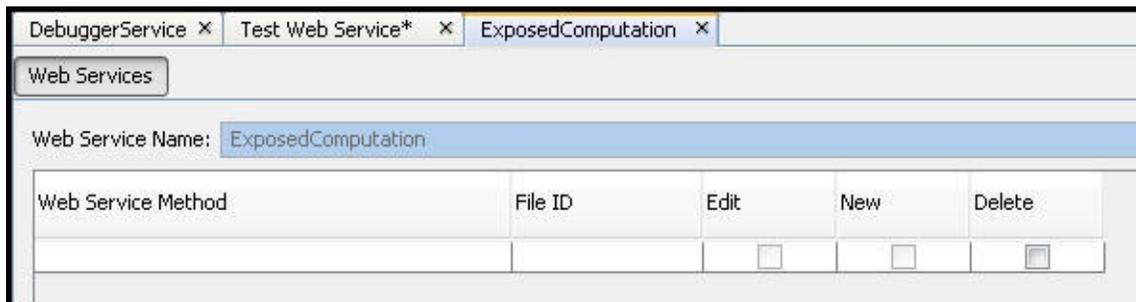
Delete Web Service

A Web Service can be deleted by right-clicking on the Web Service name in Admin Explorer and selecting the Delete option. When a Web Service is deleted all related records are removed from AsAuthCompanyWebService and AsAuthWebService.

Edit a Web Service

Right-click on a Web Service name and select **Check-out** to open the Web Service in the Configuration Area. An explanation of the fields that can be edited is given below. The Rules Palette will place a checkmark in the appropriate box in each row depending on the action the user attempts. If the Delete button is clicked, then the Rules Palette will populate the Delete checkbox with a check. If the Add button is clicked, then the Rules Palette will populate the New checkbox with a check. If existing data is modified, then the Rules Palette will populate the Edit checkbox with a check.

- Web Service Method:
- File ID: a combo box of File ID's from AsFile or Computation ID's from AsExposedComputation.



Web Service Options in Admin Explorer

You are here: Main Explorer > Main Explorer Tab

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Main Explorer Tab

The **Main Explorer** tab is one of two tabs that can be used to log into a Rules Palette environment. When logged in, the Main Explorer tab organizes information by company.

Products, plans, transactions and segments can be added and edited from this tab. A read-only view of all rules associated with a plan is available, but the rules cannot be edited. Rules are managed in the Global Explorer.



Business rules and plan rules cannot be overridden in this tab. The [Global Rules Explorer tab](#) must be used to override rules.

Main Explorer Navigation Tree

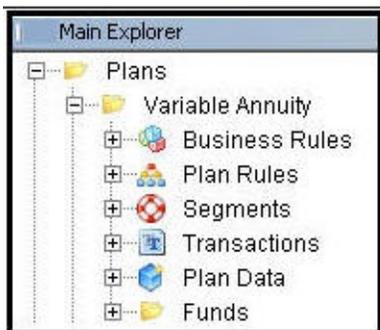
The Main Explorer is organized by company. Each company can have subsidiary company folders. Once a user drills down to the lowest company level (usually the subsidiary company), then specific company information can be accessed.

The navigation tree within the Main Explorer organizes information using the following categories:

- **Primary Company:** Contracting company representing the employer/plan sponsor offering benefits to eligible employees.
- **Subsidiary Company:** Multiple companies can exist under one primary company.
- **Product:** The Product is the line of business or highest level of coverage packaged to include all possible plan offerings, benefit packages, features, and ranges that is filed with the insurance commission. Similar Sub Products can be grouped under a Product to promote rules sharing and data inheritance. Examples of Products include Group Dental, Group Term Life, Group Disability.
- **Child Product:** Multiple Products can exist under one primary Product.
- **Business Rules:** Business rule can be overridden at the primary or Child Product level.
- **Segments:** Segments can be overridden at the primary or Child Product level.
- **Transactions:** Transactions can be overridden at the primary or Child Product level.
- **Plan:** set of Benefit Plan Options(s) for a Product/Sub-Product (e.g. Dental PPO) that is available to groups of employees under a Primary Company.
- **Company Data:** constant values for a company that are not based on any criteria.
- **Inquiry Screens:** Inquiry screens can pull various data from different tables in the OIGPA database that are relevant to the type of inquiry a business requires. Using inputs, outputs and various layout features, an Inquiry screen can be configured to display pertinent information in OIGPA.
- **Batch Screens**

Within the plan folders, information is broken down into six main categories. Each category is described below.

- **Business Rules**: Any rules that are attached to the plan, interface, screen, system, user defined, calculate and CopyBooks.
- **Plan Rules**: Any rules that are overridden at the plan level and are required for a plan to run in the application.
- **Segments**: All the segments attached to the plan.
- **Transactions**: All the transactions attached to the plan.
- **Plan Data**: Constant values for a plan that can be used for the configuration of calculations.



Plan Folders in Main Explorer

Right-Click Menus

Each main folder level in the Main Explorer has a right-click menu with available options that can be performed. Below is a list of the folder levels and the associated right-click options that will display. Any sub-folders are also listed along with the corresponding right-click options available to those folders.

- **Companies (Primary/Secondary) folder:** Right-click option is **Add New Company**

- Company Name: Right click option is **Edit Company** (Primary or Secondary) .



Company Right-Click Option

- **Product:** Right-click option is **Add Product**

- Product (Product or Child Product) Name: Right-click option is **Edit Product**.



- **Company Data folder:** No right-click options

- Company Data file has the following right-click options:
 - Check out: Allows user to check-out the Company Data file.
 - Check-in: Allows user to check-in the Company Data file.
 - Diff To: Allows user to compare one Company Data file to another file in the system.
 - Compare To: Allows user to generate a Detailed Diff Report for the selected Company Data file and another Company Data file.
 - Revert Modifications: Allows user to revert to a previous version of the file.



Company Data Right-Click Options

- **Plans** : Right-click option is **New Plan**.

- Plan Name: Edit Plan, Edit Plan Fields or Refresh
 - Edit Plan: Allows user to update Currency Code, Market Maker, Effective Date and Expiration Date for the plan.
 - Edit Plan Fields: Opens the Plan screen business rule override for that plan so that users can make updates to the fields displayed on the Plan screen.
 - Refresh: Refreshes all information for the plan so that any recently added or updated information from other CSRs will display in the Main Explorer.



Plan Right-Click Options

- **Business Rules**: No right-click menu available.
- **Segments**: Right-click options are **New Segment**, **Refresh** or **Compile**.
 - New Segment: Add a new segment to the plan.
 - Refresh: Refreshes all segment information for the plan so that any recently added or updated segment information from other CSRs will display in the Main Explorer.
 - Compile Segments: The segments will compile and any errors will display in the Engine Error Output window.

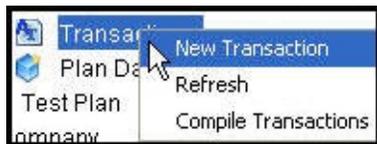


Segment Right-Click Options

- **Transactions**: Right-click options are **New Transaction**,

Refresh or Compile.

- New Transaction: Add a new transaction to the plan.
- Refresh: Refreshes all transaction information for the plan so that any recently added or updated transaction information from other CSRs will display in the Main Explorer.
- Compile Transaction: The transaction will compile and any errors will display in the Engine Error Output window.



Transaction Right-Click Options

- Plan Data: No right-click menu available.
- **Inquiry Screens:** Right-click option is **Add Inquiry**.
 - Check out: Allows user to check-out the Inquiry file.
 - Resolve: Displays all XML, including CopyBooks.
 - Revert Modifications: Allows user to revert to a previous version of the file.
- **Batch Screen:** Right-click option is **Add New Batch**.
 - Check-in: Allows user to check-in the Batch file.
 - Delete rule: Removes Batch.

You are here: Main Explorer > Prototypes > Prototype Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Prototypes

A Prototype Company is available in the Main Explorer. This company contains configuration samples, which demonstrate enhanced functionality specific to this release.

Use the Table of Contents in this help system to find the **New Prototype Samples** section. These prototypes demonstrate the new functionality added by this release. The **Previous Release Prototypes** section demonstrates functionality that was previously added to OIPA and is still available for review.

Each piece of functionality is described and a scenario is given explaining how it would be used in OIPA. Configuration requirements are listed and an explanation of the prototype sample is given.



Location of Prototype Samples in Main Explorer

You are here: Main Explorer > Prototypes > New Prototype Samples for this 10.1.2 Release > Ability to Link between Policy and Plan Segments

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Ability to Link between Policy and Plan Segments

This feature adds the ability to refer the plan segment data at policy segment level.

Prerequisites

- A group customer should be created, agreement, product, plan, segments, classgroup, class rule variables should be added. Employment relationship should be created with a client, activity EnrollmentPrototype should be added to the client and processed.
- User should be able to add the new transaction CreatePlanSegmentDetail to a policy. Security for the transaction should be given at both product and plan level.

New Items

New transaction CreateSegmentWithPlanSegmentDetail is configured.
CreateSegment BR is attached to this transaction.

Transaction name: CreateSegmentWithPlanSegmentDetail

Company name: Prototype Group Child Company

Product Name: Group Prototype Product

Configuration Details

- Existing EnrollmentPrototype transaction is configured to select the PlanSegmentNameGUID of the plansegment created for the customer, which the client is related to. CreatePolicy BR attached to the transaction is configured with PlanSegmentNameGUID attribute to create a policy segment linking it to the above plan segment.
- A new product level transaction is configured for Group Prototype Product, which has CreateSegment BR attached. The CreateSegment rule attached to it is configured with PlanSegmentNameGUID attribute. When added to the policy and processed, it takes the PlanSegmentNameGUID for the plan of the policy and links it to the segment created for the policy.
- Existing SegmentScreen BR for Group Prototype Product has been edited to include the group PlanSegmentName to display PlanSegmentName.
- Existing CalculateGeneralLifeBaseCov BR, product override for Group Prototype Product is configured with math variables of type field to access PlanSegmentName. These are then mapped to the respective database tables.
- Existing EligibleTransactionsByPolicyStatus, product override for Group Prototype Product, has been edited to include CreateSegmentWithPlanSegmentDetail transaction for pending policy status 08 and 09.

You are here: Main Explorer > Prototypes > New Prototype Samples for this 10.1.2 Release > Batch Processing Activities Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Batch Processing Activities Prototype

This feature helps the user to invoke a defined business process for all specified policies from a Batch screen and avoids the need to go to each individual policy.

Prerequisites

N/A

New Items

- New BatchScreen BR is configured at global level and prototype company override level.
- New BatchActivityDetailSearchScreen BR is configured at global level and prototype company override level.
- New Plan-Financial transactions BatchTransaction-InputModel and BatchTransaction-QueryModel are configured under System Plan for Prototype Company.
- New Policy level transactions PolicyScreenUpdate-InputModel and PolicyScreenUpdate-QueryModel are configured under Group Prototype Product.
- CopyToPolicyFields BR is attached to the policy level transactions.

Transaction name: BatchTransaction-InputModel

Company name: Prototype Company

Plan Name: System Plan

Transaction name: BatchTransaction-QueryModel

Company name: Prototype Company

Plan Name: System Plan

Transaction name: PolicyScreenUpdate-InputModel

Company name: Prototype Group Child Company

Product Name: Group Prototype Product

Transaction name: PolicyScreenUpdate-QueryModel

Company name: Prototype Group Child Company

Product Name: Group Prototype Product

Changes to Existing Items

Existing PolicyScreen BR, Group Prototype Product override has been updated with a new field UpdatedBy.

Configuration Details

- New BatchScreen BR and Prototype Company override are configured with an input model 'PolicyScreenUpdate-InputModel' batch name which is based on 'PolicyScreenUpdate-QueryModel' Query model.
- For batch PolicyScreenUpdate-InputModel, based on the tapecount entered by the user, a number of rows are displayed for user input. Action/Events have been configured to ensure mandatory fields have inputs. Total and Reconcile should be configured on Numeric columns. User is expected to enter valid Group Prototype Product policy numbers.
- For batch PolicyScreenUpdate-QueryModel, the number of rows returned by the query to fetch all Group Prototype Product policies are the number of rows displayed to the user. Again, the Action/Events have been configured.
- New BatchActivityDetailSearchScreen BR and Prototype Company override are configured for 'PolicyScreenUpdate-InputModel' and 'PolicyScreenUpdate-QueryModel' batches. These allow to search on certain fields and the results table returns the configured columns.
- New Plan-Financial transactions BatchTransaction-InputModel and BatchTransaction-QueryModel are configured under System Plan for Prototype Company which will be spawned when the respective batch is released and will spawn policy level transactions.
- New Policy level transactions PolicyScreenUpdate-InputModel and PolicyScreenUpdate-QueryModel are configured under Group Prototype Product.
- CopyToPolicyFields attached rule is attached to these policy level transactions.
- Existing PolicyScreen BR and Group Prototype Product override have been updated with a new field UpdatedBy.
- When a batch is released, transaction BatchTransaction-InputModel/BatchTransaction-QueryModel is spawned at Prototype Company level. This in turn spawns transaction PolicyScreenUpdate-InputModel/PolicyScreenUpdate-QueryModel respectively for each policy selected in the batch. The transaction copies text from the configured MV to the UpdatedBy policyfield of each policy using CopyToPolicyFields

attached BR.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > New Prototype Samples for this 10.1.2 Release > Client Search
Screen Security by Client Type

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Client Search Screen Security by Client Type

This feature provides an enhancement to the customers to limit few users from viewing the clients of specific types as needed.

Prerequisites

User should be able to see the ClientScreen.

Changes to Existing Items

Existing ClientScreen present at the 'Prototype' company is altered with the addition of new Client type 'Doctor'.

Configuration Details

- Existing ClientScreen present at the Prototype Company level is altered with the addition of client type '07' i.e.'Doctor'.
- Security for Client type 'Doctor' is not provided i.e. check box against the Client type 'doctor' is unchecked in the new collapsible pane present at the Client security page, which has all the client types.

Result: User should not be able to see the Client type 'Doctor' in Type combo of ClientScreen.

You are here: Main Explorer > Prototypes > New Prototype Samples for this 10.1.2 Release > Execute Valuation in Functions

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Execute Valuation in Functions

This feature adds an enhancement to the Math section of a user defined function that expands the capability of executing the functions.

Prerequisites

User should be able to add the "FunctionValuation" transaction in PAS application.

New Items

A new transaction called "FunctionValuation" is added at the "Functional Prototype Plan" level.

Transaction name: FunctionValuation

Company name: Prototype Individual Child Company

Product Name: Individual Prototype Product

Plan Name : Functional Prototype Plan

Function Name : Function-Valuation

Configuration Details

- The "FunctionValuation" transaction will have fields to capture valuation date and nearest price configured in it.
 - This transaction will give true or false to indicate whether the price for valuation used to value the policy were of the exact ValuationDates specified or the nearest price dates.
 - The transaction is configured at the 'Functional Prototype Plan' level and will have a new user defined function "Function-Valuation" called in it. This user defined transaction inturn calls the inbuild function MathValuation.
 - The output parameter returns true or false to indicate whether the fund prices used to value the policy as of the exact ValuationDates specified or the nearest price dates.
 - This new function will call the 'ActivityFunction' 'MathValuation' inside of it in the mathvariable section.

Result:User should be able to see the output of the 'ActivityFunction' which comes from "Function-MathValuation" in the mathsection after processing the activity.

You are here: Main Explorer > Prototypes > New Prototype Samples for this 10.1.2 Release > GetFundPosition Enhancements

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

GetFundPosition Enhancements

This feature expands the modification in input of the existing MathStatement activity function and returns value of a policy's funds based on the input criteria.

Prerequisites

User should be able to add the "EndResultFundTransfer " transaction in 'PAS' application.

New Items

A new transaction called "EndResultFundTransfer" is added at the "Functional Prototype Plan" level.

Transaction name: EndResultFundTransfer

Company name: Prototype Individual Child Company

Product Name: Individual Prototype Product

Plan Name: Standard Allocation Prototype

Attached Business Rules: ReassignAllocations

Changes to Existing Items

None

Configuration Details

- The transaction will have an activity function configured in it and the output parameter will represent the collection of fund positions with percent.
- The business rules attached to ReassignAllocations will have an allocation method as percent i.e. '01'

Result

User should be able to see the proper percentage of values allocated against funds in the output collection.

You are here: Main Explorer > Prototypes > New Prototype Samples for this 10.1.2 Release > MVAFunds
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

MVAFunds Prototype

The MVAAmountFormula provides a modified syntax to calculate the adjustment amount. The output of this function will be available in transaction math and the regular assignments will be available for MVA related funds.

Prerequisites

1. User should be able to add MVA fund at the policy level.
2. User should be able to add the 'MVADeposit', 'MVAWithdrawal' and 'MVATransfer' transactions in the application.

New Items

- Two new transactions called 'MVADeposit', 'MVAWithdrawal' and 'MVATransfer' transactions are created under 'Index Prototype Plan'.
- A new fund 'MVA Fund' is added at the 'Index Prototype Plan' level.
- A new business rule is created in the name of MVAAmountFormula which will calculate the MVAAmount depending on the MVAStartDate, MVAEndDate, and CashValue. MVA override is created for the new fund.
- A new override for InterestRateCalculation is added at the 'MVA fund' Level.

Transaction name: MVADeposit

Company name: Prototype Individual Child Company
Product Name: Individual Prototype Product
Plan Name: Index Prototype Plan

Transaction name: MVAWithdrawal

Company name: Prototype Individual Child Company
Product Name: Individual Prototype Product
Plan Name: Index Prototype Plan

Transaction name: MVATransfer

Company name: Prototype Individual Child Company
Product Name: Individual Prototype Product
Plan Name: Index Prototype Plan

Changes to Existing Items

CopyBook-AllValuesIndex is altered to include MVA Fund name into it.

Configuration Details

- **MVAAmountFormula** : The configuration is done as such to calculate the MVAAmount with the help of MVAStartDate, MVAEndDate, CashValue of MVA fund using the formula provided.
- **MVADeposit**: This configuration will have an input field which captures the deposit amount to the funds. With the help of assignment, the captured amount will be invested into MVA Fund.
- **MVAWithdrawal**: This configuration will have an input field which captures the withdrawal amount from the funds. The WithdrawalAmount captured from the transaction is added with the MVAAmountFormula output. The summed up amount will be deducted from the fund with the help of RemoveByFund assignment.
- **MVATransfer**: This transaction will remove the amount from the MVA Fund and invests the money to other allocations which is present at the Policy Level on the basis of fund count.

You are here: Main Explorer > Prototypes > New Prototype Samples for this 10.1.2 Release > Premium Tracking

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Premium Tracking

OIPA provides the ability to track, apply, and remove the premiums based on the age of the premium.

Changes to Existing Items

- Existing Initial Premium transaction under VDA plan is updated with <AssignFinancialEntry TYPE="Apply"> configuration.
- Existing Bonus transaction under VDA plan is updated with <AssignFinancialEntry TYPE="Apply"> configuration.
- Existing PartialWithdrawalWithRedemption transaction under VDA plan is updated with <AssignFinancialEntry TYPE="Withdrawal"> configuration, FinancialEntry:FinancialEntryGUIDMV:Amount is configured to fetch total amount left at Financial Entry level for the GUID referenced in the MathVariable FinancialEntryGUIDMV.
- Existing FullWithdrawalWithRedemption transaction under VDA plan is updated with <AssignFinancialEntry TYPE="Withdrawal"> configuration.
- AsCodeFinancialEntry - A new Code Value 03, Short and Long Description as 'Bonus' has been added.

Configuration Details

- Existing Initial Premium transaction under VDA plan is updated with <ValueFinancialEntry>, <AssignFinancialEntry TYPE="Apply"> configuration. User is allowed to select a FinancialEntryTypeCode and when this activity is processed, a record is inserted in AsFinancialEntry table for Gross Amount with the MoneyType as '01' and TypeCode as the one selected by the user.
- Existing Bonus transaction under VDA plan is updated with <ValueFinancialEntry> and <AssignFinancialEntry TYPE="Apply"> configuration. When the activity is processed the BonusAmount record is created in AsFinancialEntry table with the MoneyType as '67' and TypeCode as '03' - a new typecode created for Bonus in AsCodeFinancialEntry.
- Existing PartialWithdrawalWithRedemption transaction under VDA plan is updated with <ValueFinancialEntry>, <AssignFinancialEntry TYPE="Withdrawal"> configuration. Context Variable FinancialEntry:FinancialEntryGUIDMV:Amount is configured . User is allowed to select a FinancialEntryTypeCode and when this activity is processed, a record is inserted in AsFinancialEntry table for Withdrawal Amount with the MoneyType as '32' and TypeCode as the one selected by the user. The loop configured for FinancialEntry records the total balance for financial entry guides in a summing up variable TotalBalanceMV using the context variable FinancialEntry:Balance.
- Existing FullWithdrawalWithRedemption transaction under VDA plan is updated with <ValueFinancialEntry>, <AssignFinancialEntry TYPE="Withdrawal"> configuration. When the activity is processed the CashValue record is created in AsFinancialEntry table with the MoneyType as '32'.
- Existing AsCodeFinancialEntry has been updated with a new Code Value 03, Short and Long Description as 'Bonus' to be used as a TypeCode attribute value for <AssignFinancialEntry> tag in Bonus transaction.

You are here: Main Explorer > Prototypes > New Prototype Samples for this 10.1.2 Release > Transfer Indexed Funds (Money) from one Bucket to another Bucket

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transfer Indexed Funds from one Bucket to another Bucket

This feature is used to transfer the Index Funds amount (money) from one bucket to another bucket.

Prerequisites

- Investments for Index Funds should exist on a policy.
- PAS user should have access to FundsTransfer transaction of Index Prototype Plan.

New Items

Transaction name: FundsTransfer

Company name: Prototype Individual Child Company

Product Name: Individual Prototype Product

Plan Name : Index Prototype Plan

Attached BR : ReassignAllocations

Changes to Existing Items

New transaction FundsTransfer is added to EligibleTransactionByPolicyStatus BR (for active policy status) override of Index Prototype Plan.

Configuration Details

The below list of fields are configured for the transaction:

- Transfer Type

SOURCE DETAILS

- Select Fund
- Select Bucket Number

TARGET DETAILS

- Number of Funds
- Select First Fund
- Transfer Percentage For First Fund
- Select Second Fund
- Transfer Percentage For Second Fund
- Bucket Number
- Bucket Effective Date

With this transaction user is allowed to perform 3 different types of transfers between Index Funds:

Case 1: Choose the Transfer Type field value as 'Money transfer with source and target buckets as same' and then key in the required information.

Results: After processing an activity the cash value of source fund will be transferred to the target funds based on the Percentages chosen and source bucket.

Case 2: Choose the Transfer Type field value as 'Money transfer with source and target buckets as different' and then key in the required information.

Results: After processing an activity the cash value of source fund will be transferred to the target funds based on the Percentages chosen and target bucket.

Case 3: Choose the Transfer Type field value as 'Money Transfer on a specific bucket effective date' and then key in the required information.

Results: After processing an activity the cash value of source fund will be transferred to the target funds based on the Percentages chosen, target bucket and bucket effective date.

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Ability to create and maintain relationships using attached business rules

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Ability to create and maintain relationships using attached business rules

The “time slice” is a way of keying in changes to the details of an entity in OIPA through the UI instead of keying in such changes through activities. As part of OIPA-5619 (Implement History/Time Values support for Client Relationship (retrofit of Class Group design), the client relationship functionality is enhanced to extend time slice way of edits to relationship data. This feature is to enhance data intake functionality to allow recording and editing relationships with the time slice information appropriately. The feature will now plug the gap and allow Data Intake to create/update client relationships with appropriate time slice data.

Prototype Configuration Detail

Overview

This feature is to ensure a transaction can help to create a relationship with a new client or with an existing client/Group customer and also to create a new relationship between two already created clients or edit the relationship with a new time slice or editing an existing time slice.

Prerequisites

User should be able to add the 'CreateOrMaintainRelationship' transaction in application.

A new transaction 'CreateOrMaintainRelationship' is created at 'Prototype' Company Level.

Transaction name: CreateOrMaintainRelationship

Company name: Prototype Company

Plan Name :Client Plan

Business rules attached:CreateClient BR
MaintainRelationships BR
ValidateExpression BR

Transaction name: CreateOrMaintainRelationship
Company name: Prototype Company
Plan Name :Customer Plan
Business rules attached: CreateClient BR
MaintainRelationships BR
ValidateExpression BR

Changes to Existing Items

None

Results:

1.When the user adds the transaction at 'ClientLevel':

If the user selects Relationship as 'Primary' from "Relationship As" field:

'Select Client' field will be visible for selection or Creation of Client.

The 'Individual' and 'Spouse' relationship is considered.

If the user manually does not enter ClientDetails then
'CreateClient' BR will be invoked and Relationship will be added.

If the user manually enters the ClientDetails then
'MaintainRelationship' BR will be invoked and Relationship will be created with the entered client.

If the user selects Relationship as 'Secondary' from "Relationship As" field:

'Select Customer' combo field will be visible for selection of GroupCustomer .

The 'Employment' and 'FullTime' relationship is considered. .

The user has to select the Customer manually then 'MaintainRelationship' BR will be invoked and Relationship will be added

2. When the user adds the transaction at 'CustomerLevel':

Here the user will be able to add only Primary Relationship and also "Relationship As" field is disabled:

'Select Client' field will be visible for selection or Creation of Client.

The 'Employment' and 'FullTime' relationship is considered .

If the user manually does not enter ClientDetails then 'CreateClient' BR will be invoked and Relationship will be added.

If the user manually enters the ClientDetails then 'MaintainRelationship' BR will be invoked and Relationship will be created with the entered client.

3. In all the cases the Relationship added through this transaction will be in "Active" RecordStatus.

4.If the user tries to add the same Relationship record Overridible Exception "Same Relationship cannot be created again" will be thrown with the help of 'Validate Expression' BR in order to avoid the duplicate relationship.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

ABILITY TO DISPLAY DIFFERENT PRIMARY RELATIONSHIP ON CLIENT AND CUSTOMER

This enhancement is to provide the ability to display different Primary Relationship on client and customer

Prototype Configuration Detail

Overview

This new feature will introduce Only one Transaction type for all screen update activities.

Prerequisites

User should be able to add "ClientRelationship" "ClassGroup" timeslice in the 'PAS'.

Changes to Existing Items

Proper Security is provided for the "Activate" button in ClassGroup,ClientRelationship Company Pages.

ClientRelationshipScreen

Existing timeslice screen update transaction is updated to the typecode '40'.

Below mentioned changes for "ClientRelationshipScreen" is done at Global and all its override levels.

The typecode of the below mentioned transactions will be changed to '40'

which is referred in 'ClientRelationshipScreen' BR

ClientRelationshipUpdate.

ClientRelationshipActivate.

Process Immediate Attribute will be added to spawn section in "ClientRelationshipScreen" BR will be set to "Yes".

ClassGroupScreen

The typecode of the below mentioned transaction will be maintained as it was '40' which is referred in 'ClassGroupScreen' BR

ClassGroupUpdate.

Process Immediate Attribute will be added to spawn section in "ClassGroupScreen" BR and will be set to "Yes".

The screen update transactions will not be seen in GroupCustomer "Activity" screen.

Result :

1. When the user adds the timeslice to the screen in which "ProcessImmediate" tag is set to "Yes".

Automatically the timeslice will be active after processing from "Draft" status.

2. When the user adds the timeslice to the screen in which "ProcessImmediate" tag is set to "No".

The user should be able to see thunderbolt symbol next to timeslice.

User should be able to process manually and make timeslice as "Active".

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Ability to reverse the Clients and Policies created via Create Clients and Create Policy BR while Activity reversal

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Ability to reverse the Clients and Policies created via Create Clients and Create Policy BR while Activity reversal

This enhancement is to enable a way to reverse the client records or policy records by creating a way to either soft delete (put to Shadowed status) such records

Reversing of Activity with Create Clients or Create Policy attached Business Rules

Activity Reversal Processing

On reversal of an activity that creates a Client or a Policy through the CreateClients BR or CreatePolicy BR, A "business error" is thrown in case reversal is attempted in an invalid scenario:

In case of a user initiated reverse/recycle action, the condition will be evaluated before the reversal activity is added or activity details is displayed for recycle processing. In case the condition in the expression fails, the message will be displayed in a popup window. The recycle/reverse action will be stopped and the user cannot enter the reversal/recycle. See screenshot below for the popup screen display.

In case of a system initiated undo/redo action, the undo activity (and redo activity as well in case of redo) will be created automatically. In this case, when the undo activity is processed, the first step will be to evaluate the expression and if it fails, a business error will be displayed similar to all business errors (like ValidateExpressions error messages). The business error will be added on the undo activity (and not the forward activity which is already in Active status). This error will not be overrideable.

On CreateClients Business Rule and CreatePolicy Business Rule, the following shall be the expression evaluated to determine whether reversal processing is applicable or not:

CreateClients Business Rule: When reversing an activity that processes CreateClients Business Rule to create one or more client records, the following conditions will be evaluated on each client created by the CreateClients Business Rule. If any of the following conditions fail on any of the client records created, a business error stating "There are Roles or Activities that need to be manually reversed/deleted. Recycle/Reversal cannot be processed."

Validation for Activities that have some processing completed.

There should be no activities (all types of activities to be considered for this check - both program and non-program; both spawned and user-added; both Normal and Recycle activities) in a status where there is some processing (lightning bolt action) beyond just addition of activity has been already performed: TypeCode = '01' or '04'; StatusCode not in '02', '09', '12' and '34'.

If there is one or more activities with some processing (statuses defined in point i above), system

will error out – above business error will be displayed.

Validation for Roles in “Active” status that the Client is the Role Player for:

There should be no Role records in AsRole table (relating to Policy roles only; Company roles needn't be counted) or AsAgreementRole (relating to Agreement level roles) or with this Client as the Role Player that is in “Active” status.

When considering Policy Roles, only Policies which are not in "12" - Shadow status should be considered. This is required since the UI based Shadow Policy functionality allows Role records stay unchanged basis configuration.

If there is one or more Policy or Agreement roles in “Active” status – the system should display above business error.

CreatePolicy Business Rule: When reversing an activity that processes CreatePolicy BR to create a policy record, the following conditions should be evaluated on the policy created. If any of the following conditions fail, a business error stating "The policy created from this activity contains some Activity records that need to be manually reversed. Recycle/Reversal cannot be processed."

Validation for Activities that have some processing completed.

There should be no activities (all types of activities to be considered for this check - both program and non-program; both spawned and user-added; both Normal and Recycle activities) in a status where there is some processing (lightning bolt action) beyond just addition of activity has been already performed: TypeCode = '01' or '04' StatusCode not in '02', '09', '12' and '34'.

If there is one or more activities with some processing (statuses defined in point i above), system will error out – above business error will be displayed.



ReversalProcessing section in CreateClients Business Rule

The element <ReversalProcessing> in the CreateClients BR will define the way the Client record(s) created from the CreateClients BR will be handled when the transaction is reversed/recycled (both user initiated and undo/redo). The syntax will be as defined in the xml help guide

There will be two child elements <ShadowOnReversal> and <DeleteOnReversal>. Both will allow configuring a set of Expressions to be evaluated under the <Tests> element.

The expressions configured under the ShadowOnReversal and DeleteOnReversal elements will continue to be evaluated during reversal processing.

The expressions configured under these two elements can use all Activity Fields and any Activity Math

variable that was logged (into AsActivityMath table) - non-logged math variables cannot be used..

The <ReversalProcessing> is an optional element and if this is not configured, when the activity with CreateClients BR is reversed, the Client records created will be unaffected

When configured, the system will execute the reversal based on expression evaluation under the ShadowOnReversal and DeleteOnReversal elements as follows:

If all expressions configured under the <Tests> element under <ShadowOnReversal> are TRUE:

The Client record will be Shadowed

Change StatusCode of client in AsClient table to "Shadowed" status.

Update the UpdatedGMT on the AsClient table with the current timestamp.

Any activity in Pending statuses (status code '02' and '09'); move them to "Pending Shadow".

Update all related tables that involve a Status column to the status value corresponding to Shadow/Inactive/Expired.

If all expressions configured under the <Tests> element under the <DeleteOnReversal> are TRUE:

The Client record will be Deleted

The Client record from the AsClient table.

All activities linked to the client from the AsActivity table including any activities in status code '02' and '09'.

Delete related records in all related tables

This will be executed only if the ShadowOnReversal is FALSE (not TRUE). If Both <ShadowOnReversal> and <DeleteOnReversal> evaluate to true, the ShadowOnReversal takes precedence.

If both <ShadowOnReversal> and <DeleteOnReversal> evaluate to FALSE, then the Client record will stay unaffected. This is similar to current CreateClients BR where the client records are not affected.

ReversalProcessing section in CreatePolicy Business Rules

The element <ReversalProcessing> will define the way the Policy record created from the CreatePolicy BR will be handled when the transaction is reversed/recycled (both user initiated and undo/redo). The syntax will be as defined in the xml help guide.

There will be two child elements <ShadowOnReversal> and <DeleteOnReversal>. Both will allow configuring a set of Expressions to be evaluated under the <Tests> element.

The expressions configured under the ShadowOnReversal and DeleteOnReversal elements will continue to be evaluated during reversal.

The expressions configured under these two elements can use all Activity Fields and any Activity Math variable that was logged (into AsActivityMath table) - non-logged math variables cannot be used.

The <ReversalProcessing> is an optional element and if this is not configured, when the activity with CreatePolicy BR is reversed, the Policy record created will be unaffected.

When configured, the system will execute the reversal based on expression evaluation under the ShadowOnReversal and DeleteOnReversal elements as follows:

If all expressions configured under the <Tests> element under <ShadowOnReversal> are TRUE:

The Policy record will be Shadowed (soft-deleted).

Change StatusCode of policy in AsPolicy table to "Shadowed" status.

Update the UpdatedGMT on the AsPolicy table with the current timestamp.

Any activity in Pending status; move them to "Pending Shadow".

Update all related tables that involve a Status column to the status value corresponding to Shadow/Inactive/Expired.

If all expressions configured under the <Tests> element under the <DeleteOnReversal> are TRUE:

The Policy record will be Deleted.

The Policy record from the AsPolicy table.

All activities linked to the client from the AsActivity table table including any activities in status code '02' and '09'.

Delete related records in all related tables

This will be executed only if the ShadowOnReversal is FALSE (not TRUE). If Both <ShadowOnReversal> and <DeleteOnReversal> evaluate to true, the ShadowOnReversal takes precedence.

If both <ShadowOnReversal> and <DeleteOnReversal> evaluate to FALSE, then the Policy record will stay unaffected.

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Ability to view both Primary Secondary Relationships on the client screen

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Ability to view both Primary Secondary Relationships on the client screen

In OIPA, currently the Client and Customer Relationship Screens have the capability to display relationship records and details for relationships where the current Client/Group Customer is identified as the primary. In relationship records where the current Client/Group Customer is the secondary client, the user will need to navigate through the primary Client/Group Customer to view the relationship record and the associated details. This enhancement is to enable display of such relationship records where the current Client/Group Customer is identified as secondary client also in the Client or Customer Relationship screen in the context of the current client

Relationship screen - Upstream and Downstream view:

On Click of the Relationship node on the left side Menu ,two expandable nodes will be listed as a tree structure under the Relationship node.

By default on click of the Relationship Node,Downstream Relationship is selected and the first entity under the Downstream Relationship, the data for that relation will be displayed on the right side of the screen as show in the below screen shot.

The behavior of the Downstream relationship screen will be same as the existing Relationship screen ,with the below two changes

A new node(Downstream Relationships) is newly added to access this data.

The Display format of the 'Name' column in the grid results has the same format specified in the ClientScreen Business Rule for Clients and

GroupCustomerScreen for Group Customers.

The labels of the Column's will be renamed from 'Secondary Relationships' to 'Relationship Sub Type'

As mentioned above, this screen will behave exactly same the current relationship screen (based on the security provided there will be ability to Add relationships, Add Time Slices for existing relationships etc.).

Six new Filters will be present in the filter section of this screen

Relationship Sub Type - This will be a drop down box with all the Secondary Relationships configured (in Client Relationship Screen BR) for the Primary relationship category selected on the Left side node.

By default this drop down does not have a value selected.

On selection of a value from the drop down and On Click of 'Refresh' button, the grid results would refresh with the results based on the Filter Criteria

First Name - This will be a free form user enterable text field to search the secondary relationships based on the First Name column.

By default this textbox is empty.

On entering any string and On Click of 'Refresh' the grid results would refresh with the results based on the Filter Criteria.

The usage of % will be supported in this field and would work exactly as other search screens

Last Name - This will be a free form user enterable text field to search the secondary relationships based on the Last Name column.

By default this textbox is empty.

On entering any string and On Click of 'Refresh' the grid results would refresh with the results based on the Filter Criteria.

The usage of % will be supported in this field and would work exactly as other search screens

Company Name - This will be a free form user enterable text field to search the secondary relationships based on the Company Name column.

By default this textbox is empty.

On entering any string and On Click of 'Refresh' the grid results would refresh with the results based on the Filter Criteria.

The usage of % will be supported in this field and would work exactly as other search screens

As of Date -This will be a Date field and will be used to filter the relationship based on this Date.

By default this is filled with System Date.

On entering any new date and On Click of 'Refresh' the grid results would refresh with the results based on the Filter Criteria.

Display option -This will have two check boxes 'Future' and 'Past'.This will be used to display the future and past relationships based on the As of date field.

By default these two checkboxes are unchecked.

On checking any of these fields and On Click of 'Refresh' the grid results would refresh with the results based on the Filter Criteria.

Note: When <UseBusinessStatus> is set to 'Yes' in the ClientRelationshipScreen,then there would be an additional Status Filter in this section.This would follow the current behaviour.

On Click of the relationship records,all the TimeSlice or History records are displayed in a Grid.The first Label on the header section will be "Relationship Sub Type/Name".

Screenshot showing the Default behaviour on Click of relationship node(Group Customer Context).



Screenshot showing the Default behaviour on Click of relationship node(Client Context).



When the user clicks on the 'Upstream Relationship' link

No Additional nodes are provided under this node

All the Active Upstream relationships(Relationships in which the current Client or Group Customer's ClientGUID is in the SECONDARYCLIENTGUID column of ASCLIENTRELATIONSHIP table) will be displayed on the right side of the screen

The screen will have three columns by default and is not configurable currently(similar to the Downstream Relationship Screen)

Relationship Type

This column will display the 'PRIMARYRELATIONSHIPTYPE' from the ASCLIENTRELATIONSHIP table.

Relationship Sub Type

This column will display the 'SECONDARYRELATIONSHIPTYPE' from the ASCLIENTRELATIONSHIP table.

Name

This column will display the name of the client who is the Primary client for this relation('PRIMARYCLIENTGUID' in the ASCLIENTRELATIONSHIP Table)

The Display format of the 'Name' column in the grid results should follow format specified in the ClientScreen Business Rule for Clients and GroupCustomerScreen for Group Customers.
Business Rule

On click of any of the relationships displayed in this table, the relationship details are displayed (as it shows the downstream Relationship screen) but in read only mode

All the ACTIVE time slices are displayed (as it currently shows in the Relationship screen) but in read only mode

Add,Save and Cancel buttons are not available on this Screen

There will be no ability to add any new timeslices or edit any timeslices from this screen

On Click of Shadow checkbox and click on refresh,all the shadowed TimeSlice records will be displayed.

The right Click option will have the option to 'Go to Client' or 'Go to Customer' based on the client type of the primary client

Seven Filters will be present in the filter section of this screen('As of Date' and Display option of 'Future' and 'Past' will also be available and will function similar to how the function in the downstream Relationship screen today).

Relationship Type - This will be a drop down box with all the Primary Relationships that this client/group customer has with another Client.(PRIMARYRELATIONSHIPTYPE from ASCLIENTRELATIONSHIP table where the SecondaryClientGUID is the current ClientGUID)

By default this drop down will not have a value selected.

On selection of a value from the drop down and On Click of 'Refresh' button,the grid results would refresh with the results based on the Filter Criteria

Relationship Sub Type - This will be a drop down box with all the Secondary Relationships that this client/group customer has with another Client.(SECONDARYRELATIONSHIPTYPE from ASCLIENTRELATIONSHIP table where the SecondaryClientGUID is the current ClientGUID)

By default this drop down does not have a value selected.

When a selection is made on the 'Relationship Type' filter, then the list is narrowed down to the Secondary relationship's for the Primary relationship selected (all the Secondary Relationships that this client/group customer has with another Client in conjunction with the Primary Relationship selected)

To elaborate with an example, let's take the last screenshot attached. In that screen, the RelationshipType drop down will have values of 'Employment' and 'Contacts' and Relationship SubType will have values 'FullTime' and 'HRManager'.

Once the user selects 'Employment' in the RelationshipType drop down, then the RelationshipSubType drop down list will have only 'FullTime'

On selection of a value from the drop down and On Click of 'Refresh' button, the grid results would refresh with the results based on the Filter Criteria

First Name - This will be a free form user enterable text field to search the Primary relationships based on the First Name column.

By default this textbox is empty.

On entering any string and On Click of 'Refresh' the grid results would refresh with the results based on the Filter Criteria.

The usage of % will be supported in this field and would work exactly as other search screens

Last Name - This will be a free form user enterable text field to search the Primary relationships based on the Last Name column.

By default this textbox is empty.

On entering any string and On Click of 'Refresh' the grid results would refresh with the results based on the Filter Criteria.

The usage of % will be supported in this field and would work exactly as other search screens

Company Name -This will be a free form user enterable text field to search the Primary relationships based on the Company Name column.

By default this textbox will be empty.

On entering any string and On Click of 'Refresh' the grid results would refresh with the results based on the Filter Criteria.

The usage of % will be supported in this field and would work exactly as other search screens in OIPA

On Click of the relationship records,all the TimeSlice or History records are displayed in a Grid.

The first Label on the header section of this Grid Results needs to be updated to "Relationship Sub Type/Name".Currently it is "Secondary relationship/Name"

The Data shown under this column(Relationship Sub Type/Name) of the grid result should reflect the "Relationship Sub Type/Name of the PrimaryClientGUID in the AsRelationship table"

Screenshot showing Downstream relationships for a specific GroupCustomer (under Employment Primary relationship type)



Screenshot showing Downstream relationships for a specific GroupCustomer (under Contacts Primary relationship type)



Screenshot showing Upstream Relationship records for a specific GroupCustomer



Screenshot showing Upstream Relationship records for a specific Client (The Contacts and the Employment Secondary relationships show in the above



Screenshots for the group customer can be seen under the Upstream Relationship of that specific client in the below screenshot.)



Note:

The name of the Links 'Downstream Relationships' and 'Upstream

Relationships' is translatable.

The labels against the Filters is translatable.

The name of the column headers in the grid result section is also translatable.

The RelationshipType and Relationship SubType filter values is ordered Alphabetically (Ascending) in both the screens

The Label 'Secondary Relationship' on the PopUp window when a user clicks on 'New' under the DownStream Relationship will be updated to 'Relationship SubType' and also it is translatable.

There will be no Status Filter on the Upstream relationship Screen even though <UseBusinessStatus> is set to 'Yes' in the ClientRelationshipScreen Business rule. But the Status Column will be visible in the result section when <UseBusinessStatus> is set to 'Yes'

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Accessing suspense information in actions configuration

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Accessing suspense information in actions configuration

The suspense information entered/selected on the activity details screen of a transaction is now available in screen math, actions, and transaction math. Transactions may now evaluate the suspense information, perform validations, and execute calculations with the details pertaining to the suspense information for the transaction. The Suspense tab of an activity functions as an extension of the transaction fields and changes to the 'ActivitySuspenseNumber' can be used to trigger actions or the field can be populated by an action triggered by an different event.

Prototype Configuration Detail

Overview

This feature will allow the system to now be able to detect load and change events of the Suspense tab of an activity as well as update the suspense number on that tab through actions

Prerequisites

User should be able to add the "ActivityArray" transaction in PAS application. After processing the "ActivityArray" transaction

User should also be able to see the "ActivitySuspenseNumber" Transaction

New Items

A new transaction 'ActivitySuspenseNumber' is created at 'Dynamic Prototype Plan' Level.

Transaction name: ActivitySuspenseNumber

Company name: Prototype Individual Child Company

Product Name: Individual Prototype Product

Plan Name :Dynamic Prototype Plan

Changes to Existing Items

None

Configuration Details

ActivitySuspenseNumber

It is a Plan level transaction which has been configured at Dynamic Prototype Plan.

Fields configured in this transaction are outlined below:

Amount: User enterable Money field

SuspenseNumber: Which fetches the Suspense Number selected in the Suspense tab.

We have have accessed the 'ActivitySuspenseNumber' in the Math variable to showcase the accessing of ActivitySuspenseNumber functionality.

In Application when the transaction is added to the Dynamic Prototype Plan.

An ONLOAD action to display a message if the suspense number is not present, "Select Suspense detail before submitting of this activity".

An ONCHANGE action which will assign the SuspenseNumber field in the activity with the selected Suspense Number in the Suspense tab of the activity.

An ONSUBMIT action to display a message if the suspense number is not present, "Suspense Number is required" and checks to ensure Available amount of the

Suspense attached is greater than the Amount entered in the activity field.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Activity Array
Population of Entities For Activity Processing

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Activity Array Population of Entities For Activity Processing

Currently when a transaction is spawned using the ActivityArray option, we can only pass data to Activity Fields but there is no option to pass on data relating to Allocations, MultiFields, Suspense, MultiSuspense, which could be required in transaction processing. This enhancement is to address this gap and provision for passing data into such auxiliary transaction data.

There are two other related items that are included in this enhancement

ActivityArray now provides the option to "APPEND" or "SETVALUE" in the Post Assignment Math to an existing ActivityArray created in the Activity Math.

When using GenerateSuspense BR, the Suspense Number generated from the GenerateSuspense BR is now available for access in the Post Assignment Math.

Prototype Configuration Detail

Overview

This feature gives enhancement to the capability of Activity Array to support Multifields, Allocations and Suspense.

Prerequisites

User should be able to add the "ActivityArray" transaction in PAS application. After processing the "ActivityArray" transaction User should also be able to see the "SuspensePAVE" Transaction

New Items

A new transaction 'ActivityArray' and 'SuspensePAVE' is created at 'Dynamic Plan' Level.

Transaction name: ActivityArray

Company name: Prototype Individual Child Company
Product Name: Individual Prototype Product
Plan Name :Dynamic Prototype Plan

Transaction name: SuspensePAVE

Company name: Prototype Individual Child Company
Product Name: Individual Prototype Product
Plan Name :Dynamic Prototype Plan

BusinessRules Attached :

PostAssignmentValidateExpressions
Generate Suspense
MultiField-PrototypePolicyScreen

Changes to Existing Items

None

Configuration Details

"Entity Level For Linking Allocation" field will have 'Plan','Policy','Activity' for user selection.

MultiFields details are passed via MathVariable.

SuspenseDetails should be entered into "SuspenseNumber1" and "SuspenseNumber2" fields along with the field "Amount" by the user.

Allocations details will be passed depending on the selection in the "Entity Level For Linking Allocation" field.

If the user adds and process the "AddActivity" transaction "SuspenseInPAVE" transaction will be autoprocressed since PROCESSIMMEDIATE="Yes" in 'ActivitySequenceProcess' tag in the "AddActivity" transaction.

Scenario 1:

If the user selection is Plan:

Along with the suspense and multifeilds values PlanLevel allocationdetails will be sent to "SuspenseInPAVE" transaction.

Result:

User should be able to see PlanLevel Allocation,Suspense Detail and Multifeild details in ActivitySummary screen of "SuspenseInPAVE" transaction

Scenario 2:

If the user selection is Policy:

Along with the suspense and multifeilds values PolicyLevel allocationdetails will be sent to "SuspenseInPAVE" transaction.

Result:

User should be able to see PolicyLevel Allocation,Suspense Detail and Multifeild details in ActivitySummary screen of "SuspenseInPAVE" transaction.

Scenario 3:

If the user selection is Activity:

Along with the suspense and multifields values user should add ActivityLevel allocationdetails also which will be sent to "SuspenseInPAVE" transaction.

Result:

User should be able to see ActivityLevel Allocation,Suspense Detail and Multifield details in ActivitySummary screen of "SuspenseInPAVE" transaction.

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Batch Suspense

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Batch Suspense

This feature is similar to earlier suspense batch screen. The feature has the ability to create multiple suspense items from a 'master' suspense record.. Each subsidiary suspense record consumes part of the master suspense record amount by attaching itself and linking to the master record by a configurable batch number (AsSequence). The existing suspense screen rule has added the element 'SplitSuspense' which indicates if the entered suspense record may be subdivided among subsidiary suspense records within a batch. This is an advantage over prior designs where separate screen rules were needed for suspense and batch suspense.

The SuspenseSearchScreen rule will allow for the searching of suspense records by context fields and batch number. Context fields will be user configured dynamic fields that are currently searchable. Batch Number is an existing fixed field and should already be searchable. Also, Accounting on the SuspenseScreen is handled by the GenerateAccounting element. This element will continue to be used to evaluate tests of screen values to determine when accounting will execute. As such, the existing configuration will support if accounting will run when the Master or the subsidiary records are saved or both

Prototype Configuration Detail

Overview

The ability is needed in OIPA for a parent – child relationship in Suspense. We need the ability to split a 'parent' single suspense item into 'children' suspense records. This feature proposes changes to the existing SuspenseScreen and SuspenseSearchScreen Screen rules. This feature will address the requirements of both Batch and Master Suspense enhancements. The SuspenseScreen will need to be able to display and save context data regarding the suspense record.

Prerequisites

Security to the 'Split' and 'SubSuspenseDelete' buttons should be provided in the 'SuspenseRecord' Company page which is in the path as below:

Admin Explorer > Security > Application Security > Security Groups > Prototype Super > Company Security > Company Pages > Prototype > SuspenseRecord

Changes to Existing Items

Screen name: SuspenseScreen

Company name: Prototype

Screen name: SuspenseSearchScreen

Company name: Prototype

Configuration Details

SuspenseScreen:

'MasterSuspenseGUID' Screen math Variable - Contain the SuspenseGUID of the parent suspense record which indicates that the current record is a sub suspense record.

'BatchNumber' Fixed field - Is configured and the value gets populated from Sub Suspense record when the parent suspense record is empty / When user key in the value in case of suspense record without having the sub suspense records.

<SplitSuspense> Element - Is set to 'Yes', System will display the 'Split' button on a suspense record and by using this button user can attach the

suspense records.

COPYMASTERSUSPENSE attribute - Is set to 'Yes', System populates the parent suspense record fixed and dynamic fields & their values to the attaching sub suspense record.

Below are the changes that can be noticed in PAS when user access the Suspense screen:

'Split' button will be available at the extreme lower right of the SuspenseScreen.

Upon selection of button user gets the below actions:

A pop-up Suspense entry screen will be appeared by populating the parent suspense record fixed and dynamic fields & their values, User can save the sub suspense record either with/without performing any changes to the suspense details.

Below are few of the cases tested for this feature:

If the amount of a sub suspense record exceeds the remaining available amount of a parent suspense record then system throws a business error "Available master suspense amount exceeded.

If the amount of a sub suspense record equals to the remaining available amount of a parent suspense record then the parent suspense record status will be updated to 'Closed'.

Upon save of a sub suspense record the BatchNumber of this record is copied to parent suspense record if it is empty and also the sub suspense record is saved as a normal suspense record.

In the Suspense results table, sub suspense records appeared exactly below the master suspense record.

If a sub suspense record status is 'Open' with the attached amount as zero then the trash can icon is displayed to the far right on the table and

upon selection of the icon system shadowed the sub suspense record.

If a sub suspense record status is 'Open' with the attached amount as greater than zero then the trash can icon is not displayed to this sub suspense record.

Within the Suspense table the shadowed sub suspense records are not displayed under parent suspense record.

Link navigation button is appeared to the left of the trash can icon, Once user clicked then the sub suspense record is opened to user.

SuspenseSearchScreen

Allow user to search the suspense records by using context fields and the config change done for this screen is inclusion of 'BatchNumber' search field and user is able to search the suspense records based on 'BatchNumber' search field.

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Client & Policy Search Enhancement Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Client & Policy Search Enhancement Prototype

Prototype Configuration Detail

Overview

The policy/client which are created using the attached business rules,i.e. CreateClients and CreatePolicy, do not get reversed.This enhancement supports reversal of newly created Client or Policy record.

Prerequisites

User should be able to add the "ReverselOfNewClientOrPolicy" transaction in the PAS.

New Items

A new transaction 'ReverselOfNewClientOrPolicy' is created at 'Prototype' Company Level.

Transaction name: ReverselOfNewClientOrPolicy

Company name : Prototype Company

Plan Name : Client Plan

Attached BR : CreateClient Business Rule

Changes to Existing Items

None.

Results:

When the user adds the transaction at the ClientLevel the user has to select 'Client' or 'Policy' depending on the user selection the corresponding BusinessRules will be executed.

If the user checks "CreateClient" option in Transaction:

- 1.Fields such as FirstName,LastName,ClientDateOfBirth,ClientSex will be shown to the user.User has to enter values for these fields.
- 2.Field ClientTaxID will be populated with autogenerated TaxID value into it
- 3.If the user selects the option as 'Shadow' in "Reversal Operation For Client" field then below action will be performed

The "CreateClient" Business Rule will be invoked and client will be created,when the user reverses this transaction then 'ClientRecord' should get 'Shadowed' and the field "StatusCode" in DB against this clientrecord should be set to "Shadowed".

- 4.If the user selects the option as 'Delete' in "OnReversalEventForCC" field then below action will be performed

The "CreateClient" Business Rule will be invoked and client will be created,when the user reverses this transaction then 'ClientRecord' should get deleted from DB.

If the user checks "CreatePolicy" option in Transaction:

- 1.Fields such as PolicyName,IssueStateCode,PolicyNumber will be shown to the user.User has to enter values into these fields.

2.If the user selects the option as Shadow in "Reversal Operation For Policy" field then below action will be performed

The "CreatePolicy" Business Rule will be invoked and Policy will be created,when the user reverses this transaction then 'PolicyRecord should get 'Shadowed' and the field "StatusCode" in DB against this policyrecord should be set to "Delete".

3.If the user selects the option as Delete in "Reversal Operation For Policy" field then below action will be performed

The "CreatePolicy" Business Rule will be invoked and Policy will be created,when the user reverses this transaction then 'PolicyRecord' should get deleted from DB.

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Group Billing_Create Bill

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Group Billing_Create Bill

A generic structure is defined for "Create Bill" that allows the system to use a common bill detail table to store the most granular level amounts that are then aggregated up to a bill record. The bill detail records are created before the bill as they may be calculated more frequently than the aggregated bill. Bill detail will normally be created at the policy or client level. A bill will normally be created at the client or group customer level and it will collect multiple bill detail records into a single 'bill' entity

Overview

As part of prototype configuration for this feature the following transactions are created and is outlined as below:

GenerateBillDetail - Generates BillDetail record using GenerateBillDetail Business Rule.

GenerateBill - Collects multiple BillDetail records and consolidates to a single Bill record using GenerateBill Business Rule.

MaintainBillDetail - BillDetail record details can be updated or shadowed using MaintainBill Business Rule.

Prerequisites

The PAS user should have access to 'GenerateBillDetail', 'GenerateBill' and 'MaintainBillDetail' transactions.

New Items

Transaction name: GenerateBillDetail

Company name: Prototype Group Child Company

Product name: Group Prototype Product

Business rules attached: GenerateBillDetail BR

Transaction name: GenerateBill

Company name: Prototype

Plan name: Customer Plan

Function used: FindBillDetail

Business rules attached: GenerateBill BR

Transaction name: MaintainBillDetail

Company name: Prototype Group Child Company

Product name: Group Prototype Product

Business rules attached: MaintainBillDetail BR

Screen name: BillDetailScreen

Company name: Prototype Group Child Company

Product name: Group Prototype Product

Configuration Details

GenerateBillDetail Transaction and GenerateBillDetail Business Rule:

It is a Product level transaction which has GenerateBillDetail Business Rule attached to it. Once the GenerateBillDetail activity is processed it generates BillDetail record in the PENDING status based on the details selected by user.

Fields configured in this transaction are outlined below:

BillDueDate: Specifies the Due date of the Bill to be paid

BillDueAmount: Specifies the Due Amount to be paid.

BillGroupType: It is a combo field which will list out the Bill Group types i.e.; Group and Class (Group is set as default).

GroupCustomerGUID: It fetches the GroupCustomer guid of the selected Policy.

Class: It is a hidden field and will be enabled only when 'Class' is selected in the BillGroupType and will list the

ClassGroups associated to Group Customer.

BillEntityType: It is a combo field which will list out the Bill Entity types i.e.; Client, Policy and Segment (Policy is set as default).

Client: It will list out the Clients associated to this policy (Attached to Roles) and will be shown only when 'Client' is selected in the BillEntityType.

Policy: It will fetch the policy number where the activity is added.

Segment: It will list out the Segments available for this policy and will be shown only when 'Segment' is selected as BillEntityType.

GenerateBillDetail transaction which will pass the information like ReceivableDueType, BillGroupGUID,

BillGroupType, BillEntityType, BillEntityGUID, BillDueDate, BillDueAmount and BillDetailCreationDate to

GenerateBillDetail Business Rule for creating the BillDetail record.

GenerateBill Transaction and GenerateBill Business Rule:

It is a Customer level transaction which uses FindBillDetail function and also GenerateBill BR is attached to it. GenerateBill transaction will map the list of BillDetail records to a single Bill record and this information will be available in AsBillDetailGroup table.

Fields configured in this transaction are outlined below:

BillingStartDate: Beginning date for the range of bill detail records to be included and passed to the FindBillDetail activity function.

BillingEndDate: End date for the range of bill detail records to be included and passed to the FindBillDetail activity function.

BillGroupType: It is a combo field which will list out the Bill Group types i.e.; Group and Class (Group is set as default).

GroupCustomer: It fetches the GroupCustomer guid.

Class: It is a hidden field and will be shown only when 'Class' is selected in the BillGroupType and will list the Classes associated to Group Customer.

ThresholdMinimumAmount: The minimum due amount for a bill detail record to be included and passed to the FindBillDetail activity function.

ThresholdMaximumAmount: The maximum due amount for a bill detail record to be included and passed to the FindBillDetail activity function.

The GenerateBill transaction has a ReferenceIDMV Math Variable of type Identifier, this is used in case of undo/redo of the activity and also BillOwner type will be same as BillGroup type.

Activity function FindBillDetail is used in the GenerateBill Transaction which outputs an array of AsBillDetail records that should be grouped together to create a Bill.

Information like BillDetailArray, BillOwnerGUID, BillOwnerType and ReferenceID is passed to GenerateBill BR (From Transactionb Math) for the creation of Bill record.

MaintainBillDetail Transaction and MaintainBillDetail Business Rule:

It is a Product level transaction which has MaintainBillDetail BR attached to it. MaintainBillDetail transaction is used to update the amount details of a BillDetail record or shadowing of the BillDetail record based on user selection.

Fields configured in this transaction are outlined below:

BillDetailGUID: It will list out the BillDetail records which are in the Pending status.

ShadowBillDetail: Is a Checkbox field and can be used to shadow the selected BillDetail record.

ExistingAmount: It will fetch the current Amount on the selected BillDetail record.

NewAmount: It is a user enterable Currency field which will replace the ExistingAmount upon processing of the activity.

MaintainBillDetail transaction which will pass the information like ShadowBillDetail, BillDetailGUID, BillDetailChangeDate and NewAmount to MaintainBillDetail Business Rule in order to perform either update or shadow operation on the selected Bill Detail record.

BillDetailScreen Business Rule:

It is Product level Screen rule. BillDetailScreen defines the dynamic fields available to a bill detail record which will be added in the BillDetailField table.

Company name: Prototype Group Child Company

Product name: Group Prototype Product

Fields configured in this Business rule are outlined below:

BillDetailCreationDate: Creation date of BillDetail record. This field is created and updated in AsBillDetailField upon processing of 'GenerateBillDetail' activity.

BillDetailChangeDate: Change date of BillDetail record is updated or the record gets shadowed based on user inputs and upon processing of 'MaintainBillDetail' activity.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Group Billing_Data Intake (DI)

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Group Billing_Data Intake (DI)

The Group Billing Data intake functionality deals with storing, reading and processing the incoming payment information. Data Intake activity will process the reconciliation and generate the required policy level activities to ensure the monies are adjusted to the policy.

OIPA will use Data Intake method to get the payment and billing reconciliation related information. Data Intake functionality is enhanced to receive a Data Intake Payment File and corresponding Payment records, and executing the file as set of changes in the system

Files & Records

Payment information will come in Data Intake file.

Data Intake payment file can have one or more Data intake record.

Each payment and related information will represent a record in data intake payment file.



Prototype Configuration Detail

Overview

Once a Group Bill is generated to a Group Customer, when monies are received from the payer, the same will be adjusted against the Bill Line entries and reconciled through this feature where DI deals with storing, reading and processing the incoming payment information.

Prerequisites

1. PAS user should be associated to a security group where the access is given for 'Prototype' company.
2. Bill Detail records should exist at either of these entities Policy/Segment/Client.
3. Bill record should exist for GroupCustomer which groups the above bill detail records.
4. Data Intake profile against 'Payment Prototype' intake profile definition should exist for a GroupCustomer.
5. Data Intake file for 'Payment Prototype' intake profile definition should exist.
6. PAS user should be associated to a security group where the access is given for 'Prototype' company.
7. PAS user should have access to 'GenerateBillDetail' & 'Reconciliation' product level transactions.
8. PAS user should have access to 'GenerateBill', 'PaymentFilePrototype' & 'PaymentRecordPrototype' transactions present at 'Customer Plan' level.
9. PAS user should have access to 'Reconciliation' transaction present at 'Client Plan' level.

New Items

Intake Profile Definition:

New intake profile definition 'Payment Prototype' is added for 'Prototype' company.

Below is the sample DI file template for 'Payment Prototype' intake profile definition:

PaymentPrototype-DIFileTemplate.txt

PaymentFilePrototype transaction:

Plan Name - Customer Plan
Company Name - Prototype
Transaction Type - IntakeFile
Processing Order - 200

With this transaction the Data Intake file gets validated by Data Intake engine and then returns the count of received payment records to intake file screen by using 'CopyToIntakeFileFields' business rule.

PaymentRecordPrototype transaction:

Plan Name - Customer Plan
Company Name - Prototype
Transaction Type - Intake-Record
Processing Order - 201

With the execution of this transaction the 'Reconciliation' activity will be added along with the suspense details at respective entity level (Policy / Client) against the processed Payment Reference record.

Note: There will be separate 'Reconciliation' activity added for each processed Payment Reference record, If the Payment Reference Record is associated to either Policy/Segment entity type then the 'Reconciliation' occurs at Policy level and if the record is associated to 'Client' then the 'Reconciliation' occurs at Client level.

GenerateSuspense Business Rule

This attached business rule will create a suspense record (With the amount as sum of all the payment reference records amount) and then pass the created suspense information to activity math section.

Test Scenario:

1. Consider a policy associated to a GroupCustomer plan.
2. Access the policy and then create a Bill Detail against the Policy entity by using 'GenerateBillDetail' transaction.
3. Access the GroupCustomer (where the above policy belongs to) and then create a Bill (Such that the above bill detail record is grouped) against the GroupCustomer by using 'GenerateBill' transaction.
4. Prepare the DI (Data intake) file based on above generated Bill detail record.
5. Using DI utility execute the above DI file.
6. With the successful execution of DI file the Bill detail record should be in reconciled state.

Changes to Existing Items

None

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Group
Billing_Reconciliation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Group Billing_Reconciliation

The Billing Reconciliation functionality adds new screens, business rules and transactions to process the reconciliation and generate the required policy or client level activities to ensure the monies are applied to the bill detail records.

The Billing Reconciliation is accomplished through two screens; Group/List Bill Search Screen and Group/List Bill Reconciliation Screen. From the Bill Reconciliation Screen, based on the reconciliation action is the ability to trigger specific actions into the Policy/ /Client/ from which the bill detail originates. The same reconciliation could be processed using data intake to process the reconciliation activities.the following

Business Rules support the Reconciliation process:

Group/List Bill Search (screen)

Group/List Bill Reconciliation (screen)

Bill Detail Search (screen)

Bill Detail Reconciliation Search (screen)

MaintainBillDetailReconciliation (Business Rule)



Prototype Configuration Detail

Overview

With this feature user has the ability to reconcile the bill such that the

monies are adjusted against the bill detail.

Prerequisites

Following Business Rules should be configured:

BillSearchScreen
BillDetailReconciliationSearchScreen
BillDetailSearchScreen
BillReconciliationScreen

User should have access to the following transactions:

Reconciliation
ReconciliationReversal

New Items

'BillSearchScreen' Business Rule:

Business Rule is configured to showcase the 'Group/List Bill' search feature.

'Bill Search' screen can be accessed through a new main menu item called 'Group/List Bill'.

<Group> values like 'Bill', 'BillDetail', 'BillDetailField' & 'GroupCustomer' are supported for search criteria fields.

Following are the search criteria's configured for prototype scenario:

BillDetailReferenceID
CustomerNumber

For Bill records with BillOwnerTypeCode as Group Customer, the Group Customer needs to be linked through the BillOwnerGUID as the ClientGUID of the Group Customer.

For Bill records with BillOwnerTypeCode as Class, the Group Customer needs to be linked through the BillOwnerGUID as the ClassGUID. Search results are displayed as per the configuration and also default sorted by BillReferenceID.

Following are the fields configured as part of results section:

BillReferenceID
CustomerNumber
BillGeneratedDate

Upon selection of any one of the search result then user is navigated to the 'Group/List Bill Reconciliation' Screen.

'BillDetailReconciliationSearchScreen' Business Rule:

Business Rule is configured to showcase the BillDetailReconciliationSearch feature which supports editing of the reconciled bill detail records.

Following are the search criteria's configured for prototype scenario:

Amount
SuspenseNumber
StatusCode
<Group> values like 'BillDetailReconciliation', 'BillDetail', 'BillDetailField' & 'GroupCustomer', 'Client', 'ClientField', 'Policy', 'PolicyField', 'Segment' & 'SegmentField' are supported for search criteria fields.

Following are the fields configured as part of results section:

BillDetailReferenceID
PolicyNumber
SuspenseNumber

Results are filtered based on the criteria entered by the user and upon selection of any one of the search result then user is navigated to the

'Group/List Bill Reconciliation' Screen.

Bill Detail Reconciliation search screen allows user to edit the bill amounts for multiple times before clicking of 'Save' / 'Cancel' button on the Group/List Bill

Reconciliation Screen.

When user selects 'Cancel' button then the Adjust context is not considered and also user is returned back to initial load of 'Group/List Bill Reconciliation' Screen.

When user selects 'Save' then a confirmation window is appeared and upon confirming the reconciliation entries is saved and associated policy/client/group customer activity is triggered.

'BillDetailSearchScreen' Business Rule

Business Rule is used for the LinkSuspense purpose and the 'BillDetailSearchScreen' is appeared as one of the part of 'Bill Detail Search' section.

Following are the search criteria's configured for prototype scenario:

Amount

BillGroupType

PolicyNumber

<Group> values like 'BillDetail','BillDetailField' & 'GroupCustomer','Client','ClientField','Policy','PolicyField','Segment' & 'SegmentField' are supported for search criteria fields.

Following are the fields configured as part of results section:

Amount

PolicyNumber

Results are filtered based on the criteria entered by the user.

'BillReconciliationScreen' Business Rule

This screen allows the user to reconcile the Group/List Bill that is selected.

Header section of 'Bill Reconciliation' screen is displayed with the below information:

Bill Reference ID
Group Customer Number
Class Name
Group Bill Date

The next section is 'Bill Reconciliation' section which allows a user to perform reconciliation and below are the data displayed as part of this section:

Total Amount Due
Current Reconciled Amount

By using either 'Link Suspense' or 'Adjust' buttons user can reconcile the bill details.

When user selects 'Cancel' button then the Adjust/Link Suspense context is not considered and also user is returned back to initial load of 'Group/List Bill

'Reconciliation' Screen.

When user selects 'Save' then a confirmation window is appeared and upon confirming the reconciliation entries is saved and associated policy/client/group customer activity is triggered.

'Reconciliation' Transaction

Transaction Type - Policy Financial

Processing Order - 1500

This transaction is called under the <ReconciliationTransactionOnEntity> element of 'BillReconciliationScreen' BR.

Following are the set of fields configured for this transaction where the data is populated from 'BillReconciliationScreen'.

BillDetailReferenceGUID

SuspenseNumber

AmounttoReconcile

'MaintainBillDetailReconciliation' Business Rule is attached to this transaction in order to allow reconciliation to be processed.

After processing the transaction the bill detail record status is changed to Reconciled if the entire amount of the bill detail is reconciled..

'ReconciliationReversal' Transaction

Transaction Type - Policy Financial

Processing Order - 1000

This transaction is called under the

<ReconciliationReversalTransactionOnEntity> element of 'BillReconciliationScreen' Business Rule.

Following are the set of fields configured for this transaction where the data is populated from 'BillReconciliationScreen'.

OriginatingActivityGUID

'ShadowPendingActivities' Business Rule is attached to this transaction in order to shadow the reversed reconciliation record associated to the OriginatingActivityGUID.

After processing the transaction user should be able to see that Reconciliation done has been reversed and the status of the bill detail record is changed to Under reconciliation.

Note: Both the 'Reconciliation' & 'ReconciliationReversal' transactions are available under the below levels of Prototype company:

Client Plan
Customer Plan
Group Prototype Product

Existing Items

None

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Migrate Business Rule Changes to the system without an Application Restart

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Migrate Business Rule Changes to the system without an Application Restart

Currently in the PAS system when the application.mode in the PAS.properties file is set to "Production" the application does not regularly refresh the cache for control reasons. This means that when Rules Palette migrates configuration to an environment where the application.mode is set to "Production", the PAS application must be manually restarted in order to acknowledge and utilize the new configuration. To avoid this situation, the PAS system has been modified to clear cache based on specific settings activated by release management when a package is successfully deployed.

This is a requirement that is part of OIPA's capability to be 24x7 available. This is especially critical in the context of group administration since coverage changes within a group could be made from OIPA and such changes may include rule changes from rules palette which need to be migrated more often than usual rule migrations which would follow a periodic schedule

Currently when a migration package is deployed to an OIPA environment, the application must be restarted manually to clear the cache and pick up the new configuration. This is applicable to all deployments to all OIPA environments.

OIPA must be enhanced to accept a request to clear the OIPA cache when application.mode is set to "PRODUCTION".

Two components are needed: a new sequence record and a new application property for all OIPA property files.

An AsSequence record named ConfigurationUpdatedGMT.

The record contains a GMT marking completion of a Rules Migration.

The request for a cache clearing is associated to Palette's update of this sequence record.

A property in called `application.configurationCacheTimeout`.

This will denote the time in minutes that OIPA should keep objects in cache before checking for updates.

A value of 0 or less or the property is missing (compatibility): OIPA will never check for updates to `ConfigurationUpdatedGMT`.

A value greater than 0: OIPA will check for updates to `ConfigurationUpdatedGMT` after the specified number of minutes has passed.

OIPA process to determine when to clear the cache.

OIPA keeps in cache the value of `ConfigurationUpdatedGMT` from `AsSequence`.

This value must be obtained from `AsSequence` upon application restart.

The value in the cache is updated to the value in `AsSequence` after the cache is cleared.

When the application.mode indicates "PRODUCTION" and objects have been held in the configuration cache for the requisite number of minutes indicated by application.configurationCacheTimeout:

If OIPA's cached value of ConfigurationUpdatedGMT is equal to the value of ConfigurationUpdatedGMT from AsSequence, the configuration cache is left untouched.

If OIPA's cached value of ConfigurationUpdatedGMT is less than the value of ConfigurationUpdatedGMT from AsSequence, the configuration cache is cleared.

When the cache is cleared, OIPA stores the new value of ConfigurationUpdatedGMT from AsSequence in the cache.

Clearing the configuration cache forces OIPA to go to the database to fetch rules and configuration data (no change to existing functionality).

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Money Field
Justification

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Money Field Justification

Configuration can control whether money fields are left or right justified. An optional element, <MoneyFieldAlignment> is available in the CompanyCosmetics business rule to indicate if money fields display as left or right justified.

The allowed element values are "Left" or "Right".. If any value other than the allowed values are configured, the system will ignore the configuration and the default alignment is applied. The default alignment is 'Right'. This formatting applies to all money fields that fall under the applicable primary company level.. The following is the example of the Prototype:

Prerequisites

'CompanyCosmetics' Business Rule overrides should be present for the below list of primary companies - Alamere Group Insurance, International, Prototype

The following changes are made in 'CompanyCosmetics' Business Rule

Prototype configuration for 'CompanyCosmetics' Business Rule override is done as below:.

'Alamere Group Insurance' Company: <MoneyFieldAlignment> element is not configured.

Result: In OIPA all money field values display right justified (default behavior)

'International' Company: <MoneyFieldAlignment> element is configured with value as 'Right'.

<MoneyFieldAlignment>Right</MoneyFieldAlignment>

Result: In OIPA all money field values display right justified.

'Prototype' Company: <MoneyFieldAlignment> element is configured with value as 'Left'.

<MoneyFieldAlignment>Left</MoneyFieldAlignment>

Result: In OIPA all money field values display left justified

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > MultiSuspense Handling for Group Billing

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

MultiSuspense Handling for Group Billing

This MultiSuspense feature extends legacy functionality under the group billing context. The feature allows the system to process multiple suspense items in an activity (as supported in prior and current versions of PAS) but the UI has been modified to bring it in line with other multi pick screens in PAS such as the AllocationScreen. Prior versions of the system allowed users to save suspense items to which attachments may be made with no specified allotment of money. Whereas the current version allows users to save suspense items only when specified allotments of money are made. The prior version capabilities have been returned as well

Prototype Configuration Detail

Overview

Configuration is set up to view the remaining available amount of a suspense record.

Prerequisites

1. 'SuspenseSearchScreen' businessrule should be available.
2. Transaction having multiple suspense support.

Changes to Existing Items

SuspenseSearchScreen Business Rule:

Added the new result column 'AvailableAmount' to the existing Business Rule override present at 'Prototype' Company level. This is a virtual column reference and does not exist in the database but is calculated on the fly by subtracting the attached amount from the original suspense amount.

Transaction:

1. Access the transaction that supports the addition of multiple suspense records.
2. Navigate to the 'Suspense' tab of above activity.
3. Search and select the Suspense Number and then tab out.

Result:

New result column 'Available Amount' will be shown with the unattached amount details of the selected suspense record.

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Nodes Deletion from Agreement Tree

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Nodes Deletion from Agreement Tree

OIPA does not provide ability to delete the Agreement from Agreement tree. Delete option is required to provide the ability to delete an agreement in case user wants to. As a part of group gaps this functionality is included in 10.1 wherein the feature improves the business user experience by accessing and deleting a Group Customer's Agreements/Agreement data

This new feature will allow the user to delete the Group Customer's Agreements.

Prerequisites

User should provide access to 'Delete' button for 'Agreement' Company page present at 'Prototype' and 'Alamere Group Insurance' SecurityGroup levels.

Changes to Existing Items

'AgreementScreen':

Security for 'Delete' button is provided only for 'Alamere Group Super' and 'Prototype Super' Security group.

Business Rules override present at "Prototype" company and "Alamere Group Insurance" company level has been modified to showcase this new feature

Below are the details of the configuration updates:

Business Rules override present at "Prototype" Company:
New tag <AllowDelete> is configured with only one sub element <StatusCode> with value as "Pending".

Result:

If security is present for 'Delete' Button:

System allows the user to delete the agreements which are in "Pending" Status and also checks if agreement is not associated to any of the following agreement entities Role/Plan/ClassGroup/Product

If it is associated then 'Cannot delete the agreement hierarchy since there is a data for one or more of the agreement entities

Role/Plan/Classgroup/Product.'

error message will be displayed.

If security is Not present for 'Delete' Button: System does not show 'Delete' button.

Business Rules override present at "Alamere Group Insurance" Company:

New tag <AllowDelete> is configured with sub element <StatusCode> with values as "Pending" and "Active".

Result:

If security is present for 'Delete' Button:

System allows the user to delete the agreements which are either in "Pending" Or in "Active" Status and also checks if agreement is not associated to any of the following agreement entities

Role/Plan/ClassGroup/Product

If it is associated then 'Cannot delete the agreement hierarchy since there is a data for one or more of the agreement entities

Role/Plan/Classgroup/Product.'

error message will be displayed.

If security is Not present for 'Delete' Button: System does not show 'Delete' button.

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Security at Product Level

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Security at Product Level

A Product is defined in Group as a Group Benefits Plan Template which provides the basis for a Group Customer's Plan instance and/or serves to group similar Group Benefits Plan Templates (referred to as "Sub-Products"). When Plans are created they will belong to either a Product or Sub-Product. Plans can be created through Group Customer screens or through Palette.

For group policy administration, user has the ability to create plan dynamically using OIPA application. That means user can create plan without using palette and avoids the need of migrating the changes in production environment. But this ability gets defeated as there is no way to give plan level security dynamically. System has the ability to define plan level security at parent product level and all plans should automatically inherit it. This feature will address this need and allow user to create /use plan dynamically without involving any changes to palette. It will also avoid the need for frequent migration of security data from development environment to production environment.

Security Consideration

When a Plan is created through OIPA, plans would receive a copy of parent Product security as default. All levels of security at company and plans as defined in palette , all security levels defined at the immediate parent and all transaction security available to the plan will be copied.

Any Change in permissions for a Plan from this default would need to be made through Palette for that plan

For the copied permissions for the plans created through OIPA to take effect, logout/login back to OIPA is permissible although not preferable

If created in Palette; the security is set up through Plan Security Node in Admin Explorer. When Created through OIPA, at present user is forced to

set up security via palette to access the plan pages in OIPA. The palette changes and security changes defined in this document will allow user to set up security at Product level or sub product level which a plan can use as default. For Plans created in OIPA, this would allow user to access plan Pages without the need to set up security permissions via palette.

Setting up Plan Pages security at Product Level or Child Product Level in Palette

To set up Product level Security, user would need to navigate to the Admin Explorer in Palette

User opens Security | Application Security | Security Group | Name of the Security Group |Plan Security

Expanding Plan Security Node currently lists all companies and subsidiary Companies in OIPA at the same level



Under each subsidiary company, all products available under it will be displayed



Expanding a Product name will display the following :

1. Product Name Node
2. Plan Pages Folder
3. Plan Folder (Contains individual plan nodes for all plans available under this Product)
4. hierarchically display Child Product Nodes if any



Security can be set up at Product level for the plan pages. Plans under the product will not inherit this security automatically .Any changes to the Plan pages security at product level would NOT propagate downstream. But any plan created in OIPA under the product will copy this security set up as default .

For any Child Products under the Product hierarchy. Plans under the child product will not inherit child product security automatically .Any changes to the Plan pages security at Child product level would NOT propagate downstream . But any plan created in OIPA under the child product will copy this security at set up as default .

When a Product or Child Product or Plans are created in Palette, there will not be any default security. User would need to go to Admin Explorer and set up the security permissions for plan pages explicitly at any level

Provide right click menu options at each parent or child Product name nodes that will allow user provision to Grant Access to all plan pages or Remove access to all plan pages. User can also individually set security by opening plan pages folder and navigating to each plan page security node



Additional to above two right click menu options, provide option at Child Product/s and plan levels on right click to "Copy Parent Access". This will allow option for user to copy a parents security permissions to a child instead of creating from scratch. User can then edit those permissions or keep it as is for that level. Please note that this option to copy parent access is available only if security permissions are available at immediate parent. If security is not set up at immediate parent , do not show this option on right click.

Wherever copy parent access option is allowed and user selects it , give a message which user would confirm with OK as: "Existing security will be overwritten if this operation is completed.

Before a plan security is set up , it is not required to set up Product and Child Product security in the hierarchical structure, but it is up to the configure to take care of the fact that if a plan is created dynamically in OIPA, there needs to be security permissions available at the immediate parent for plans to be accessible for the users without setting up permissions in palette for those plans

While checking in security permissions at product and Child product level in Palette, give the following warning message when there are children in lower hierarchy (plans or child products) available : "The security permissions will not be automatically copied down to products or plans in lower hierarchy"

All Plan level security set up functionality available today will not change and will continue to be available as is.

Setting up Transaction Security in Palette at Product/ Child product Levels

After Company, Product and /or Plan security have been defined, the transactions associated with the company, Product and plans are displayed under the Transaction Security folder in Admin Explorer

Currently all companies and subsidiary companies will be displayed under the transaction security node at the same level like in plan security Under each subsidiary company , all product name nodes will appear



Opening the individual products will display Transactions Folder that would list all transactions under the product , Plans Folder ,Child Products Folder in that specific order. Within Plan Folder, all Plans under the Product will be displayed . Opening the node for the plans will display all plan transactions



Within the child product folder all Child product names available will be displayed. Under each Child Products, the hierarchical display of folders will be similar to Product, as shown in below image



Security can be added or removed to all transactions in a product by right-clicking on the product name node. Security can also be assigned to individual transactions by opening the Product folder and then the transactions folder and then selecting specific transaction.

Security can be added or removed to all transactions in a child product/s by right-clicking on the child product's name. Security can also be assigned to individual transactions by opening the child Product folder and selecting a specific transaction. Plans under the child product will be displayed under child product hierarchically

Opening up plan folder will list all plans under the Product or child product/s. Security can be set up for Plan transactions as it is currently

When a transaction is created at Product level , those transaction nodes will be added to downstream child products and Plans in Palette currently. When transactions are created at child product level, those transaction nodes are also added to Plans underneath. In palette transaction security , allow option at Child Product and Plans below to copy parent transaction security permissions access on right click. This option will be available only if security permission setting are available for those transactions at immediate parent level. The right click menu option will be "Copy Parent Access ". When user selects this option , give a warning as : "Existing security will be overwritten if this operation is completed. Do you wish to continue?"

While checking in security permissions at product and Child product level transactions in Palette, give the following warning message when there are children in lower hierarchy (plans or child products) available : "The security permissions will not be automatically copied down to products or

plans transactions in lower hierarchy"

Configuration Detail :

This feature has not introduced any new configuration or changes to existing configuration, but plan pages under product and transactions currently within the product hierarchy would need security manually added.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Security Needed by Inquiry Screen

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Security Needed by Inquiry Screen

Currently the ability to restrict creation of new policy records by a user is accomplished by restricting the access to “Save” button in the Policy screen. In a similar way, Client/Group Customer creation is restricted through restricting the access to “Save” button in the Client/Group Customer screen. This limits a user from both creating a new record and editing an existing record. This enhancement provides the ability to have a separate security setting to allow/restrict creation of new Policy/Client/Group Customer which can be used on top of the existing “Save” button authorization to limit New record creation without affecting the ability to edit existing records.

OIPA supports 3 different levels of Inquiry Screen rules accessed off of the Main Menu, Policy screens or Client/Group Customer screens.

Main Menu Level

Currently access is granted by authorizing the Company Security > Company Pages > 'Primary Company' > Inquiry security page. If access to that security page is given, any member of that Security Group will be allowed to select any InquiryScreen rule created at the Main Menu level for the applicable Primary Company.

The 'Inquiry' main menu button will now be visible if the Security Group has authorization for one or more Main Menu level InquiryScreen rules for the applicable Primary Company

When the user hovers over the Inquiry button the list of Main Menu level InquiryScreen Names/Rules the Security Group has access to will be displayed. OIPA will use best match to determine which version of the Inquiry Screen rule is applicable.

If the Security Group does not have access to at least one Main Menu level InquiryScreen rule for the applicable Primary Company the 'Inquiry' button will not be visible on the Main Menu.



Policy Level

Currently access to the Inquiry button (off of the Policy pages) is granted for each 'Policy' page by selecting the 'Inquiry' button on the applicable security pages

For example, Plan Security > 'Company' > 'Plan' > Plan Pages > Policy security page - Inquiry button.

If authorization is granted the Security Group/user can select any Policy level InquiryScreen Name/Rule applicable to that policy.

The 'Inquiry' button will now be visible from any 'Policy' page as long as the Security Group has access to at least one 'Policy' level InquiryScreen name/rule for the policy's plan, product(s) or company.

When the user hovers over the Inquiry button from any Policy page, the InquiryScreen Names/Rules the Security Group has access to will be displayed. OIPA will use best match to determine which version of the Inquiry Screen rule is applicable for this policy (see screen shot below)

During analysis of this feature it was discovered that the PolicyCommentsSearch and FinalAction security pages do not currently include the 'Inquiry' button security option, this appears to have been an oversight when those features were implemented. As part of this feature the 'Inquiry' menu option should be available from those 'Policy' screens (as well as any other 'Policy' screen) if the appropriate InquiryScreen

name/rule overrides and security exist.

If the Security Group does not have access to any Policy Level InquiryScreen name/rule then the Inquiry button will not appear on any 'Policy' screens.



Client Level

Currently access to the Inquiry button (off of the Client or Group Customer pages) is granted for each 'Client' or 'Group Customer' page by selecting the 'Inquiry' button on the applicable security pages

For example, Company Security > Company Pages > 'Primary Company' > Client security page - Inquiry button.

If authorization is granted the Security Group/user can select any Policy level InquiryScreen Name/Rule applicable to that policy.

The 'Inquiry' button will be viewable from any 'Client' or 'Group Customer' page as long as the Security Group has been granted access to at least one

'Client' level InquiryScreen Name/Rule for the applicable Primary Company.

When the user hovers over the Inquiry button from any Client or Group Customer page, the InquiryScreen Names/Rules the Security Group has access to will be displayed.

During analysis of this feature it was discovered that the ClientsCommentSearch security page does not include the option for the 'Inquiry' button, this appears to have been an oversight

when the feature was implemented. As part of the feature the 'Inquiry' menu option should be available from that 'Client' screen (as well as all other 'Client' and 'Group Customer' screens) if the appropriate security exists.

If the Security Group does not have access to any Client Level InquiryScreen name/rule then the Inquiry button will not appear on any 'Client' or 'Group Customer' screens.



Configuration Details

This change does not require any configuration changes.

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Security on Main Menu & Client View Buttons

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Security on Main Menu & Client View Buttons

In OIPA, currently the ability to restrict the creation of new Policy, Client or (Group) Customer record by a user is to limit access to the 'Save' button (on the applicable security page). This prohibits both saving a new Policy, (Group) Customer, Client or editing an existing one. In order to have a separate security setting to allow/restrict creation of new Policy/Client/(Group) Customer apart from the authorization setting on the "Save" button, we have enabled the security setup. To ensure some users/security groups are provided access to edit records without access to new Policy/Client/(Group) Customer creation. The following are the examples of the Prototype Configuration for Policy/Client/Group customer level.

Prototype Configuration Detail

Prerequisites

The following are the security user considered for showcasing of prototype scenarios:

Prototype Super
Prototype Tester
Prototype Analyst

Existing Items

'Prototype Super' Security Group:

'Policy' Plan Page:

Security for 'New' button is present by default for the entire plans of the below companies:

Prototype Group Child Company
Prototype Individual Child Company

1. 'Policy' plan page:

Admin Explorer > Security > Application Security > Security Groups > Prototype Super > Plan Security > Prototype Individual Child Company > Dynamic Prototype Plan > Plan Pages > Policy

Once security for 'New' button is available/given then by default security for 'Save' button is also available.

Result: In OIPA when the user associated to 'Prototype Super' security logs in then the 'New' sub menu option under 'Policy' main menu is shown and also user is able to 'Save' the existing/new policy content.

2. 'Client' Company Page:

Security for 'New' button is present by default in 'Client' Company page which is in the path as mention below:

Admin Explorer > Security > Application Security > Security Groups > Prototype Super > Company Security > Company Pages > Prototype > Client

Once Security for 'New' button is available/given then by default security for 'Save' button is also available.

Result: In OIPA, when user associated to 'Prototype Super' security group is logged in then the 'New' sub menu option under 'Client' main menu is shown and also user is able to 'Save' the existing/new client content.

3. 'GroupCustomer' Company Page:

Security for 'New' button is present by default in 'GroupCustomer' Company page which is in the path as mention below:

Admin Explorer > Security > Application Security > Security Groups > Prototype Super > Company Security > Company Pages > Prototype > GroupCustomer.

Once 'New' button is available/given then by default security for 'Save' button is also available.

Result: In OIPA, when user associated to 'Prototype Super' security is logged in then the 'New' sub menu option under 'Group Customer' main menu is shown and also user is able to 'Save' the existing/new Group Customer content.

'Prototype Tester' Security user:

1. 'Policy' Plan Page:

Security is removed for 'New' button and security to 'Save' button is given in the 'Policy' Plan Page of Functional Prototype Plan which is in the path as mention below:

Admin Explorer > Security > Application Security > Security Groups > Prototype Tester > Plan Security > Prototype Individual Child company > Functional Prototype Plan > Plan Pages > Policy

Result: In OIPA, when the user associated to 'Prototype Tester' is logged in, 'New' sub menu option under 'Policy' main menu is not shown but 'Search' sub menu option is shown and also user is able to 'Save' the existing Policy content of Functional Prototype Plan.

2. 'Client' Company Page:

Security is removed for 'New' button and security to 'Save' button is given in the 'Client' Company page which is in the path as mention below:

Admin Explorer > Security > Application Security > Security Groups >

Prototype Tester > Company Security > Company Pages > Prototype > Client

Result: In OIPA, when the user associated to 'Prototype Tester' security is logged in, 'New' sub menu option under 'Client' main menu is not shown but 'Search' sub menu option is shown and also user is able to 'Save' the existing Client contents.

3. 'Group Customer' Company Page:

Security is removed for 'New' button and security to 'Save' button is given in the 'Group Customer' Company page which is in the path as mention below:

Admin Explorer > Security > Application Security > Security Groups > Prototype Tester > Company Security > Company Pages > Prototype > Group Customer.

Result: In OIPA, when the user associated to 'Prototype Tester' security is logged in, 'New' sub menu option under 'Group Customer' main menu is not shown but 'Search' sub menu option is shown and also user is able to 'Save' the existing Group Customer contents.

'Prototype Analyst' Security user:

1. 'Policy' Plan Page:

Security is removed for both 'New' and 'Save' buttons in the 'Policy' Plan Page of of Dynamic Prototype Plan which is in the path as mention below:

Admin Explorer > Security > Application Security > Security Groups > Prototype Analyst > Plan Security > Prototype Individual Child company > Dynamic Prototype Plan > Plan Pages > Policy

Result: In OIPA, when the user associated to 'Prototype Analyst' security is logged in, 'New' sub menu option under 'Policy' main menu is not

shown and also 'Save' button is not available for the existing Dynamic Prototype Plan Policies.

2. 'Client' Company Page:

Security is removed for both 'New' and 'Save' buttons in the 'Client' Company page which is in the path as mention below

Admin Explorer > Security > Application Security > Security Groups > Prototype Analyst > Company Security > Company Pages > Prototype > Client

Result: In OIPA, when the user associated to 'Prototype Analyst' security is logged in, 'New' sub menu option under 'Client' main menu is not shown and also 'Save' button is not available for the existing Clients.

3. 'Group Customer' Company Page:

Security is removed for both 'New' and 'Save' buttons in the 'GroupCustomer' Company page which is in the as mention below

Admin Explorer > Security > Application Security > Security Groups > Prototype Analyst > Company Security > Company Pages > Prototype > Group Customer.

Result: In OIPA, when the user associated to 'Prototype Analyst' security is logged in, 'New' sub menu option under 'Group Customer' main menu is not shown and also 'Save' button is not available for the existing Group Customers.

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Segment Definition to include User Enterable and Static fields

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Segment Definition to include User Enterable and Static fields

Currently in case of Group customer s 'Plan Segment' section, when a user edits a segment, the details displayed on the screen for editing are the actual Segment fields and the same is saved under AsPlanSegmentNameFields table. This data currently is obsolete as the data is pertaining to a specific segment instance and not specific to a Plan Segment Name. This functionality is enabled to capture PlanSegmentName level fields based on the configuration of PlanSegmentName definition

Prototype Configuration Detail

Overview

Currently in case of Group customers 'Plan Segment' section when a user edits a segment, the details displayed on the screen for editing are the actual Segment fields and the same is saved under AsPlanSegmentNameFields table.

This data currently is obsolete as the data is pertaining to a specific segment instance and not specific to a Plan Segment Name.

This functionality will enable us to capture PlanSegmentName level fields based on the configuration of PlanSegmentName definition.

Prerequisites

User should have access to the BaseCoverageBasic Segment of Group Prototype Product.

Changes to Existing Items

Segment name: BaseCoverageBasic

New Item: PlanSegment

Company name: Prototype Group Child Company

Product name: Group Prototype Product

Configuration Details

BaseCoverageBasic Segment and PlanSegment :

Existing BaseCoverageBasic segment is used for the prototype configuration. Along with the existing configuration of the BaseCoverageBasic segment which will be in the Xml section, PlanSegmentFields, PlanSegmentEvents and PlanSegment tabs are added.

When the user load the Segment in Palette it will be having three more tabs along with the existing ones:

PlanSegmentFields – Provides the visual editor for PlanSegment Fields

PlanSegmentEvent - Provides the option for PlanSegment Action/events configured

PlanSegment - Provides the configuration of PlanSegment.

Xml under Segment tab will go in XMLData column as the way it does today

ResolveRule option will give two option i.e Segment and PlanSegment to allow user choose which rule to resolve.

SegmentMaturityAge : Specify the duration of the Segment which is set to 60.

In the Application when the user navigates to the 'PlanSegment' section of the Agreement of a GroupCustomer, user should be able to view the fields SegmentMaximumFaceAmount and SegmentMaturityAge in PlanSegment section instead of the fields configured the Xml section of the Segment.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for 10.1.1 Release > Time Slice for Plans

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Time Slice for Plans

The "time slice" is a way of recording the changes of an entity across the time period. It provides a way for a user to enter the data on the entity screen instead of using the Activities. OIPA already support time slicing for Class Group and Client relationship entities. This feature is to create the similar functionality to support the time slicing for the Plan and PlanSegment entity. The ability to record the changes for Plan and PlanSegment data across the time period was kept out of scope while considering wildcat scope (OIPA V10.0). This feature is to build the time slice functionality for Plan and PlanSegment entity and allow system to create/update plan and PlanSegment with appropriate time slice data.

Prototype Configuration Detail

Overview

This feature extends the time slicing functionality to Plan and PlanSegment entities. This enables the user to easily view and update the Plan and PlanSegment for multiple time periods.

OIPA already support time slicing for Class Group and Client relationship entities. With this feature user should also have the ability to enter a change (past, current-dated, or future) from the same UI page where plan or related data is viewed.

In addition this feature will retain the benefits provided by activity processing – undo/redo capability, history/audit trail of changes made.

Prerequisites

User should have access to the 'PlanTimeSliceTest' transaction, BaseCoveragePlus Segment of Group Prototype Product.

User should have created a Group customer to add the PlanScreen

Business Rule

New Items

Screen name: PlanScreen

Company name: Prototype Group Child Company

Transaction name: PlanTimeSliceTest

Company name: Prototype Group Child Company

Product name: Group Prototype Product

Transaction name: PlanScreenUpdate

Company name: Prototype Company

Plan name: Customer Plan

Attached rule: ProcessChangeRequest

Existing Items

Screen name: PolicyScreen

Company name: Prototype Group Child Company

Product name : Group Prototype Product

Segment name: BaseCoveragePlus

Calculate Rule name: CalculateGeneralLifeBaseCov

Company name: Prototype Group Child Company

Product name: Group Prototype Product

Configuration Details

PlanScreen Business Rule:

It is configured at 'Prototype Group Child Company' which has a 'MaximumFaceAmount' field configured in the screen and will be spawning 'PlanScreenUpdate' transaction which will pass the PlanName to the transaction.

When Plan is added to the Agreement, the plan Table which has been modified based on the Plan

Time slice requirement shows the below observation:

The grid only displays Plan entities that have a non-shadowed record status by default, only three columns i.e Plan Name, Product Name and Company Name columns will be displayed on the Plan Entity Grid. Rest all other column will be removed:

The Plans table section will also support right-click menu navigation options.

'Go To Agreement' navigates the user to the Agreement Screen/Plans tab and passes the Plan identifier and Plan Effective Date values to pre-select the targeted Plan Agreement record.

Plan Time slice grid

The grid is always influenced by the selection of the Plan entity and the value of the As Of Date. which will display time slices for the Plan entity selected.

The grid by default displays non-shadowed time slices ordered by descending effective dates.

The system auto-selects the active time slice where the As Of Date is between a time slice's effective and expiration dates. If there is no active time slice for given As Of Date then system will auto selects the first slice in plan time slice grid.

Any row of the grid may be selected by the user to display the Plan time slice's detailed field data.

On Selection of Plan Time Slice it will display two tab Plan Details and Plan Segments with the details pertaining to the selected time slice.

Plan Details

If it does not have ANY Plan history (i.e 'time slice') records in 'Active' or

'Pending' status and only the initial record exists for the Plan and it is still in 'Draft' status then if such time slice selected, all the details available to edit except EffectiveTo (Expiration Date), Company and Product Name

If it has a Plan history (i.e 'time slice') records in 'Active' or 'Pending' status and selected time slice is in 'Draft' status then none of the plan fixed fields except Status (Will be mapped to AsPlan.BusinessStatusCode) will be available for editing. All the Dynamic field will be available for editing.

Plan Segments

If it does have ANY Plan history (i.e 'time slice') records in 'Active' or 'Pending' status and selected time slice is in 'Draft' status then if selected segment belongs to such time slice then , none of the PlanSegmentName fixed field will be available for editing. All the Dynamic field will be available for editing.

In selected plan time slice is in 'Draft' status, 'Add' And 'Delete' option will be available. Current functionality will be applicable for a new plan segment addition/Deletion.

In all other cases 'Add' and 'Delete' option will not be available. Right click on Plan Segment will only have 'Delete' (If it met other required condition) option. 'Edit' option will be removed.

If the status of the time slice is 'Pending' or 'Active', all the details are non-editable in both the tabs –Plan Details, Plan Segments. Further, in Plan Segments tabs – 'Add' and 'Delete' options will not be available for the user.

Add First Plan Time Slice

System will continue to support the new plan addition from the Agreement Plan Screen.

When a new plan will be added, System will treat it as a first slice for that plan entity and create a new plan time slice in 'Draft' status and

EffectiveTo field will be disabled.

PlanSegmentName should be unique within a Plan Time Slice. If duplicate then system should throw an error 'PlanSegmentName is duplicate.'

'Add' Plan Time Slice

When a user clicks on the 'Add' button, a pop-up screen displays an option for the user to enter the Effective Date

The Add button is not visible until there is at least one Active (record status) time slice for the selected Plan Entity

Add New Time Slice Pop-up opens and user may use the date picker or type the date in which the new time slice will be effective

Error is thrown when there is an invalid effective date

If the user is creating a time slice where the effective date is equal to the effective date on an existing Active time slice then it shows an exception 'Invalid Effective Date for new Plan Time Value Record'

If Cancel button is selected it closes the pop-up and returns the user to the Plan screen as it was left.

If Next button is selected it closes the pop-up and returns the user to the Plan screen

No time slice will be selected in the Plan Time Slice Grid if one had been selected prior to pressing the Add button

No new time slice will be added until the save or submit buttons have been pressed

The system displays the two tabs 'Plan Details' and 'Plan Segments'

The Plan Detail screen is enabled and Save and Cancel buttons are visible

Effective Date is populated with the effective date entered on the pop-up

All fixed and dynamic fields are pre-populated as follows except for the Expiration Date, Record Status and Business Status fields

If the new time slice predates all existing time slices for the Plan, the initial detail field data is defaulted from the oldest existing time slice with an Active record status

Otherwise, the initial detail field data is defaulted from the immediately preceding time slice with an active record status

All dynamic fields are enabled and may be modified unless otherwise designated by configuration

None of the plan fixed fields except Business Status field will be editable.

Effective From is pre-populated and the Effective To will be set up in an activity when the slice becomes Active

Once the user clicks 'Save' after editing the 'Plan Details 'tab values, a new time slice is created in 'Draft' status(Record Status).

Plan Time Slice Differences

Left most column of Plan Time Slice table which will contain check boxes. These check boxes are used to select the Time Slices that will be compared for differences.

Once the user selects two Time Slice check boxes have been selected the check boxes on the other time slice rows will be disabled and the 'Show Diff' button will be enabled

The user will select the 'Show Diff' button and the Difference Report pop up will open and display the following sections in the pop up screen:

It will have two tabs 'Plan Details' and 'Plan Segments'

Plan Details: This tab will show the differences between the Plan detail fields of the two time slices.

The column headings for the differences would show the effective from and effective to dates of the two time slices along with the Status of the time slice in bracket. Irrespective of the order of selecting the time slices, the differences would always show the time slice pertaining to an earlier time interval in the left and the later time slice in the right.

Plan Segments: This will show a comparison of the available 'Segment' between the two time slices.

This will be a comparison of the tree structure as two images side by side. The trees will reflect the available segment for the given plan.

It will have a bottom tabular section . That section is blank until the user selects the Segments from Plan Segment Section to be compared.

Once a user selects a Segment from one tree structure, the system automatically selects the same name Segment from the other tree structure, if such a Segment exists. If such a Segment exists, the system displays the bottom tabular section showing the differences between the two segment. Else, the system displays a message requesting user to select a segment which exists in both the tree structures.

Plan Time Slice Test

PlanTimeSliceTest transaction is configured to access the Plan data and PlanSegment data through PlanLoop and PlanSegmentLoop respectively.

PlanTimeSliceTest transaction has the following fields configured:

PlanSegmentName : Fetches the Plan Segment name by using PlanSegmentLoop

PlanEffectiveFrom : Fetches the Effective date of the Plan by using the PlanLoop

PlanName : Fetches the name of the Plan by using the PlanLoop. And also these values are captured in the mathvariables.

When the transaction is processed successfully PlanSegmentName, PlanEffectiveFrom, PlanName fields will be populated on submit and also it is visible in the math section.

BaseCoveragePlus and CalculateGeneralLifeBaseCov :

Existing BaseCoveragePlus Segment is used for prototype configuration. In addition to the existing fields following fields have been configured in the BaseCoverageplus Segment:

PlanNameFromSegment : Fetches the Name of the plan from the segment itself by using PlanLoop

PlanEffectiveDateUpdatedFromSegment : Fetches the Effective date of the plan from the segment itself by using PlanLoop

PlanSegmentNameUpdatedFromSegment : Fetches the segment Name of the plan from the segment itself by using PlanSegmentLoop

PlanNameFromCalculateGeneral : updated by the CalculateGeneralLifeBaseCov which fetches the Plan name by using PlanLoop

PlanEffectiveDateUpdatedFromCalculateGeneral : updated by the CalculateGeneralLifeBaseCov which fetches the Plan Effective date by

using PlanLoop

PlanSegmentNameUpdatedFromCalculateGeneral : updated by the CalculateGeneralLifeBaseCov which fetches the Segment name by using PlanSegmentLoop

When the BaseCoveragePlus Segment is added PlanNameFromSegment, PlanEffectiveDateUpdatedFromSegment and PlanSegmentNameUpdatedFromSegment will be populated from segment on submit where as the fields PlanNameFromCalculateGeneral, PlanEffectiveDateUpdatedFromCalculateGeneral and PlanSegmentNameUpdatedFromCalculateGeneral will be populated from the CalculateGeneralLifeBaseCov of the segment.

PolicyScreen Business Rule:

Existing PolicyScreen which is at Group Prototype Product is used for prototype configuration. In addition to the existing fields following fields have been configured :

PlanEffectiveDate : Fetches the Effective date of the plan by using PlanLoop.

PlanSegmentName : Fetches the Segment Name by using PlanSegmentLoop.

When the Policy of Disability product is added PlanSegmentName and PlanEffectiveDate will be populated on submit of the policy.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Activity Requirement Screen Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Activity Requirement Screen Prototype

This prototype was configured to demonstrate the recently renamed ActivityRequirementScreen (formerly RequirementScreen) business rule.

Prototype Prerequisites

- OIPA user credentials for the Functional Prototype Plan

Prototype Configuration

- The ContractRenew transaction was created to demonstrate that the ActivityRequirementScreen business rule functions exactly as it did before it was renamed. In the Main Explorer window, navigate to **Companies | Prototype | Products (if Products are enabled) | Individual Prototype Product (if Products are enabled) | Plans | Functional Prototype Plan | Transactions | Policy Transaction | ContractRenew** to view the transaction configuration.
- The DeliveryRequirements business rule has been attached to the ContractRenew transaction to facilitate requirement processing, as is typical for transaction requirement configuration. In the Main Explorer window, navigate to **Companies | Prototype | Products (if Products are enabled) | Individual Prototype Product (if Products are enabled) | Plans | Functional Prototype Plan | Transactions | Policy Transaction | ContractRenew | Attached Rules | DeliveryRequirements** to view the business rule configuration. The key configuration for this business rule is explained below. This is previously existing functionality.
 - The OpenDate and CloseDate fields of the ContractRenew activity are both required to have values for the activity to process
- The GeneratePendingRequirements business rule has been attached to the ContractRenew transaction to facilitate requirement processing, as is typical for transaction requirement configuration. In the Main Explorer window, navigate to **Companies | Prototype | Products (if Products are enabled) | Individual Prototype Product (if Products are enabled) | Plans | Functional Prototype Plan | Transactions | Policy Transaction | ContractRenew | Attached Rules | GeneratePendingRequirements** to view the business rule configuration. The key configuration for this business rule is explained below. This is previously existing functionality.
 - The PlanGUID of the plan to which the transaction belongs must match that defined in the AsRequirementDefinition database table
 - The values of the RenewalDate and DueDate fields will be

copied from the RenewalDateMV and DueDateMV
MathVariables, respectively.

View Prototype in OIPA

1. Log in to a Rules Palette environment.
2. In the Global Rules Explorer window, navigate to **Business Rules | Screen | ActivityRequirementScreen**.
3. Open the ActivityRequirementScreen XML file.
4. Note that the parent tag of the rule has been changed from <RequirementScreen> to <ActivityRequirementScreen>.
5. Log in to OIPA using credentials for a user belonging to the Functional Prototype Plan.
6. Open a policy or create a shell policy.
7. Add the ContractRenew activity.
8. Process the ContractRenew activity.
9. On the Activity screen, click the icon to the left of the activity name to open the Activity Requirement screen.
10. Enter a date in the **Close Date** field. Doing so will satisfy the requirement specified in the DeliveryRequirements business rule attached to the activity.
11. Click **OK**.
12. Process the activity. It should process as is expected.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Agreement and Agreement Role Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Agreement and Agreement Role Prototype

OIPA now supports the ability to create Agreements various categories and types of Agreements and relate various agreement roles to these Agreements. Refer to [Agreement Overview](#) section and [Agreement Roles](#) sections for the various business rules to be configured for handling these functionalities.

Prototype Explanation

A set of Agreement definitions are created in OIPA and the prerequisite AsCode details are set up to ensure further Agreement types can be created from Rules Palette. For the existing Agreement definitions, we can create Agreements and Agreement Roles in the OIPA application.

View Prototype in OIPA

1. Login to OIPA with a Prototype company user credentials.
2. Search for a Group Customer through the Customer -> Search menu option.
3. Select a Group Customer and navigate to the Agreements link.
4. The various Agreement Categories defined in the AsCode with codename 'AsCodeAgreementCategory'.
5. On selecting one of the categories, the Agreement tree structure of all agreements created under this category is displayed. Further new Agreements can be created or existing agreements can be edited.
6. Under each agreement on which Agreement roles are configured to be allowed, there is a 'Roles' tab next to the Agreement Details tab. Selecting the Agreement Roles tab provides the list of existing roles and allows adding further roles through the "Add" button which provides the option to use a new client or existing client or group customer.
7. Use the various menu options and the tabs in the screen to play around with the functionality created in prototype. Also, refer to the [Agreement Products prototype](#) and [Agreement Class Groups Prototype](#) for further functionality.
8. Log in to the Rules Palette.
9. In the Global Rules Explorer window, navigate to **Agreements**.
10. View the various Agreement definitions configured.
11. Use the "Add New" option available on the right click menu on Agreements node to create further Agreement types.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Agreement Class Groups Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Agreement Class Groups Prototype

OIPA now supports the ability to use Agreements in order to create and link one or more Class Groups with a Group Customer.

Prototype Explanation

The existing global agreement business rule for agreement definition 'Master Agreement - Insured' and 'Bank Letter' are configured such that it can be related to one or more Class Groups. Therefore, when a user creates an agreement of type Master Agreement - Insured or Bank Letter, he/she will be able to create Class Groups with this agreement.

Instead, when the user selects agreement type 'Funding Letter' or 'Funding Account', while creating a new agreement, he/she won't be able to relate this agreement to class groups.

Business Rules

- **Agreement:** A new Agreement of type 'Master Agreement - Insured' and 'Bank Letter' have been created in the Agreements node in Global Rules Explorer. The 'Master Agreement - Insured' is created under the AgreementCategoryCode 'CNTR' (Contract) while the 'Bank Letter' is under the category 'BL' (Bank Letter).

View Prototype in OIPA

1. Login to OIPA with a Prototype company user credentials.
2. Search for a Group Customer through the Customer -> Search menu option.
3. Select a Group Customer and navigate to the Agreements link.
4. Select 'Contract' category agreements and create a new agreement of type 'Master Agreement - Insured' or select an existing agreement of type 'Master Agreement - Insured'. Or similarly select 'Bank Letter' category and select or create an agreement of type 'Bank Letter'
5. Select the Class Groups tab. This lists the various class groups already linked under the agreement and allows linking further class groups or editing linked class groups and deleting linkages.
6. Perform the various actions available to understand the functionality.
7. Log in to the Rules Palette.
8. In the Global Rules Explorer window, navigate to **Agreements**.
9. View the XML configured for the 'Master Agreement - Insured' and notice that the CLASSGROUP attribute is set to 'Yes'. Similarly view the 'Bank letter' agreement configuration.
10. Select on of the other Agreement types and view the CLASSGROUP attribute to understand the configuration set up needed to allow/disallow class group linkage under a specific agreement type.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Agreement Products Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Agreement Products Prototype

OIPA now supports the ability to create Agreements in order to link a Group Customer's purchased Product(s) together to create and support the customer's plan(s). The "Master Agreement - Insured" type Agreement is used to demonstrate the feature in OIPA.

Prototype Explanation

The existing global agreement business rule for agreement definition 'Master Agreement - Insured' is configured such that it can be related to Product. Therefore, when a user creates an agreement of type Master Agreement - Insured, he/she will be able to associate a product with this agreement. From the product pop up selector, a user can select the product which he/she wants to associate with the agreement. Once the product is linked to the agreement, the details are displayed under the 'Product Agreement Details' section.

Instead, when the user selects agreement type 'Bank Letter', while creating a new agreement, he/she won't be able to relate this agreement to a product. Similarly, as the PRODUCT attribute is not specified for other agreement types such as Master Agreement-ASA, Funding Account, etc, these agreements also cannot be linked to a product.

Business Rules

- **Agreement:** A new Agreement of type 'Master Agreement - Insured' has been created in the Agreements node in Global Rules Explorer. The 'Master Agreement - Insured' is created under the AgreementCategoryCode 'CNTR' (Contract).

View Prototype in OIPA

1. Login to OIPA with a Prototype company user credentials.
2. Search for a Group Customer through the Customer -> Search menu option.
3. Select a Group Customer and navigate to the Agreements link.
4. Select 'Contract' category agreements and create a new agreement of type 'Master Agreement - Insured' or select an existing agreement of type 'Master Agreement - Insured'.
5. Select the Products tab. This lists the set of Products linked with this agreement and also allows for further products to be linked with the Agreement. There is an option to also delete Products which are linked already (so long as there are no related AsPlan records created under this Product for the current selected Agreement).
6. Log in to the Rules Palette.
7. In the Global Rules Explorer window, navigate to **Agreements**.
8. View the XML configured for the 'Master Agreement - Insured' and notice that the PRODUCT attribute is set to 'Yes'.
9. Select on of the other Agreement types and view the PRODUCT attribute to understand the configuration set up needed to allow/disallow plan linkage under a specific agreement type.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Assessment Screen
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Assessment Screen Prototype

The Assessment screen, previously called the Impairment screen, has been enhanced with several additional features. Because the Assessment screen is completely code-driven, no configuration is necessary. Only security setup is required for the screen.

Prototype Prerequisites

- An application for a Case should exist with impairment records.

Prototype Configuration

- For the Super security group, in the NBU Term Life Plan of the Alamere Individual Company, security is set for the AllowImpairmentDelete button on the PolicyAssessment Plan Page. In the Admin Explorer window, navigate to **Security | Application Security | Security Groups | Super | Plan Security | Alamere Individual Company | NBU Term Life Plan | Plan Pages | PolicyAssessment** to view the security settings. No other security groups are provided access to the AllowImpairmentDelete button on this Plan Page.

View Prototype in OIPA

1. Log in to OIPA using credentials belonging to the Super security group of the Alamere Individual Company.
2. Open an application record and select **Assessments** from the menu on the left side of the screen.
3. Note that the Show Shadows checkbox is visible and available for selection, and that the trash can icon displays for active Impairment records.
4. Select the trash can icon for one of the Impairment records.
5. Note that the selected Impairment record is deleted.
6. Check the **Show Shadows** checkbox.
7. Note that the shadowed Impairment record now displays on the screen.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Classes and Class Groups Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Classes and Class Groups Prototype

The population of clients associated to a Group Customer may be classified into Class hierarchies which are organized into Class Groups. It is this set of hierarchies that determines the benefits that may be offered to participants. This prototype will demonstrate the set-up of the Class Groups screen and Class screens.

Prototype Explanation

This prototype contains configuration of the ClassGroupScreen, ClassScreen and ClassGroupUpdate transaction.

The ClassGroupScreen allows a user to establish and maintain the class groups and classes for a Group Customer. It also presents the time valued detail history and allows navigation to the contained classes. This prototype's configuration is contained in the global version of the rule.

The <UseBusinessStatus> is configured as 'Yes', which indicates that the business status should be used. The combo for Status can be seen on the ClassGroup Screen which is populated using the code values from AsCode for code name 'AsCodeClassGroupBusinessStatus'.

The <DisplayTab> elements is configured to show the following tabs in the Class Group Screen by setting the value as "Yes" for the attributes ShowClassGroupDetails and ShowClasses.

Another attribute "ClassNodes" set as "Expanded" indicates the class nodes within the tab "Classes" are expanded.

The data that will be returned in the Summary Grid once a specific class group is selected from the list of available class groups from the ClassGroupScreen. This section is configured specifically for

each class group type.

The <Spawns> element is configured specifically for each ClassGroupType. The group customer level activity “ClassGroupUpdate” is spawned at the ‘EffectiveFrom’ date when the ‘Submit’ action is executed on the class group.

Once the class group is saved, the class group is added in ‘Draft’ status. On executing the ‘Submit’ action, the class group status is changed to ‘Pending’.

ClassGroupType – “03” (Banking) action and events are:

OnLoad: This event triggers on the load of the Class Group Screen when the Banking type has been selected. The following warning message is displayed: “Class Group Screen is loaded.”

OnChange: This event triggers when the field value of ClassGroupName is changed. The following error message is displayed: “Class Group Name is changed.”

OnSubmit: This event validates the field value of EffectiveFrom. When empty of value the following message is displayed: “Effective Date is a required field.”

The ClassScreen presents the dynamic data stored with the class. This prototype’s configuration is contained in the Prototype company override.

The ClassGroupUpdate transaction is of type ScreenUpdate. It is spawned by ClassGroupScreen when the class group is saved. The transaction’s configuration is contained in the Prototype company’s Customer Plan override.

The value for the transaction field ‘ClassGroupName’ is passed

from the ClassGroupScreen business rule to this transaction as a spawn field.

This transaction can be processed either by system or by the user. When this transaction is processed, the Class Group status turns to 'Active'.

ProcessChangeRequests is an attached rule and is key to changing the class group status.

To execute the screens in OIPA, the following prerequisites must be satisfied:

1. A Group Customer record should exist in the Prototype company.
2. Agreements should exist for the Group Customer and in a state where it may be associated to new Class Groups.

View Prototype in OIPA

1. Log in to the Rules Palette.
2. In the Global Rules Explorer window, navigate to Transaction.
3. View the ClassGroupUpdate transaction.
4. In the Global Rules Explorer window, navigate to Business Rules -> Screen.
5. View the ClassGroupScreen and ClassScreen rules.
6. Login to OIPA with a Prototype company user credentials.
7. Search for a Group Customer per the prerequisites listed above through the Customer -> Search menu option.
8. Select the Group Customer and navigate Agreements link.
9. Select an Agreement type and a specific agreement instance that is ready to accept new Class Groups.
10. Select the Class Groups tab below the instance tree.
11. Press the Add button and the Class Group type to add to the

Agreement.

12. Enter and save the class group data.

13. The new class group will now be visible within the Class Groups navigation link.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Client and Policy Search Enhancement Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Client and Policy Search Enhancement Prototype

The results of searches on the Client Search screen and Policy Search screen can now be sorted by configurable criteria.

Prototype Configuration

- The PolicySearchScreen business rule was configured to sort search results based on the values of three fields: PolicyNumber, StatusCode and IssueStateCode. In the Global Rules Explorer, navigate to **Business Rules | Screen Rules | PolicySearchScreen | Plan Overrides | PolicySearchScreen (Functional Prototype Plan)** to view the configuration, the key aspects of which are explained below.
 - The <OrderBy> element is repeated three times in order to use the values of the following three fields in the sorting of the search results:
 - PolicyNumber
 - StatusCode
 - IssueStateCode
 - PolicyNumber has an ordinal value of 1, meaning it is the main criterion used in the sort operation. The results will be sorted in ascending order with respect to PolicyNumber.
 - StatusCode has an ordinal value of 2, meaning it is the secondary criterion used in the sort operation. The results will be sorted in ascending order with respect to StatusCode.
 - IssueStateCode has an ordinal value of 3, meaning it is the tertiary criterion used in the sort operation. The results will be sorted in descending order with respect to IssueStateCode.
 - The IssueStateCode column is configured to be hidden in the search results table, but is still used as a criterion in the sorting operation.
- The ClientSearchScreen business rule was also configured to sort search results based on the values of three fields: LastName, AddressLine1 and AddressLine2. In the Global Rules Explorer, navigate to **Business Rules | Screen Rules | ClientSearchScreen | Company Overrides | ClientSearchScreen (Prototype Company)** to view the configuration, the key aspects of which are explained below.
 - The <OrderBy> element is repeated twice in order to use the

following data in the sorting of the search results:

- LastName
 - Address, a composite column composed of the AddressLine1 and AddressLine2 database columns
- LastName has an ordinal value of 1, meaning it is the main criterion used in the sort operation. Because no value was specified for the ORDER attribute in the configuration, the search results will by default be sorted in ascending order with respect to LastName.
 - AddressLine1 has an ordinal value of 2, meaning it is the secondary criterion used in the sort operation. The results will be sorted in descending order with respect to AddressLine1.
 - AddressLine2 has an ordinal value of 3, meaning it is the tertiary criterion used in the sort operation. The results will be sorted in descending order with respect to AddressLine2.

View Prototype in OIPA

1. Log into OIPA using credentials for a user assigned to the Prototype Company.
2. Navigate to the Policy Search screen.
3. In the **Company Name** field, select **Prototype Individual Child Company**.
4. If Products are enabled, in the **Product** field, select **Individual Prototype Product**.
5. In the **Plan** field, select **Functional Prototype Plan**.
6. Enter some search criteria and select **Find**.
7. Note that the search results are sorted as described in the Prototype Configuration section above.
8. Navigate to the Client Search screen.
9. Enter some search criteria and click **Find**.
10. Note that the search results are sorted as described in the Prototype Configuration section above.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Comment Enhancement Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Comment Enhancement Prototype

Commenting functionality has been enhanced to allow comments to be added to requirement and impairment records. This functionality is dependent on security configuration, which controls if a user can view, add, edit or delete comments, as well as if a user can view a requirement or impairment record's comment history. The new functionality is also dependent on several configurable business rules, as is described below.

Prototype Prerequisites

- An OIPA user associated to the Super security group must exist.

Prototype Configuration

- The global CommentsScreen business rule contains the <RequirementComments> and <ImpairmentComments> elements, which define the fields that display when the Comments screen is accessed via the Policy Requirement screen and the Assessment screen, respectively. In the Global Rules Explorer window, navigate to **Business Rules | Screen | CommentsScreen** to view the configuration. The key configuration for this business rule is explained below.
 - The following fields are configured within the <RequirementComments> element, and will therefore display on the Comments screen when accessed via the Policy Requirement screen:
 - Functional Department
 - Category
 - Requirement Comment Field
 - The following fields are configured within the <ImpairmentComments> element, and will therefore display on the Comments screen when accessed via the Assessment screen:
 - Functional Department
 - Category
 - Impairment Comment Field
 - The <UseTemplates> element has a value of Yes in both the <RequirementComments> and <ImpairmentComments> sections. This means that the Comments screen for both requirements and impairments will use [comment templates](#).
- The global CommentsSearchScreen business rule contains the <RequirementComments> and <ImpairmentComments> elements, which define the fields that display when the Comments Search screen is accessed via the Policy Requirement screen and the Assessment screen, respectively. In the Global Rules Explorer window, navigate to **Business Rules | Screen | CommentsSearchScreen** to view the configuration. The key

configuration for this business rule is explained below.

- The following fields are configured within the <RequirementComments> element, and will therefore display in both the Comment Search and Comment Search Result sections of the Comments Search screen when accessed via the Policy Requirement screen:
 - Requirement Comment Field
- The following fields are configured within the <ImpairmentComments> element, and will therefore display in both the Comment Search and Comment Search Result section of the Comments Search screen when accessed via the Assessment screen:
 - Impairment Comment Field
- Button security has been set on two new [Plan Pages](#): [RequirementComments](#) and [ImpairmentComments](#). To view the security configuration, navigate in the Admin Explorer window to **Security | Application Security | Security Groups | Super | Plan Security**, choose a company and navigate to the **RequirementComments** and **ImpairmentComments** Plan Pages. The security for these Plan Pages is described below.
 - The following buttons have security set in the RequirementComments and ImpairmentComments Plan Pages:
 - Add
 - View
 - Update
 - Delete
 - View History
 - The following actions are allowed based on the security for the buttons above:
 - AddComment
 - DeleteAllComments
 - DeleteUserCommentsOnly
 - UpdateAllComments

- UpdateUserCommentsOnly
 - ViewAllComments
 - ViewAllCommentsHistory
 - ViewUserCommentsHistoryOnly
- The PolicyRequirement Plan Page has security set for two new buttons. In the Admin Explorer window, navigate to Plan Pages for the security group Super to view the security configuration. The new security options for this Plan Page are described below:
 - Two new buttons are able to have security set:
 - RequirementComments
 - PolicyComments
 - The PolicyAssessment Plan Page has security set for two new buttons. In the Admin Explorer window, navigate to Plan Pages for the security group Super to view the security configuration. The new security options for this Plan Page are described below:
 - Two new buttons are able to have security set:
 - ImpairmentComments
 - PolicyComments

View Prototype in OIPA

1. Log in to the Rules Palette and click on the **Admin Explorer** tab.
2. Navigate to **Security | Application Security | Security Groups | Super | Plan Security** and continue to navigate to a company of your choosing.
3. Expand the **Plan Pages** node and check out the **ImpairmentComments** Plan Page.
4. Configure the security settings for the available buttons and check the Plan Page back in.
5. Repeat steps 3 and 4 for the RequirementComments Plan Page.
6. Log in to OIPA with user credentials belonging to the same company as the Super security group used above.
7. Navigate to the Assessment screen. Note that all of the actions and buttons that were enabled are able to be used.
8. Repeat step 7 for the Requirement screen.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Data type IDENTIFIER prototype with Group related screens

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Data type IDENTIFIER prototype with Group related screens

OIPA now supports the ability to use DATATYPE="IDENTIFIER" fields in the following Group related screen Business Rules:

AgreementRelationship, AgreementRoleScreen, AgreementScreen, AlternateNameScreen, ClassDetailScreen, ClassGroupScreen, ClassPlanScreen, ClassSegmentNameScreen, EnrollmentScreen, ClientRelationshipScreen, GroupCustomerScreen, IntakeProfileScreen, PhoneScreen. At a transaction level, the group related transactions at group customer level will also provide support to DATATYPE="IDENTIFIER".

Prototype Explanation

The following business rules will be used to demonstrate this new functionality.

Business Rules

- **'ClientRelationshipScreen' BR:** We have added the configuration to showcase the support of 'IDENTIFIER' Screen MathVariable Type.
- **'AgreementRoleScreen' BR:** We have added the configuration to showcase the support of 'IDENTIFIER' Field DataType.
- **'AddAgreementRole' Transaction :** We have added the configuration to showcase the support of 'IDENTIFIER' MathVariable Type.

View Prototype in OIPA

1. To view the Identifier configuration in the ClientRelationshipScreen BR:
 1. Log in to OIPA with credentials for a user belonging to the 'Prototype Super' security group.
 2. From the top menu, select Customer -> Search. Search and select one Group Customer record.
 3. Click on the Relationship navigation option on the left navigation menu.
 4. Select the 'CONTACT' primary relationship. Click Add Relationship. Select 'ACCTMGR' as the Secondary Relationship.
 5. Select an applicable for the 'Identifier Number Generation Based On' combo field. The 'IdentifierNumber' field will get populated with the appropriate identifier value. The same will get recorded along with the relationship.
2. To view the Identifier configuration in the AgreementRoleScreen BR:
 1. Log in to OIPA with credentials for a user belonging to the 'Prototype Super' security group.
 2. From the top menu, select Customer -> Search. Search and select one Group Customer record.
 3. Click on the Agreement navigation option on the left navigation menu. Select any one Agreement record.
 4. Navigate to the 'Roles' tab of 'Agreement' screen.
 5. Add or access the agreement role record details and then click on 'Save' button.
 6. The 'Identifier Number' field is filled with the unique number which is a concatenation of static value 'ARN' and 8 digit sequence number (Ex: ARN00000001).
3. To view the Identifier configuration in the AddAgreementRoletransaction:
 1. Log in to OIPA with credentials for a user belonging to the 'Prototype Super' security group.
 2. From the top menu, select Customer -> Search. Search and select one Group Customer record.
 3. Click on the Activities navigation option on the left navigation menu.

4. Add and process an 'AddAgreementRole' activity.
5. For the newly created agreement role, 'Identifier Number' field data is filled with the 'IdentifierNumber' math variable data from the transaction.
6. Recycle the activity. The originally generated Identifier number will be fetched and override the newly generated identifier number.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Disbursement Approval Threshold Limits Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Disbursement Approval Threshold Limits Prototype

OIPA now supports the ability to configure threshold limits for each user group for Disbursement Approval. Using this, feature, limits can be set up for all user profiles using the DisbursementAmount field or any other decimal/integer/money field identified in the disbursement search results in the disbursement approval screen.

Prototype Explanation

There is no configuration changes needed for this page. Existing DisbursementApprovalScreen configuration can be used and security can be set up to explain this functionality. Following three security profiles have been created in the prototype configurations to allow demonstrating this feature.

Security Profiles

- **'Prototype Analyst' Security user:** For this user, the ApprovalLimits authorization button is not checked and so the disbursement approval will be allowed without any limits and the user can approve all disbursement records irrespective of threshold limits.
- **'Prototype Tester' Security user:** For this user, the ApprovalLimits authorization button is checked and the threshold pane is used to set up a minimum threshold of 10,000USD and maximum threshold of 15,000 USD against the DisbursementAmount field. This user can approval all disbursement records where the disbursement amount is between 10,000USD and 15,000USD.
- **'Prototype Super' Security user:** For this user, the ApprovalLimits authorization button is checked and the threshold pane is used to set up a minimum threshold of 0USD and maximum threshold of 100,000 USD against the DisbursementAmount field. This user can approval all disbursement records where the disbursement amount is between 0USD and 100,000USD.

View Prototype in OIPA

1. Login to OIPA with a Prototype Analyst security group user credentials.
2. Navigate to the Disbursement Approval Screen.
3. All the disbursement records will have the 'Approval Status' and 'Disapprove Reason' fields enabled for edit, thus allowing the user to approve all disbursement records irrespective of threshold limits.
4. Login to OIPA with a Prototype Tester security group user credentials and navigate to the Disbursement Approval screen.
5. Disbursement records for all records with DisbursementAmount between 10,000USD and 15,000USD will have the 'Approval Status' and 'Disapprove Reason' fields enabled for edit while for all other disbursement records, these two fields will be disabled.
6. Login to OIPA with a Prototype Tester security group user credentials and navigate to the Disbursement Approval screen to view the same threshold limit functionality.
7. Log in to the Rules Palette.
8. In the Admin Explorer window, navigate to Security and select a security group.
9. Select Company Pages and under that, select the DisbursementApproval page node.
10. Provide authorization to the ApplyApprovalLimits authorization button. The threshold limit page is available for edit and the application displays one record on which the user is allowed to select the field name and enter required threshold limit information. Currently only one field and one range can be specified under DisbursementApproval page.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Enrollment
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Enrollment Prototype

Employees/Members of a specific group customer class can be enrolled to the benefits offered by a Group Customer. OIPA allows for evaluating the eligibility of a specific member and enrolling the employee/member to appropriate benefits. The following prototype transactions and business rules are created to demonstrate the ability to enroll employees/members into benefits.

Prototype Explanation

The following transactions – EvaluateMembership, EnrollmentPrototype and Activate - have been created in the Prototype company.

The EvaluateMembership transaction will evaluate the membership rules defined for the class and determine the benefits eligible. The transaction is defined as a

Group Prototype Product override.

The Enrollment Prototype transaction is the Enrollment type transaction which will be presented as part of the Enrollment screen when specific benefit coverage(s) are selected for enrollment. The process of enrolling creates the enrollment record. The Enrollment Prototype transaction creates a policy with the selected coverages. The client upon whom this executes takes the roles of Eligible Member, Insured and Owner on the new policy and Participant-Member on each coverage. The transaction Activate will be added as an activity to the new policy. The transaction is defined as a Group Prototype Product override.

The Activate transaction updates a policy, to which it is added, with an activation date and changes the policy status to active. The transaction is defined as a Group Prototype Product Override

To execute enrollment, the following prerequisites must be satisfied:

1. A Group Customer record should exist in the Prototype company.
2. An agreement should exist under the Group Customer that establishes an association of the Group Prototype Product and the segment definitions BaseCoveragePlus and BaseCoverageBasic

3. A Class Group should be linked under the Agreement in which one or more classes of type Eligibility should be available,. Using the Plans and Segments created under the Group Customer, one or more Class Sub-plan records should be created.

4. An Employment type of relationship needs to be established between the Group Customer and 1 or more of the Group Customer's client population Prior to running eligibility.

View Prototype in OIPA

1. Log in to the Rules Palette.
2. In the Global Rules Explorer window, navigate to **Transaction**.
3. View the EvaluateMembership and EnrollmentPrototype transactions.
4. In the Admin Explorer window, select the Enrollment Transaction node. The existing enrollment transaction records can be viewed.
5. Expand navigation to the EnrollmentPrototype transaction and right click to "View Details" or "View Enrollment Transaction".
6. Login to OIPA with a Prototype company user credentials.
7. Search for the Group Customer, per the prerequisites listed above through the Customer -> Search menu option.
8. Select the Group Customer and navigate to the Client Relationship link.
9. Go to the Individual Client by selecting "Edit Client" right click menu option under relationships and click "Add Activity".
10. Add and process the EvaluateMembership activity at the client level.
11. Now select the Enrollment screen from the left navigation options under the Client. The same Enrollment screen can be accessed by using right click context menus in Client Relationship Screen as well.
12. All the coverages that are eligible for the client are available for selection. Select one or more of the coverages and proceed to enter details for the EnrollmentPrototype transaction (presented as part of the enrollment screen) and press "Enroll" or "Save" button.
13. On Click of Enroll, the client is enrolled for the coverage(s) a policy is created and the PolicyName can be found in the transaction math of the processed EnrollmentPrototype Transaction).
14. On Click of Save a pending EnrollmentPrototype transaction will be present in the Client Activities Screen.

15. Once a client is enrolled he is not allowed to enroll again.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Extend Create Policy Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Extend Create Policy Prototype

OIPA now has the ability to optionally insert pending activities or programs on a new policy as it is created with the Create Policy APE. New Configuration is added to the Create Policy APE on the Create Policy transaction in the Functional Prototype Plan to Illustrate this functionality

Prototype Explanation

When a new policy is inserted using the CreatePolicy APE, configuration should support simultaneously inserting pending activities into the new policy similar to a Spawn section of a transaction. The activities should be conditionally created based on evaluation of a test criterion.

Likewise, when a new policy is inserted using an the CreatePolicy APE, configuration should support simultaneously inserting segment or policy level Programs to the new policy similar to how the segments are built in the CreateSegments APE. The Programs should also be conditionally created based on evaluation of a test criterion.

Configuration Overview

Configuration is provided to

- Add a new Activities section to CreatePolicy APE to insert a StatusChange activity on the new policy
- Update the CreatePolicyUsingNewValues xml to calculate a new status value to pass to the StatusChange activity.

View Prototype in OIPA

1. Login to OIPA with a Prototype company user credentials.
2. Select a Functional Prototype Plan
3. Create a new Policy
4. Add Activity and process the activity having the create policy APE
5. The new policy record is created and pending activities get inserted on the new policy

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Illustration
Processing Using AsFile Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Illustration Processing Using AsFile Prototype

OIPA now supports the ability to manage illustrations on an In Force policy through the RequestType="Illustration" setting. There is a set of Illustration related tables that have been created where illustration transaction data will be recorded till the output XML is generated after which records created in these tables will be deleted.

Prototype Explanation

A new AsFile with FileID = "ILS" is configured to demonstrate this functionality. This AsFile uses the Policy-Illustration type transaction "PolicyIllustration" configured under the Functional Prototype Plan.

View Prototype in OIPA

1. Use SOAP protocol and trigger the AsFile with FileID = "ILS". The payload for the SOAP request should contain the required information. For sample, please use the request XML sample provided below.
2. Verify the output Illustration response.
3. Log in to the Rules Palette.
4. In the Admin Explorer window, navigate to File named "Illustration (Prototype)".
5. View the XMLData and XSLT sections to understand the configuration syntax and options.
6. In the Main Explorer, navigate to the Functional Prototype Plan and select the Illustration transaction under policy transactions. View the configurations for reference.

Sample SOAP XML request

The following XML can be used as a sample to trigger the illustration request in OIPA. Please note: only the Body of the SOAP request (including the payload) is provided. The header and main SOAP envelope elements should be added appropriately and the required user credentials (prototype company user) should be provided correctly.

```
<soapenv:Body>
<fil:processFileReceived
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<fileId xsi:type="xsd:string">ILS</fileId>
<xml xsi:type="xsd:string"><![CDATA[<Tests>
<Policy>
<PolicyNumber>TEST1234</PolicyNumber>
<PlanName>Functional Prototype Plan</PlanName>
<CompanyName>Prototype Individual Child Company</CompanyName>
<IssueStateCode>02</IssueStateCode>
<CreationDate>2009-04-06 09:18:45</CreationDate>
<PolicyName>Test123453</PolicyName>
<PlanDate>2009-04-06 09:18:45</PlanDate>
</Policy>
<IllustrationRequest>
<StartDate>2013-04-18 09:18:45</StartDate>
<EndDate>2014-09-18 09:18:45</EndDate>
<StopOption>01</StopOption>
<StopOptionYears>14</StopOptionYears>
</IllustrationRequest>
<IllustrationBasis>
<InterestAssumption>01</InterestAssumption>
<InterestAssumptionRate>13.65</InterestAssumptionRate>
<MortalityAssumption>02</MortalityAssumption>
<MortalityAssumptionBlend>12</MortalityAssumptionBlend>
<DividendAssumption>03</DividendAssumption>
<DividendAssumptionBlend>2014</DividendAssumptionBlend>
<AnchorBasisIndicator>1</AnchorBasisIndicator>
```

```
</IllustrationBasis>
<IllustrationTransaction>
<PrimaryType>02</PrimaryType>
<SecondaryType>11</SecondaryType>
<TertiaryType></TertiaryType>
<Amount>100000</Amount>
<TransactionMode>1</TransactionMode>
<StartDate>2013-04-18 09:18:45</StartDate>
<EndDate>2030-02-09 09:18:45</EndDate>
<IncreasePercent>4.5</IncreasePercent>
<SupplementalSolveType>02</SupplementalSolveType>
<CurrencyCode>IND</CurrencyCode>
</IllustrationTransaction>
<IllustrationReport>
<IllustrationReportType>01</IllustrationReportType>
</IllustrationReport>
<IllustrationVector>
<VectorRequestCode>101</VectorRequestCode>
</IllustrationVector>
<TransEffDate>2014-02-09 09:18:45</TransEffDate>
</Tests>]]></xml>
</fil:processFileReceived>
</soapenv:Body>
```

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Legal Residence
Country Code Configuration Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Legal Residence Country Code Configuration Prototype

OIPA now supports limited ability to configure the LegalResidenceCountryCode field. The LegalResidenceCountryCode element in ClientScreen business rule can be used to change the display name and the order of listing of country codes from AsCountry table.

Prototype Explanation

The LegalResidenceCountryCode element is an optional element and to demonstrate the functionality, the same has been configured in the Client Screen BR of Prototype Company while in the Client Screen BR of International Company, this element is not configured. In the Prototype Company, the display name is changed to 'Legal Residence Country' and the order of country listing is changed to US, CA, JP, ES, GB instead of the alphabetical order of display. In the International Company, the display name will be defined by the translation values set up and the order will be according to the alphabetical order of country names in AsCountry table.

Business Rules

- **ClientScreen:** The existing ClientScreen BR configured at the company override level for Prototype Company and International Company are changed to demonstrate this functionality.

View Prototype in OIPA

1. Login to OIPA with a Prototype Company user credentials.
2. Navigate to the new client creation through Client -> New.
3. See the country code field. This will show the label 'Legal Residence Country' and the country listing in the drop down will be listed in this order - US, CA, JP, ES, GB.
4. Login to OIPA with a International Company user credentials and navigate to Client -> New.
5. See the country code field. This will show the label 'Legal Residence' (dependent on the translation setting) and the country listing in the drop down will be listed in this order - CA, ES, GB, JP, US.
6. Log in to the Rules Palette.
7. In the Global Rules Explorer window, navigate to **Business Rules> Screen> ClientScreen**.
8. View the XML configured for the different Company level overrides to see the configuration example for LegalResidenceCountryCode element..

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > New CopyTo Rules
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

New CopyTo Rules Prototype

This prototype was configured to demonstrate the functionality of several new CopyTo attached business rules.

Prototype Prerequisites

- OIPA user credentials with access to the Prototype Company and the GroupObjectsCopyToBR activity

Prototype Configuration

- The GroupObjectsCopyToBR transaction was overridden for the Customer Plan under the Prototype Company. In the **Global Rules Explorer** window, navigate to **Transactions | GroupObjectsCopyToBR | Plan Overrides | GroupObjectsCopyToBR (Customer Plan)** to view the configuration for this transaction. The key configuration is explained below.
 - The combo field CopyToRule is configured to display a drop-down list of all the CopyTo rules attached to the transaction.
 - The combo field Record is configured to filter its list of values depending on the selection made from the CopyToRule field.
 - An OnChange event is configured to hide the Record field if CopyToGroupCustomerFields is chosen from the CopyToRule field.
 - Based on the value selected from the CopyToRule field, the corresponding attached business rule will execute.
- The GroupObjectsCopyToBR transaction was given the following attached rules, all of which are new CopyTo rules for the 10.0 release. Each business rule's behavior upon being called from the GroupObjectsCopyToBR is also described below.
 - CopyToAgreementFields: Copies data to the fields on the Agreement screen
 - The values available in the Record field will be a concatenation of the name of an agreement available to the group customer and that agreement's current status.
 - When the transaction is processed, the AgreementName field on the Agreement screen will be updated with a value that is the current agreement name concatenated with the text "UpdatedbyCopyTo."
 - CopyToAgreementRoleFields: Copies data to the fields on the Agreement Role screen
 - The values available in the Record field will be a concatenation of an agreement role type and an agreement

name available to the group customer.

- When the transaction is processed, the AgreementRoleStatus field on the Agreement Role screen will be updated with the value "98," which is the AsCodeAgreementRoleStatus code for "Inactive."
- CopyToClassFields: Copies data to the fields on the Class screen
 - The values available in the Record field will be the names of all classes available to the group customer.
 - When the transaction is processed, the ReportNumber field on the Class screen will be updated with the value "12345."
- CopyToClassGroupFields: Copies data to the fields on the Class Group screen
 - The values available in the Record field will be a concatenation of the name of a class group available to the group customer and the status of all class group records for the group customer.
 - When the transaction is processed, the BusinessStatusCode field on the Class Group screen will be updated with the value "Pending."
- CopyToClientAltIdFields: Copies data to the fields on the Alternate Name screen
 - The values available in the Record field will be a concatenation of an alternate name available to the group customer and that alternate name's Effective From date.
 - When the transaction is processed, the EffectiveTo field on the Alternate Name screen will be updated with the EffectiveDate of the activity if the EffectiveFrom date of the selected alternate name is earlier than the activity's EffectiveDate.
- CopyToClientRelationshipFields: Copies data to the fields on the Group Customer Relationship screen
 - The values available in the Record field will be a concatenation of an AsCodeSecondaryRelationshipType

code value for a relationship record belonging to the group customer and the EffectiveDate of that relationship record.

- When the transaction is processed, the LastCopyToUpdatedDate field on the Group Customer Relationship screen will be updated with the activity's EffectiveDate.
- CopyToGroupCustomerFields: Copies data to the fields on the Group Customer screen
 - The Record field will be hidden when this value is chosen from the CopyToRule field.
 - When the transaction is processed, the CustomerNumber field on the GroupCustomer screen will be updated with a new value generated by the identifier.
- CopyToIntakeFileFields: Copies data to the fields of an Intake File record
 - The values available in the Record field will be a concatenation of the name of a Intake Profile available to the group customer and the Profile's IntakeFileID.
 - When the transaction is processed, the ReceivedRecordCount field on the Intake File will be updated with the value "01."
- CopyToIntakeProfileFields: Copies data to the fields of an Intake Profile record
 - The values available in the Record field will be a concatenation of the name of a Intake Profile and the EffectiveFromDate of that Data Intake Profile.
 - When the transaction is processed, the FatalErrorThreshold field on the Intake Profile will be updated with the value "14."
- CopyToIntakeRecordFields: Copies data to the fields of an Intake Record record.
 - The values available in the Record field will be a concatenation of the IntakeRecordID of an Intake Record available to the group customer and the status of that

Intake Record.

- When the transaction is processed, the PreProcessorErrorCount field on the Intake Record will be updated with the value "02."

View Prototype in OIPA

1. Log into OIPA with credentials for a user belonging to the Prototype Company.
2. Open a group customer record for a group customer with all of the screens/records mentioned above configured.
3. Click the **Add Activity** button on the Secondary menu.
4. Select the **GroupObjectsCopyToBR** activity.
5. Select a CopyTo rule from the **CopyToRule** field.
6. Note that the corresponding value detailed above displays in the Record field.
7. Click **Ok**.
8. Navigate to the screen or record corresponding to the selected CopyTo rule.
9. Note that the value of the field mentioned above has been updated with the appropriate value.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Product-Plan Data Inheritance Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Product-Plan Data Inheritance Prototype

This prototype was configured to demonstrate the inheritance of data within the Product/Child Product/Plan structure. See the Product-Plan Data Inheritance page for a detailed explanation of data inheritance within the Product/Child Product/Plan structure.

Prototype Configuration

- The Term product override of the PlanScreen business rule was configured to demonstrate the utilization of data inheritance. In the **Main Explorer** window, navigate to **Companies | Prototype | Subsidiary Companies | Prototype Group Child Company | Products | Group Prototype Product | Child Products | Term Products | Business Rules | PlanScreen** to view the prototype configuration.
 - The Term Product override of the PlanScreen business rule was configured with three fields, which will be utilized by all Plans related to the Term Product:
 - MaximumFaceAmount
 - MaximumPremiumAmount
 - MaximumIssueAge

View Prototype in OIPA

1. Log in to a Rules Palette environment containing the Prototype company structure.
2. Navigate to the PlanScreen override explained above and open the business rule to view the configuration. Note the fields configured on the **Fields** tab.
3. Navigate to all PlanScreen overrides within Plans related to the Term Product. View the fields configured on the **Fields** tab. Note that the MaximumFaceAmount, MaximumPremiumAmount and MaximumIssueAge fields are displayed in grey, indicating that they are inherited from a higher override level.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Requirement Shading Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Requirement Shading Prototype

This prototype was configured to demonstrate the ability to color-code policy requirements on the Requirement screen.

Prototype Prerequisites

- OIPA user credentials with the necessary security settings to view the Policy Requirement screen

Prototype Configuration

- The CompanyCosmetics business rules was overridden for the Alamere Insurance Company in order to showcase this functionality. In the **Global Rules Explorer** window, navigate to **Business Rules | System | CompanyCosmetics | CompanyCosmetics (Alamere Insurance)**. The key configuration in this business rule is explained below.
 - The <PolicyRequirementShadings> element is present in configuration, as are <Shading> sub-elements.
 - Each <Shading> element has a STATUS attribute specifying the status to which that particular shading color applies. The value of each <Shading> element is a color name defined in the [HTML and CSS color specification](#).
- The AsCodeRequirementStatus code name is configured with code values corresponding to the available requirement statuses.

View Prototype in OIPA

1. Log in OIPA using the Alamere Insurance user ID and password.
2. Open an existing policy containing requirements or create a shell policy with requirements.
3. Navigate to the Policy Requirement Screen.
4. Note that each requirement is colored according to its current status.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Undo Redo of Client Level Transaction

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Undo Redo of Client Level Transaction

OIPA, adds a new feature for Undo/Redo functionality at the client and customer level. New transaction type is now added under Prototype Company for Client Plan named as "Client financial". This new transaction type will be a Reversible/Reversing type of transaction. Earlier the existing type was "Client-Financial" transaction type, now being "Client-Financial -Reversible-Reversing type. The following are available

- Client-Financial (Client-Financial-Reversible-Reversing)
- Client-Financial-Reversible-Non-Reversing

New Transaction Type System Codes are added in the palette to recognize new AsCode transaction types (mentioned above).

Configuration Overview

The new transactions 'ClientFinancialReversibleReversing' and 'ClientFinancialReversibleNonReversing' are added for the Client Plan in Prototype Company. These transactions are expected to get processed on the following way

- On processing the 'ClientFinancialReversibleReversing' activity, the 'ClientFinancialSpawn1' activity gets spawned.
- On recycling/reversing the 'ClientFinancialReversibleReversing' activity, will undo/redo all the active activities above it. As the spawned activity 'ClientFinancialSpawn1' is a reversible, non reversing type, it gets reversed as well.
- On processing the 'ClientFinancialReversibleNonReversing' activity, the 'ClientFinancialSpawn2' activity gets spawned.
- On reversing the 'ClientFinancialReversibleNonReversing' activity, the activities above will not get affected. The spawned activity 'ClientFinancialSpawn2' is not reversed on reversing

Pre-requisite

The OIPA user should have the access to both the transactions in the application and should have the necessary security in order to access the 'Delete' checkbox for the transaction 'ClientFinancialReversibleReversing'.

Prototype Example

ClientFinancialReversibleReversing

- This transaction has only the field 'EffectiveDate' configured.
- On reversing this activity, all the activities above it, other than the non reversible type, will also get reversed, and a new instance of the current activity is not created.
- On recycling this activity, all the activities above it, other than the non reversible type, will also get reversed, and a new instance of the current activity is created.
- The spawned activity 'ClientFinancialSpawn1' which is of type 'ReversibleNonReversing', also gets reversed.

ClientFinancialReversibleNonReversing

- This transaction also has only the field 'EffectiveDate' configured.
- On reversing this activity, the other activities above will not get affected.
- The spawned transaction 'ClientFinancialSpawn2' will also remain unchanged on reversing this activity.

ClientFinancialSpawn1

This transaction also has only the field 'EffectiveDate' configured. It is spawned by transaction 'ClientFinancialReversibleReversing'.

ClientFinancialSpawn2

This transaction also has only the field 'EffectiveDate' configured. It is spawned by transaction 'ClientFinancialReversibleNonReversing.

View Prototype in OIPA

1. Login to OIPA with a Prototype company user credentials.
2. Select a Client Plan
3. Create a new Policy and process it
4. Add new Activity and process the activity for the above transaction types
5. View the result for the new functionality explained above

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > UnmatchedResultSearchScreen Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

UnmatchedResultSearchScreen Prototype

This prototype was configured to demonstrate the ability to configure the UnmatchedResultSearchScreen business rule, to search for unmatched requirement results using the Unmatched Requirement Result Search screen and to match the unmatched results to requirements on the same screen.

Prototype Prerequisites

- The UnmatchedResults Company Page must be given the proper security settings in order to access the UnmatchedResultSearchScreen in OIPA.

Prototype Configuration

- The UnmatchedResultSearchScreen business rule was overridden for the Alamere Insurance company in order to demonstrate the new functionality. The key configuration for this prototype is explained below.
 - The configuration of the requirement result search section only contains the opening and closing <Search> tags, meaning that the default search fields will display. The default fields are:
 - Requirement Status
 - Result Status
 - Received From
 - Received To
 - The table displaying the results of the search for requirement results is configured in the <Results> section. The following columns are configured:
 - Requirement Name
 - Requirement Status
 - First Name
 - Last Name
 - Date of Birth
 - The search section of the Find Requirement tab is also configured to only contain the default search fields. The default fields are:
 - Requirement Status
 - Last Name
 - First Name
 - Created Date From
 - Created Date To
 - The table displaying the results of the search for requirements is configured in the <Results> section. The following columns are configured:
 - Requirement Name

- Policy Number
- First Name
- Last Name

View Prototype in OIPA

1. Log in to OIPA using credentials for a user belonging to the Alamere Insurance company.
2. Select **Requirements** > **Unmatched Results** from the main menu at the top of the screen.
3. The Unmatched Requirement Results Search screen will open. Note that the default search fields mentioned above display. Enter some search criteria in the requirement result search fields.
4. Any results returned from the requirement result search will appear on the screen. Note that the columns configured for the search results table display on the screen.
5. Click on the **Result Details** icon next to a requirement result returned from the search.
6. Click on the **Find Requirement** tab. Note that the default search fields mentioned above display.
7. Enter some search criteria to search for requirements with the same requirement definition as the selected requirement result.
8. Any requirements fulfilling the search criteria will appear on the screen. Note that the columns configured for the search results table display on the screen. Click on the **Requirement Details** icon to view detailed information for the requirement.
9. Select the **Match** button displayed next to a requirement returned from the search to attach it to the requirement result.
10. Click **OK** in the dialog box to finalize the match.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Activity Address Add and Update

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Activity Address Add and Update Prototype

OIPA provides the ability to configure a transaction that can add or update address information and then attach it to a client or array of client records.

There are two business rules that can be attached to the transaction to support this functionality.

- **CreateClientAddress:** attached to a transaction to identify the ClientGUID or ClientGUIDArray that the address should be added to when the activity successfully processes.
- **CopyToAddressFields:** attached to a transaction to copy information from Activity fields to Address fields.

In addition, there are two additional business rules that support this functionality.

- **AddressScreen:** address types and associated fields are defined in this rule. This should not be attached to the transaction.
- **MultiFields:** this rule can be used to allow users to select a number from a configured drop down box on the screen, which determines the number of subsequent fields that should be displayed.

Refer to the XML Configuration Guide for a complete explanation of the elements and attributes supported in each rule.

Configuration Required for Adding or Updating Address Information Using an Activity

There are several steps involved in configuring a transaction that will add or update address information.

- A transaction must be created to capture the address information.
- The **CreateClientAddress** rule must be attached to identify the client/clients that receive the new address information.
- The **AddressScreen** business rule must be configured. The transaction will access the rule to pull the address fields in for the user to enter the new address through the use of the <Address> element configured within the transaction configuration..

Prototype Explanation

A prototype was configured to demonstrate all aspects of the activity address functionality. There are six transactions and four business rules that were configured to demonstrate the functionality. An explanation of each piece of configuration is provided below.

Business Rules

- **MultiField-Phones** business rule was created. This is a simple two field rule that allows the user to add up to three different types of phones to the Address screen (Home, Work, Cell).
- **AddressScreen** business rule was updated to include the newly created MultiField-Phones rule in its Address Type 03, 04, and 05 sections.
- **CreateClientAddress** has four versions that were created and attached to AddFamilyAddress, ClientAddressClearOnRecycle, ClientExpireAddDefaultAddress, and ClientExpireAndAddAddress transactions. The configuration demonstrates the <ClientGUID> element (filled using activity Field and also MathVariable), and <ClientGUIDArray> (MathVariable).
- **CopyToAddressFields** business rule was attached to ClientExpireAddDefaultAddress ClientExpireAndAddAddress transactions, and are used to expire selected addresses.

Transactions

- **AddFamilyAddress**: this is a policy-level financial non-reversible, non-reversing transaction that contains an <Address> element that uses the DEFAULTADDRESSTYPE attribute. The <AddressTypes> sub-elements limit the number of available address roles. CreateClientAddress is attached to this transaction. The purpose of the transaction is as follows:
 - allow the user to select a Client-Role combination on a policy in the Activity Details screen.

- complete a new address for the selected address role country on the activity Address tab.
 - generate an array of clientGUIDs (in a math variable) of all clients with the same last name as the Client-Role selected on the Activity Detail screen.
 - add the newly created address to each client identified in the ClientGUIDArray. This is governed by the CreateClientAddress rule attached to the transaction. This transaction also makes use of new syntax that allows transaction math to access activity AddressGUID (Activity:Address:AddressGUID) and then use it to access specific Address Field (CountryCode).
- **ClientExpireAddDefaultAddress:** this is a client-level financial non-reversible/non-reversing transaction that contains an <Address> element that uses the COPYSOURCEADDRESSGUID attribute set equal to Address Field, configured through the use of the <Calculated> element with a METHOD="FORCE" attribute that identifies the Default Address. This transaction allows a user to accomplish the following tasks:
 - expire the default address(through the use of CopyToAddressFields rule)
 - add a new address of the same address role (through the use of CreateClientAddress rule and Client GUID math variable). In order to designate this new address as the default address, the user will have to actually go to the Address screen and click on the default radio button.
- **ClientExpireAndAddAddress:** this is a client-level financial non-reversible, non-reversing transaction that contains an <Address> element that uses <AddressTypes> sub-elements to limit the number of available address roles. This transaction allows a user to accomplish the following tasks:
 - select an address to expire
 - expire the address (through the use of CopyToAddressFields rule)
 - add a new address from the list of address roles designated in <AddressTypes> sub-element (through the use of CreateClientAddress rule and Client GUID field). The user can designate the new address as Default, by actually going to the Address screen and clicking on the Default radio button.

- **ClientSpawnAddress:** this is a client-level financial non-reversible, non-reversing transaction that contains an <Address> element that uses the COPYSOURCEADDRESSGUID attribute set equal to Address field. This transaction allows a user to accomplish the following tasks:
 - select a client address
 - spawn a policy-level PolicyAddressChangeLetter activity through the use of spawn code 09
 - pass the address GUID of the selected address to activities
 - make sure that the spawned policy-level PolicyAddressChangeLetter activity contains Address tab
- **PolicyAddressChangeLetter:** this is a policy-level financial transaction that contains an <Address> element that uses the COPYSOURCEADDRESSGUID attribute set equal to Address field (with the value passed from ClientSpawnAddress activity).
- **ClientAddressClearOnRecycle:** this is a client-level financial transaction that contains an <Address> element that uses the COPYSOURCEADDRESSGUID attribute set equal to Address field. This transaction allows a user to accomplish the following tasks:
 - demonstrate the ClearOnRecycle functionality of the Address field. Select the **Update Address** check box, to fill-in the Address field's combo-box (empty on load), and then select a COPYSOURCEADDRESSGUID attribute. When the activity is processed and then recycled, even though the Address combo-box becomes empty again for the new pending activity, the Address tab of the activity will still reflect the clone of the address that was there while the activity was active. If the user chooses to change the address on the new pending activity, the Address tab should then change the clone to reflect the newly selected address.
 - demonstrate the effect of deleting or recycling an activity when it added an address.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Activity Confirmation Screen

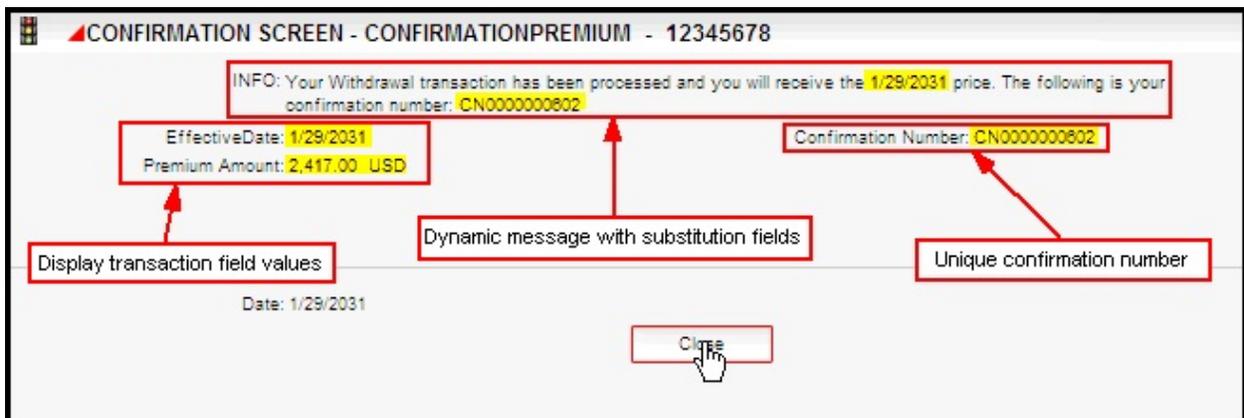
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Confirmation Screen Prototype

The Confirmation screen allows OIPA to display information to the user when an activity is added. For example, this enables a user to provide a unique confirmation number to give to a customer on the phone. If the optional ConfirmationScreen business rule is configured and attached to a transaction, then a confirmation screen automatically opens when a user adds a new activity to the Activity screen, recycles an activity, or returns to the Activity screen from viewing existing activity details (accessed from the Activity Detail Icon).

The confirmation screen may be configured to generate and display a confirmation number, using either an [Identifier field](#) or an [Identifier math variable](#). The Identifier math variable supports a new capability to generate a unique confirmation number each time the main activity spawns another activity. The confirmation screen also may display dynamic [messages with substitution fields](#), and the values of screen math or fields from the main body of the transaction.

The recommended practice is to use an Identifier field to generate the confirmation number of the originating (parent) activity, and use Identifier math variables (from the transaction's Math section) to generate confirmation numbers that may be passed to child activities that require their own unique confirmation numbers.



Confirmation Screen with Confirmation Number

Scenario

A customer service representative enters the necessary information on a withdrawal screen while on the phone with a customer. The CSR processes a verification of withdrawal information and provides the customer with detailed information on the withdrawal. The CSR adds the withdrawal activity from the Verification screen and a Confirmation screen is displayed with a message that includes the activity effective date and a unique confirmation number. The CSR provides the effective date and confirmation number to the client. The CSR then processes the pending withdrawal activity. Disbursement and Withdrawal letters are spawned by the withdrawal activity, each with a unique confirmation number.

Example 2

A servicing agent has delivered the necessary paperwork to the office to process a beneficiary change for a client. The CSR enters the necessary information in the beneficiary change activity window and presses the OK button to add the activity. The Confirmation screen is displayed with a message that includes the activity effective date and a unique confirmation number. The CSR provides this confirmation information to the agent.

Prototype Samples in Rules Palette

There are two prototype transactions configured to demonstrate confirmation screen functionality. Navigate through the following folders in the Main Explorer to locate the configuration sample: **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions.**

- **ConfirmationPremium** This transaction is considered the parent activity. It has several features that demonstrate the full range of ConfirmationScreen functionality:
 - **ConfirmationNumber Identifier field**

This field is disabled by default, not hidden. It displays on the activity but does not accept data. This field is passed to the Confirmation screen. It has a <ClearOnRecycle> element set to Yes, which clears the ConfirmationNumber field if the ConfirmationPremium activity is manually recycled. If the activity is reversed as the result of a system undo-redo cycle, the ConfirmationNumber field is not cleared.
 - **ConfirmationNumberMV Identifier math variable**

This variable value is passed to a child activity (ConfirmationNumberMath). The configuration associated with ConfirmationNumberMV has a mechanism to persist the original ConfirmationNumberMV value if the activity is re-done due to a system-generated undo/redo.
 - **An attached TransactionCosmetics business rule**

This rule is configured with a <Tooltip> element that displays the activity Confirmation Number when the cursor is “hovered” over the Activity Details icon on the Activity Screen.
 - **An attached ConfirmationScreen business rule**

This rule is configured to display the values of several fields configured within the main body of the transaction. It also generates a dynamic message with substitution fields from the main body of the transaction.
 - **An attached QuoteScreen business rule**

This rule is configured to demonstrate the compatibility of the QuoteScreen with the ConfirmationScreen; and of Add New Activity with the confirmation screen (by adding the ConfirmationNumberMath transaction).

- **ConfirmationNumberMath:** This transaction is considered the child, or spawned activity. This transaction demonstrates that the Identifier math variable can be passed from the parent (spawning) activity (ConfirmationPremium) to the child (spawned) activity (ConfirmationNumberMath). It contains the following configuration features:
 - **ConfirmationNumber text field**

This demonstrates that the value of the ConfirmationNumberMV Identifier math variable in the ConfirmationPremium transaction can be passed to a child activity. The format of the confirmation number math variable does not need to match the format of the confirmation number Identifier field. Also, the Identifier math variable and the Identifier field sequence numbers are separate series from the AsSequence table.
 - **Attached ConfirmationScreen business rule**

This rule demonstrates that a parent activity (ConfirmationPremium) with an attached QuoteScreen, can use the Add New Activity feature of the QuoteScreen to spawn a child activity (ConfirmationNumberMath) with an attached ConfirmationScreen.

The ConfirmationPremium transaction and the attached QuoteScreen rule are configured so that the value generated by the ConfirmationNumberMV math variable may be passed to the ConfirmationNumberMath activity in two ways:

- By adding the ConfirmationNumberMath activity from the Add New Activity tab of the Quote Screen
- By processing the ConfirmationPremium transaction and spawning the ConfirmationNumberMath activity

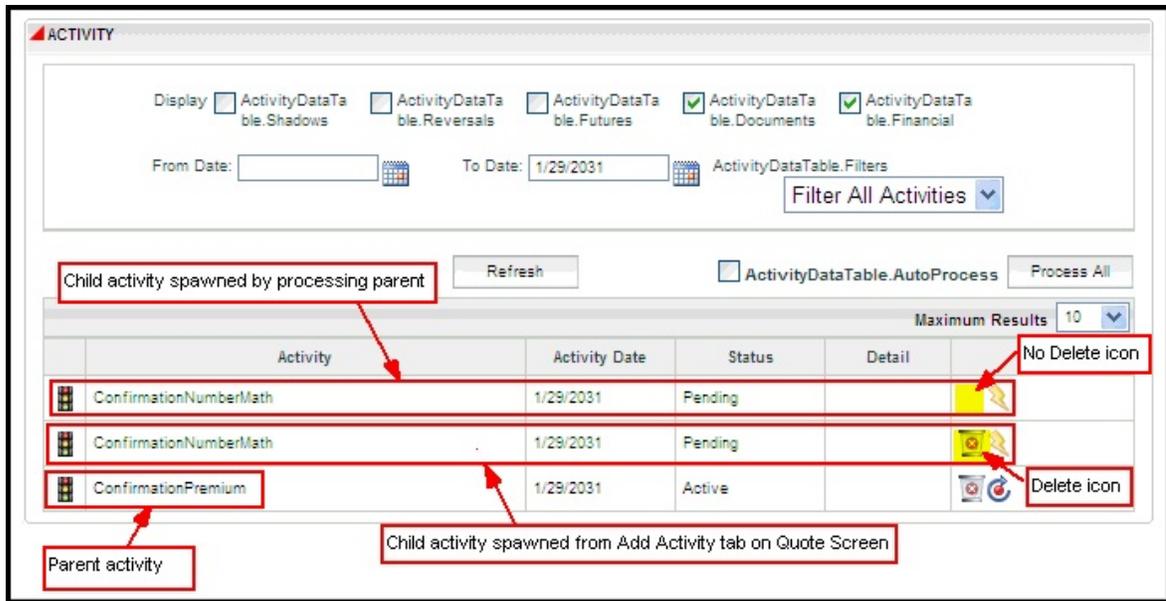
If you use both — Add New Activity from the Quote screen, and

processing the ConfirmationPremium activity — two separate ConfirmationNumberMath activities are added to the Activity Screen. In each child activity, a unique ConfirmationNumber field value is generated, passed from the ConfirmationNumberMV math variable in the ConfirmationPremium transaction. This demonstrates that the Identifier math variable can pass a unique confirmation number from a parent activity to a child activity, and that the QuoteScreen business rule is compatible with the ConfirmationScreen business rule.

View New Feature in OIPA

To view the prototype example in OIPA:

1. Log into the Prototype Child Company and select the Functional Prototype Plan.
2. Create a new policy and add roles, a suspense record, and at least one segment, then click the Activities link.
3. Click **Add Activity** from the Secondary menu and select the ConfirmationPremium transaction, then click **OK**.
4. From the Activity screen, select the Activity Details icon to the left of the pending ConfirmationPremium activity. This will open the Activity Details screen.
5. Select the Quote button to open the Quote Screen.
6. Select the Add New Activity tab on the Quote Screen, and click on the Select button. The ConfirmationNumberMath activity window opens.
7. Select Ok to spawn the child activity, then select Close on the confirmation screen that follows.
8. Close the Quote Screen by selecting the 'X' icon at the top right.
9. Select Ok on the ConfirmationPremium Activity Details screen, then select Close on the confirmation screen to close it. The Activity screen lists the ConfirmationNumberMath activity.
10. Process the ConfirmationPremium activity from the Activity screen. A second ConfirmationNumberMath activity appears on the Activity screen. Note that the ConfirmationNumberMath activity that you added with the Add New Activity tab differs from the ConfirmationNumberMath activity spawned by processing the ConfirmationPremium activity. It has a Delete icon, and the spawned activity does not.



11. Process both ConfirmationNumberMath activities. Note that the ConfirmationNumberMath activity you added from the Quote screen has both Delete and Recycle icons, while the spawned activity does not have either one. If the parent activity is recycled, the spawned activity is removed, but the added activity is not.
12. After ConfirmationPremium processes, click the Activity Detail icon to the left of the activity.
13. Check the Entry Fields link and the Math link. Note that the sequence numbers in the confirmation numbers do not follow the same series. The Identifier math variable sequence is distinct from the Identifier field sequence. All Identifier fields will obtain sequence numbers from one series of sequential integers, while Identifier math variables obtain sequence numbers from another series.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Automatically Update Activity Effective Date

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Automatically Update Activity's Effective Date

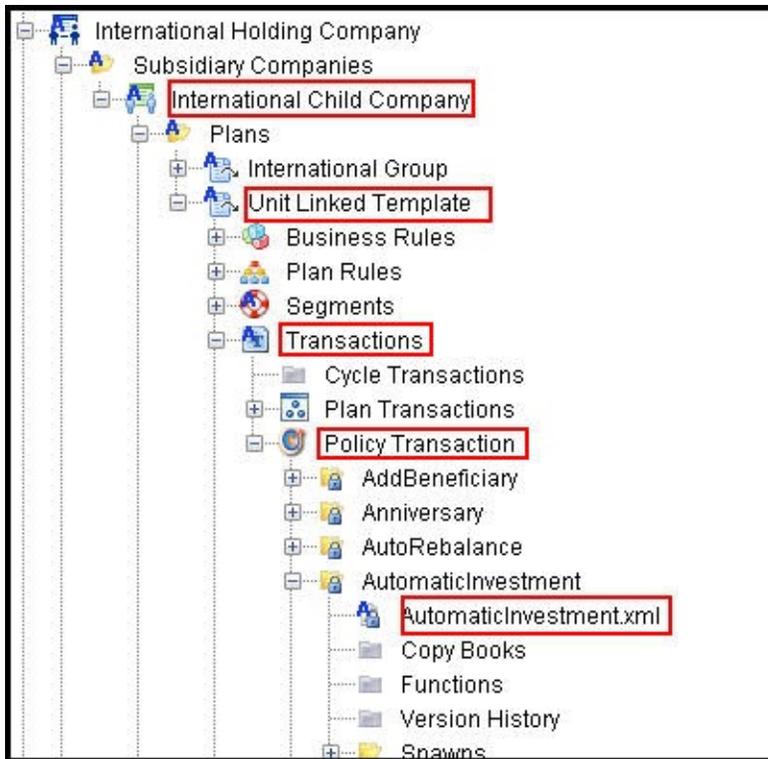
OIPA can automatically update the effective date of an activity as a way to transfer Gain/Loss to the client. This new functionality is available for Unit Linked activity processing. It validates that an activity can process and then systematically sets the activity's effective date to the current system date without accessing the UI. The effective date will be updated automatically when criteria is in place that requires the date to be advanced.

A new element, **Transitions**, will force an activity that would otherwise go into Pending or NUV Pending to go into Queued status. The following transactions in the Unit Linked Template will go to a Queued status if there are prior activities in NUV Pending status that also use any funds that are included in the current activity's allocations:

- AutomaticInvestment
- AutoRebalance
- Switch
- Withdrawal

Withdrawal is the only transaction that will automatically update the activity effective date; the others will hold the original transaction date.

Navigate to **Main Explorer | International Holding Company | Subsidiary Companies | International Child Company | Plans | Unit Linked Template | Transactions | Policy Transactions | *Name of Transaction***. Open the XML Source pane and scroll down to the Transitions section to view XML for this new element.



Navigation for AutomaticInvestment Transaction in Unit Linked Template

Configuration Requirements

- Policy-Financial transactions have a new section in the Transaction General Pane called **Transitions**.
 - **New Transaction Wizard**: When a new Policy-Financial transaction is created, the wizard will have a new checkbox in the Template step for Transitions. Checking this box will add a Transitions section to the Transaction General Pane.
 - **Transaction General Pane**: The Transitions section will have two fields:
 - **Method**: A drop down box is provided with one available selection, which is Valuation. This is the default setting and is a required attribute that identifies the validation that is driving the Queue determination. When the transitions check box is unchecked this field should be blank.
 - **AdvanceToSystemDate**: An auto-complete text field is provided that can accept a literal value of **Yes** or **No**. This auto-complete text box loads all field names and math variable names that are text datatype. This field resolves copybooks contained within the transaction. If a literal value of Yes or No value is not used, then a field name or math variable name that holds a valid value of Yes or No should be selected. This is an optional field. The default value will be blank when the Transitions check box is selected or unselected and if blank, then the behavior will be the same as if No had been entered.

XML Example

```
<Transaction>  
<EffectiveDate STATUS="Enabled" TYPE="SYSTEM"/>  
<Transitions>  
<Queue METHOD="VALUATION"  
ADVANCETOSYSTEMDATE="Yes"/>  
</Transitions>  
</Transaction>
```

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Transactions > Transaction Comments

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transaction Comments

Transactions can be configured to support comments. This allows CSRs to add notes or additional details to an activity in OIPA. Security controls whether these comments can be viewed by all users or only the user who added the comment. Security also controls whether a user is able to add comments to an activity.

Scenario

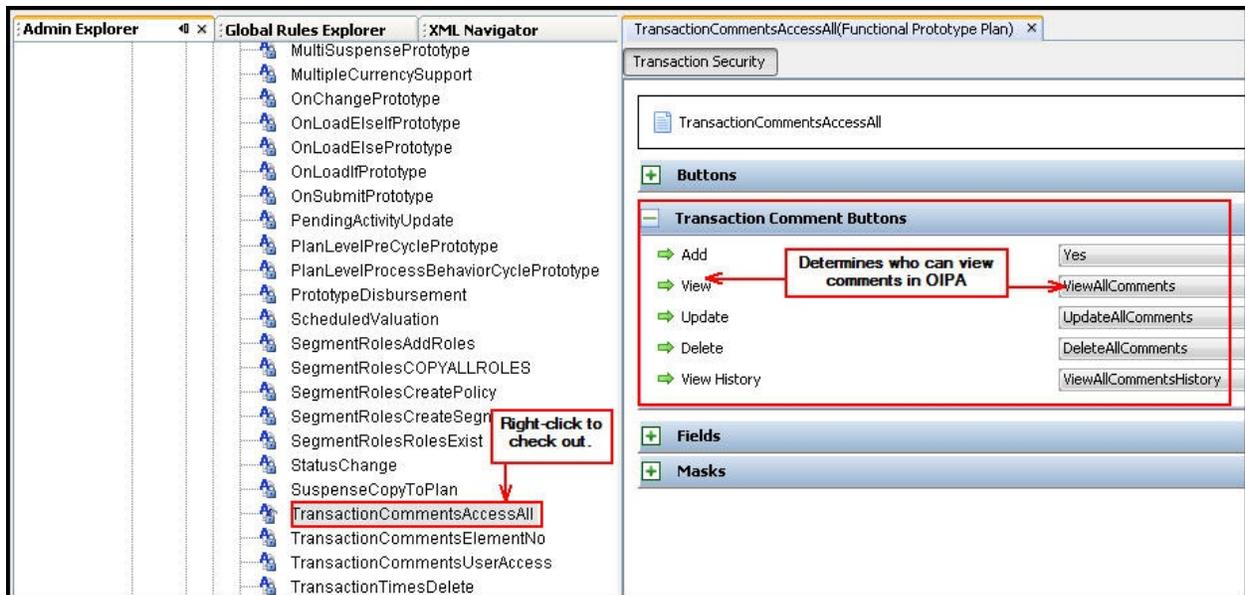
A CSR opens the Activity screen for a policy. The comment icon for one of the activities indicates that comments exist. The CSR clicks the comment icon and reads all comments added for that activity.

A CSR opens the Activity screen for a policy. The comment icon for one of the activities indicates that comments exist. The CSR clicks the comment icon but is only able to see the comments he added. All other comments added by other CSRs are not visible.

Configuration Requirements

- A transaction must be configured to support comments. This means the <AllowComments> element must be configured with a value of **Yes**. Two transactions were created in the Functional Prototype Plan with this element.
 - TransactionCommentsUserAccess : Navigate to **Global Explorer | Transactions | TransactionCommentsUserAccess (Functional Prototype Plan)**.
 - TransactionCommentsAccessAll : Navigate to **Global Explorer | Transactions | TransactionCommentsAccessAll (Functional Prototype Plan)**

- Security must be defined for the comments in [Transaction security](#). Security was added for comments on the two transactions mentioned above. The **Prototype Super** security group contains the transactions with the two security options defined. Navigate to **Admin Explorer | Security | Application Security | Security Groups | Prototype Super | Transaction Security | Prototype Child Company | Functional Prototype Plan**. Then locate the transaction names.
 - **TransactionCommentsUserAccess**: This transaction's comment security allows users in the Prototype Super security group to add comments to instances of this transaction. The user is restricted to viewing, updating and deleting only those comments added by that user.
 - **TransactionCommentsAccessAll**This transaction's comment security allows users in the Prototype Super security group to add comments, view all comments, delete all comments, update all comments and view all comment history associated to instances of this transaction.



Transaction Security in Admin Explorer

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Update Fields With Activity

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Update Fields With Activity

The CopyToPendingActivityFields business rule was enhanced to use conditional logic to determine whether the activity fields of an activity are updated or not after the activity processes.

Error logic is included for cases where an activity attempts to update the activity fields of an activity that is not in pending status. This will produce an error and the activity process will be halted.

Configuration Requirements

Two transactions were created to demonstrate this functionality. The PendingActivityUpdate activity specifies the fields that will be updated. The ActivityToUpdate activity receives the information identified in the PendingActivityUpdate activity.

The CopyToPendingActivities business rule is attached to the PendingActivityUpdate transaction and can be found in the Attached rules folder under the transaction.

- **PendingActivityUpdate:** This activity determines which fields will be updated. Navigate to **Global Explorer | Transactions | PendingActivityUpdate (Functional Prototype Plan)**. An explanation of the configuration is provided below.
 - **Update Which Field:** this field is a radio button. It allows the configuration to attempt update of different fields: a date field and a money field. Value 01 is Effective Date and Value 02 is Grace Money.
 - **Update Multiple Activities:** This field is a radio button. It allows the configuration to attempt the update of multiple activities whether or not there are actually multiple activities to update. Value 01 is Yes and Value 02 is No.
 - **Update Which Activity:** This field is a radio button. This field allows the configuration to attempt the update of pending or active activities of the ActivityToUpdate transaction. It also allows the activity to attempt to update the executing PendingActivityUpdate activity. Value 01 is Pending Activity. Value 02 is Same Activity and Value 03 is Active Activity.
 - **Required Text:** This is a text field. A value is required with each execution of the activity. Using the **Update Which Activity** radio button above, a single activity will attempt to update **Required Text** on the target activity. If there are multiple target activities, one activity will be targeted at random.
 - **Update Date To:** This is a date field. It is only available when **Update Which Field** is the Effective Date. • If a value is set,

then the Effective Date of the targeted activity(ies) will attempt to be updated with the date provided here. An activity cannot modify another activity's Effective Date that will place the updated activity prior to the currently executing activity.

- **Update Grace Money:** This is a money field. It is only available when **Update Which Field** is Grace Money. If a value is set, then the Grace Money Received field of the targeted activity(ies) will attempt to be updated with the value provided here.
- **ActivityToUpdate:** This activity receives the updates as determined by PendingActivityUpdate. Navigate to **Global Explorer | Transactions | ActivityToUpdate (Functional Prototype Plan)**. There are two fields configured in this activity.
 - **Grace Money Received:** This is a money field. If the PendingActivityUpdate field named UpdateWhichField is set to Grace Money then this field will receive the value provided in the PendingActivityUpdate field named Update Grace Money. The default value is 0.
 - **Required Text:** This is a text field. The information entered into the Required Text field on the PendingActivityUpdate activity will be copied to this field. If there are multiple activities, one activity will be targeted at random.
- **CopyToPendingActivityFields** business rule: This rule demonstrates the use of conditional logic through the conditional statements within the <Test> element tags. The <FromCollection> element is used to update multiple activities. The ACTIVITYGUID attribute in the <Activity> element is used to identify a single activity to update.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Address Change Letter

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Address Change Letter Prototype

When an address is changed in OIPA, configuration supports the ability to spawn a client level activity that may be configured to generate an address change letter. Configuration can also direct that the client level activity spawn a policy level activity that contains the changed address information.

The Spawn IF logic within the AddressScreen business rule determines when and if a client level activity is generated. Address types defined in the rule must have an OnSubmit event with a Spawn section in order to use this functionality.

Configuration Required for Address Change Letter

There are several business rules and transactions that demonstrate the address change letter functionality.

- The **ClientScreen** business rule must identify the plan the transaction is created on in the `ACTIVITYPLAN` attribute of the `<Client>` element. If a client type is not associated with the plan, then the client level activity cannot be processed for that client.
- The **AddressScreen** business rule configuration needs to include an `OnSubmit` event section in each address type that should generate an address change letter. The `OnSubmit` event section should include a `spawn` element that identifies the `ClientAddressChangeLetter` activity. Once the address changes are made and the `OnSubmit` validations pass, the address will be saved and the activity spawned for the client.
 - Only client-level activities can be spawned and the Address screen is the only screen where this is supported.
 - `SpawnCode 03` is the only code that can be used.
 - The `IF` attribute sets the spawn conditions. The `AddressScreen`'s fields and screen math variables (local screen math executed by `OnSubmit` and global) must be available for the conditional expression. References to screen math variables in the `spawn` section follow the syntax **ScreenMathID:MathVariable name**. This is the same syntax used for actions.
 - Each necessary field in the spawned activity should be identified in a `<SpawnField>` section. When the spawned activity executes, it will not complete correctly or successfully if the necessary input into the activity has not been provided.
 - The spawned activity will only spawn for the client to which the address belongs. It may not be spawned to any other client.
- A **ClientAddressChangeLetter** activity must be configured. This is the activity identified in the `OnSubmit` `spawn` section of the `AddressScreen` rule. This activity can also be configured to spawn a policy level activity that captures the updated address information.

Configuration Explanation

- **ClientScreen** business rule: This rule was overridden at the Prototype Company level. All client types are associated to the Client Plan, where the ClientAddressChangeLetter was created. Each <Client> element has an attribute called **ACTIVITYPLAN="Client Plan">** that identifies the plan when the client level activity was created. Navigate to **Global Explorer | Business Rules | Screen Rules | ClientScreen (Functional Prototype Plan)**.
- **AddressScreen** business rule: This rule was overridden at the Prototype Company level. The **Residence** address type (03) spawn section demonstrates the configuration that spawns the ClientAddressChangeLetter. A Spawn section was added to OnSubmit. Once the OnSubmit validations pass, the address is saved and the activity is spawned on the client. The spawn identifies the transaction name. Spawn IF logic is used to make the spawn conditional so any changes to inconsequential fields won't prompt the letter. The Spawn IF logic determines when and if it is appropriate to spawn the ClientAddressChangeLetter activity. Navigate to **Global Explorer | Business Rules | Screen Rules | AddressScreen (Functional Prototype Plan)**.
- **ClientAddressChangeLetter**: This transaction was created as a client level activity on the Client Plan. Navigate to **Global Explorer | Transactions | ClientAddressChangeLetter (Client Plan)**.

View Prototype Example in OIPA

1. Log in OIPA using the Prototype Company user ID and password.
2. Click **Client | New** from the Main menu.
3. Type the new client information and save the record.
4. Click **Address** from the Left Navigation menu.
5. Add an address and save.
6. Select the saved address and make a change to the address information. Save the changes.
7. Click **Activities** in the Client screen Left Navigation menu. Notice the ClientAddressChangeLetter activity is listed in pending status.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Application Screen
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Application Screen Prototype

OIPA now supports an Application screen, which allows a user to view an Application's data.

Prototype Explanation

The following business rules will be used to demonstrate this new functionality.

Business Rules

- **PolicyScreen:** This business rule will be overridden at the system level to demonstrate the ability to create the Application screen. Navigate to **Global Rules Explorer | Plan Rules | PolicyScreen** to see the rule.

View Prototype in OIPA

1. Log in to the Rules Palette.
2. In the Global Rules Explorer window, navigate to **Business Rules | Plan Rules | PolicyScreen**.
3. Right-click on the PolicyScreen node and select **New PolicyScreen Override**.
4. The Create New Rule Override wizard will open. Click **Next** on the first screen of the wizard.
5. Select all of the desired override information, making sure to select **NBU** from the **System** drop-down box.
6. Click **Finish** to close the wizard and create the override.
7. Log in to OIPA with credentials for a user belonging to the company in which the PolicyScreen override was just created.
8. From the top menu, select **Case > Search Case**.
9. Enter search criteria in the Case Search Criteria section of the screen.
10. Select a case from the Case Search Results. The Case screen will open.
11. Select an application record. The Application screen will open, displaying the application's information.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases >
AllowPlanDateModifications Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

AllowPlanDateModifications Prototype

The <AllowPlanDateModifications> element in the PolicyScreen rule controls when the Plan Date fixed field becomes disabled on OIPA's Policy screen.

Prototype Explanation

The following business rules were configured in the Prototype Company to demonstrate this new functionality.

Business Rules

- **PolicyScreen** (Functional Prototype Plan): This business rule is configured so that the Plan Date field is disabled when the policy is in Active, Inactive or Grace status. When the policy is in any other status, the Plan Date field is editable. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Plan Rules | PolicyScreen** and click on the **XML Source** pane. The key configuration for this business rule is explained below.
 - <AllowPlanDateModifications> is set to "Yes," meaning that the Plan Date field will not be automatically disabled when the policy is saved for the first time.
 - <DisabledStatus> has a value of "01,05,03," meaning that the Plan Date field will not be editable in the Active, Grace or Inactive statuses.
- **PolicyScreen** (Dynamic Prototype Plan): This business rule is configured so that the Plan Date field is disabled when the policy is saved for the first time. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Dynamic Prototype Plan | Plan Rules | PolicyScreen** and click on the **XML Source** pane. The key configuration for this business rule is explained below.
 - <AllowPlanDateModifications> is set to "No," meaning that the Plan Date field will be automatically disabled when the policy is saved for the first time.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password.
2. Select **Policy > New**.
3. In the Plan field, select **Functional Prototype Plan**.
4. Enter enough information on the screen to create a shell policy.
5. Click **Save**. Note that the Plan Date field remains editable after saving the policy.
6. Add the activities necessary to bring the policy into Active status.
7. Navigate back to the Policy screen and note that the Plan Date field is no longer editable.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Benefit Split
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Benefit Split Prototype

In OIPA, Variable Annuity products use Benefit Split functionality to generate a policy's guaranteed payment stream upon annuitization. There are many configuration steps involved in establishing this process. This prototype example gives the location of all the pieces of configuration that support the Benefit Split process.

Refer to the [Benefit Split](#) section of this help for complete configuration steps.

Prototype Examples

Benefit Split functionality was configured in the **Dynamic Prototype Plan**.

Plan Rules

These rules can be found in the **Main Explorer** under **Prototype Company | Subsidiary Company | Prototype Child Company | Dynamic Prototype Plan**. The specific folder under the Dynamic Prototype Plan where the rule resides is identified next to each rule.

- **FundListForAllocation**: this rule is located in the **Business Rules | Attached** folder.
- **PolicyScreen**: this rule is located in the **Plan Rules** folder.
- **EligibleTransactionsByPolicyStatus**: this rule is located in the **Plan Rules** folder.
- **CalculateGeneralBenefitSplit**: this rule is located in the **Business Rules | Calculate** folder.

Funds

The following funds were configured to support Benefit Split processing. These funds are located in the **Admin Explorer** under **Administration | Funds | Prototype Company**.

- ABL Fixed
- Dynamic Fixed
- Variable Fund 1
 - Variable Fund 1 125
 - Variable Fund 1 125 3.0 (Benefit Fund)
 - Variable Fund 1 125 3.5 (Benefit Fund)
 - Variable Fund 1 125 5.0 (Benefit Fund)
 - Variable Fund 1 160

- Variable Fund 1 160 3.0 (Benefit Fund)
 - Variable Fund 1 160 3.5 (Benefit Fund)
 - Variable Fund 1 160 5.0 (Benefit Fund)
- Variable Fund 2
 - Variable Fund 2 125
 - Variable Fund 2 125 3.0 (Benefit Fund)
 - Variable Fund 2 125 3.5 (Benefit Fund)
 - Variable Fund 2 125 5.0 (Benefit Fund)
 - Variable Fund 2 160
 - Variable Fund 2 160 3.0 (Benefit Fund)
 - Variable Fund 2 160 3.5 (Benefit Fund)
 - Variable Fund 2 160 5.0 (Benefit Fund)

Map Group

The following Map Group was created to support Benefit Split. It can be found in the Admin Explorer in **Administration | Map Groups**.

- InvestmentRate

Segment

The following segment was created in the Dynamic Prototype Plan. It can be found in the Main Explorer under **Prototype Company | Subsidiary Company | Prototype Child Company | Dynamic Prototype Plan | Segments**.

- BenefitSplit

Transactions

The following transactions were configured to demonstrate Benefit Split.

They can be found in the Main Explorer under **Prototype Company | Subsidiary Company | Prototype Child Company | Dynamic Prototype Plan | Transactions**. A list of the attached business rules for each transaction is also provided.

- ABLStart
 - TransactionBusinessRulePacket
 - TransactionCosmetics
- ApplyPremium
 - TransactionBusinessRulePacket
 - TransactionCosmetics
- Issue
 - StatusChange
 - CreateSegments
 - TransactionBusinessRulePacket
 - TransactionCosmetics
- Switch
 - DoBenefitSplitChange
 - TransactionBusinessRulePacket
 - TransactionCosmetics
- QDROSplit
 - CreateSegments
 - DoBenefitSplitChange
 - TransactionBusinessRulePacket
 - TransactionCosmetics

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Called Event and Call External Event Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Called Event and Call External Event Prototype

OIPA now allows Events in a transaction to generate an Action that results in triggering an event configured in a BR associated with the transaction. This feature is provided to cater to a situation where the action to be performed in the target BR is to be triggered on an event defined in the transaction but the action to be performed is also dependent on the values in one or more fields in the target BR.

This functionality will be supported by defining an Action in the target BR (MultiField BR is used to demonstrate the capability in this case) as a CALLEDEVENT with an ID attribute. Such a called event defined in the target BR can be triggered from the transaction through an Action of type CALLEXTERNALEVENT which is defined in the transaction with the same value in the ID attribute.

Currently this feature is supported in MultiFields BR and TransactionAllocationScreen BR. This may be extended to other associated BRs in future.

Scenario

In a specific transaction, a specific event to assign value or validate values in a Multifield in one or more instances of the multifield is required to be triggered on the basis of a specific event inside the transaction. The event is dependent on the values in both the transaction field and the multifield values.

Prototype Configuration

- The enhancement is added to an existing transaction 'MultifieldEvents' in the Functional Prototype Plan under Prototype Company. The events of type 'CALLEXTERNALEVENT' are configured in the transaction which in turn invokes the events of type 'CALLEDEVENT' present in the 'Multifield-MultifieldEvents'.
 - Transaction Field: "TestField" is created as a Transaction field to demonstrate this functionality. Based on the value of TestField and the event type, the corresponding events are triggered in the Multifield.
 - Transaction Events: The following events are configured to demonstrate the functionality.
 - OnLoad: In the transaction, on load event, the Action of type 'CALLEXTERNALEVENT' is configured which invokes the event 'MFEEventOnLoad' present in the MultiField – MultiFieldEvents business rule.
 - OnChange: In the transaction, the Action of type 'CALLEXTERNALEVENT' is configured which invokes the event 'MFTestTextEventOnChange' and 'MFTestComboEventOnChange' present in the MultiField – MultiFieldEvents business rule.
 - OnSubmit: In the transaction, the Action of type 'CALLEXTERNALEVENT' is configured which invokes the event 'MFEEventOnSubmit' present in the MultiField – MultiFieldEvents business rule.
 - Multifield Events: This is an existing multifield business rule, where four multifields- TextTestMF, ComboTestMF, RadioTestMF and CheckTestMF are configured. The events invoked by transaction are configured in the Multifield-MultiFieldEvents multifield. The following events are configured in the same:
 - MFEEventOnLoad: This event is associated with the multifield CheckTestMF. It invokes actionset MFActionOnLoad which displays a warning message and marks the field CheckBoxTest1 as 'CHECKED' on load.

- MFTestTextEventOnChange: This event is associated with multifield TextTestMF. Based on the following values provided to the field TestText2 of index 0, the corresponding action is configured for the field TestText1 of index 0:
 - Hide
 - Show
 - Disable
 - Enable
 - Assign
 - ReadOnly
 - Test - Display and error message.
- MFTestComboEventOnChange: This event is associated with multifield ComboTestMF. It invokes queryset "TestComboOptionsActionOnChange" for field "ComboTest5".
- MFEventOnSubmit: This event is configured for multifield RadioTestMF. It invokes actionset MFActionOnSubmit which displays a warning message.

View Prototype in OIPA

1. Log in OIPA using the Prototype Company user ID and password.
2. Click **Policy | New** from the Main menu.
3. Add a shell policy to test the prototype.
4. Click **Add Activity** on the Secondary menu.
5. Select the MultifieldEvents activity from the Activity drop down box.
6. Make changes to the various fields view the action event triggered as explained above to view this functionality in action.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Case Screen
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Case Screen Prototype

OIPA now supports a Case screen, which allows a user to create and edit case records as part of the New Business Underwriting process.

Prototype Explanation

The following business rules were configured as global rules to demonstrate this new functionality.

Business Rules

- **CaseScreen:** This business rule was configured to display a set of predefined fields when a case record is selected. To view the prototype configuration, navigate in the Global Rules Explorer to **Business Rules | Screen Rules | CaseScreen** and select the XML Source pane. The key configuration for this business rule is explained below.
 - A text field called CaseDetails field is configured with a display name of "Case Description" and a date field called CreationDate is configured with a display name of "Creation Date."
 - An action is configured to display an error message if the Case Description field is left empty.
 - The table that displays the case records is configured with Application Number, Issue State, Status, Insured First Name and Insured Last Name columns.

View Prototype in OIPA

1. Log in to OIPA using credentials for a user belonging to the Holding Company.
2. In the top menu, select **Case > Search Case**.
3. Enter search criteria for the desired case record. Minimally, enter a wildcard symbol (%) in the Case Number field.
4. Click **Find**.
5. Click on a case record from the Case Search Results.
6. Note that the fields configured in the XML Source pane of the CaseScreen business rule display.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Client Screen
Change Activity

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Client Screen Change Activity

OIPA provides the ability to generate activities based on updates to Client records. Spawn functionality was added to the ClientScreen business rule to support this enhancement.

Refer to the XML Configuration Guide for a complete explanation of the elements and attributes supported in the ClientScreen business rule.

Configuration Required

The following configuration must be in place to allow updates made to client records to spawn client level activities.

- The ClientScreen rule must be updated with Action and Events logic, including a spawn section, for each client type where spawns apply. An explanation of the necessary Action and Events logic is provided below.
 - <Events> section
 - The Event Type to provide Spawn functionality must be TYPE="ONSUBMIT"
 - The Event section can call any of the following: Math, ActionSet, QuerySet and Spawns.
 - <Math> section
 - Use this section if logic is to be processed in a non-global (local) ScreenMath. The tag should say Global="NO".
 - The ID in Math (<Math ID="name"> identifies the ScreenMath to be processed.

Example: the logic that might be used with Spawns would be to identify if any fields have been changed. Math variables may be set in ScreenMath to then trigger a conditional Spawn.

- <ActionSet> section
 - Use this section to provide warning or error messages, set field values, hide/reveal fields or disable/enable fields.
Example: the logic that might be used with Spawns is to provide a warning to the user if a field change has caused a client level transaction to be generated.
- <QuerySet> section
 - Use this section if there is a need to repopulate a comboBox or to repopulate a field.
Example: the logic that might be used with Spawns is to provide changed information needed for a Spawn.
- <Spawns> section

- Use this section to generate a client level transaction. This should be set as a conditional Spawn to prevent a client level activity from always generating. Make sure the transaction identified already exists as a client level transaction.
- Only SPAWNCODE="03" (spawn based on date) may be used. Therefore, the configuration must include setting a date to spawn the client level transaction.
- Client level transactions must be created, which will spawn when the spawn conditions are met.

Prototype Explanation

The following business rule and transactions were configured in the Functional Prototype Plan to demonstrate this new functionality.

Business Rules

- The **ClientScreen** business rule was created with all the Action and Events configuration in place to demonstrate how a change to a client field can cause an activity to be spawned. The important parts of configuration are explained below. This rule must be overridden at the Primary Company level. to view configuration, navigate in the Global Explorer to **Business Rules | Screen | ClientScreen | Company Overrides | ClientScreen (Prototype Company)**.
 - Changes made to the following Client screen fields for an Individual Client (client type 02) will trigger transactions. Trigger is caused by a field change notification when the “Save” button is selected. No transactions are triggered/spawned on initial save of the client.
 - Last Name
 - Tax ID
 - InitialTaxIDSource
 - DateofBirth
 - DateofDeath
 - <Events> section provides a wrapper for processing Math, ActionSet and Spawns.
 - <Math> section identifies if this is the initial entry of the client (use of ClientGUID). It also is used to identify if the above mentioned fields have changed and provides a date to use for the transaction spawn.
 - <ActionSet> section is used to provide a warning that fields have changed and to notify the user of the transaction that was spawned.
 - <Spawns> section identifies the conditions when a transaction should be spawned. Fields from the Client screen are compared to math variables from ScreenMath to determine if the field value changed. If a

change occurred, then transaction will be spawned.

Transactions

There are four client level transactions configured to demonstrate the use of a Spawn section in the Client screen. Navigate in the Main Explorer to **Companies | Prototype Company | Plans | Client Plan | Transactions| Client Level Transactions** to view the configuration for each.

- **ClientInfoChangeLetter1:** this transaction is a stub, only meant to show that a transaction may be spawned and in itself provides no additional functionality. This letter is spawned when a change is made to the client's last name.
- **ClientInfoChangeLetter2:** this transaction is a stub, only meant to show that a transaction may be spawned and in itself provides no additional functionality. This letter is spawned when a change is made to the TaxID or TaxID Source.
- **ClientInfoChangeLetter3:** this transaction is a stub, only meant to show that a transaction may be spawned and in itself provides no additional functionality. This letter is spawned when a change is made to the Date of Birth.
- **ClientInfoChangeLetter4:** this transaction is a stub, only meant to show that a transaction may be spawned and in itself provides no additional functionality. This letter is spawned when a Date of Death is added or changed.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password. Make sure the user has privileges to view the Client screen and client activities.
2. Create a new Individual client and save the initial data.
3. Make a change to any or all of the following fields:
 - Last Name
 - Tax ID
 - TaxIDSource
 - DateofBirth
 - DateofDeath
4. You have the option of typing a value into the field named, **Date To Send Correspondence**.
 - If you fill in a value for this date, that is what you will see on any activities that are spawned. If you do not fill in this date, the system date will be used.
 - If you use this date, be sure to type a date that is the same as or earlier than the system date.
5. Click **Save**. At the top of the screen a warning message will display alerting you that information has changed and letters were generated.
6. Click the **Activities** link in the Left Navigation menu on the Client screen. The Client Activity screen should show pending activities. The correlation between change and generated activity is shown below.
 - Last Name change: ClientInfoChangeLetter1
 - Tax ID or TaxIDSource change: ClientInfoChangeLetter2
 - Date of Birth change: ClientInfoChangeLetter3
 - Date of Death change: ClientInfoChangeLetter4
7. Navigate back to the Client screen by clicking the **Client** link in the Left Navigation menu.
8. Make an update to a field that is not one of the fields identified above.
9. Click **Save**. No warnings should display.

10. Navigate back to the Client Activity screen. No activities should be added.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Comment Template
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Comment Template Prototype

The comment template feature allows generic comments to be created that can then be applied to various records in OIPA. Configuration determines if comments are created at the Policy, Segment, Activity, Suspense or Client level. Security controls the View, Add, Update, Delete, or View History actions, to limit users to only their own comments or grant users access to all comments.

Configuration Required

The following configuration must be in place to support comment templates.

- The **CommentScreen** rule must contain a <UseTemplate> element, which is set to Yes. This element can appear in any of the comment sections and can be repeated as many times as needed to turn on template support for that section. The Client Comment Screen uses the DefaultComments configuration.
- The **CommentSearchScreen** rule must be configured to support a user's ability to search for existing comments.
- **AsCodeCommentType** must be populated with available types of comments.
- **Administration - All Non-Security Administration CheckIn/CheckOut** privilege must be added to users who want to work with templates.
- OIPA security groups should be configured to control comment viewing and access for CSRs in OIPA.
- Comment Templates, if used, are created through the Admin Explorer **Comment Template** node.

Prototype Explanation

The following business rules were configured in the Functional Prototype Plan to demonstrate this new functionality.

Business Rules

- The **CommentScreen** business rule was created to support comment template configuration. The important parts of configuration are explained below. To view configuration, navigate in the Global Explorer to **Business Rules | Screen | CommentScreen** .
 - <UseTemplate> element appears in all sections to support the use of templates in all supported areas.
 - fixed and dynamic fields are present in the rule configuration as well as specific fields by Comment Type (ex, Policy Comment has Policy Comment Field).
 - Action/Events configuration shows an example of OnLoad (populating field value), OnChange (showing hidden field), and OnSubmit (showing Error message).
 - Dynamic fields pull in values from AsCode.

- The **CommentSearchScreen** business rule was created to define search criteria and the display of results. The important parts of configuration are explained below. To view configuration, navigate in the Global Explorer to **Business Rules | Screen | CommentSearchScreen** .
 - <UseTemplate> element appears in all sections to support the use of templates in all supported areas.
 - fixed and dynamic fields are present in the rule configuration as well as specific fields by Comment Type (ex, Policy Comment has Policy Comment Field).
 - Action/Events configuration shows an example of OnLoad (populating field value), OnChange (showing hidden field), and OnSubmit

(showing Error message).

- Dynamic fields pull in values from AsCode.

- Comment Templates were created to demonstrate the various levels where templates can be applied. To view configuration, navigate in the Admin Explorer to **Administration | Comment Templates** .
 - **Global Template**: there is one global template in this folder. It should be available for all types of comments.
 - **Primary Company Template**: there is one primary company template in this folder. It should be available to client and suspense comments under that company.
 - **Subsidiary Company Template**: there is one subsidiary company template in this folder. It should be available to policy, segment and activity comments under that subsidiary company.
 - **Product Template**: there is one Product template in this folder. It will only display if Product functionality is turned on for the environment and it should be available to policies, segments and activities under that Product.
 - **Plan Template**: there is one plan template in this folder. It should be available to policies, segments and activities under that plan.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password. Make sure the user has privileges to view comments.
2. Open a policy.
3. Click the **Add Comments** link on the Secondary menu of the Policy screen. This will open the Comments screen.
4. Select the comment details needed using the checkboxes and drop down boxes, then enter the comment text.
5. Click **Save**.
6. Close the window when finished.
7. Click the **Comments** link on the Left Navigation menu. This will bring up the Comment Search screen.
8. Enter any search criteria and search for the comment you just created.
9. Click **Refresh** to view the search results.
10. Close the window when finished.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Case Search Screen
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Case Search Screen Prototype

OIPA now supports a Case Search screen, which allows a user to create and edit case records as part of the New Business Underwriting process.

Prototype Explanation

The following business rules were configured as global rules to demonstrate this new functionality.

Business Rules

- **CaseSearchScreen:** This business rule was configured to display a set of predefined fields when a case record is selected. To view the prototype configuration, navigate in the Global Rules Explorer to **Business Rules | Screen Rules | CaseSearchScreen** and select the XML Source pane. The key configuration for this business rule is explained below.
 - Four fixed fields are configured: UpdatedDateRangeFrom, UpdatedDateRangeTo, CreatedDateRangeFrom and CreatedDateRangeTo.
 - A drop-down field called StatusCode is configured, which pulls the values for selection from AsCodeCaseStatus.
 - The table that displays the case records is configured with Case Number, Case Name, Application Number, Case Status, Insured First Name, Insured Last Name and Last Updated columns.

View Prototype in OIPA

1. Log in to OIPA.
2. In the top menu, select **Case > Search Case**.
3. Note that the fields configured in the XML Source pane of the CaseSearchScreen business rule display.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases >
CreateAdditionalRates Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

CreateAdditionalRates Prototype

OIPA now supports the ability to generate a set of new date-based rates using an activity. These rates will be added to an existing rate table, as long as the rate table contains only one rate group.

Configuration Required

The following configuration must be in place to support the addition of date-based rates using an activity:

- The CreateAdditionalRates rate group must exist with at least one date-driven rate.

Prototype Explanation

The following business rules and transactions were configured in the System Plan, located within the Prototype Company, to demonstrate this new functionality.

Transactions

- **PlanLevelCreateAdditionalRates:** This transaction allows the user to add new rates to a specific rate group that contains date-driven rates. It fetches the last applicable rate for the most recent date for the selected rate group, then multiplies this retrieved rate with the Rate Factor entered by the user. The resulting value is then stored in a collection math variable. Upon processing this activity, the generated rate will be added to AsRate. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | System Plan | Transactions | PlanLevelCreateAdditionalRates** and click on the **XML Source** pane. The key configuration for this transaction is explained below.
 - The RateEffectiveDate field allows the user to enter the effective starting date for the rate.
 - The RateFactor field allows the user to enter the multiplier to calculate the new rate based on the original rate.

Business Rules

- **CreateAdditionalRates:** This business rule adds the new rate to AsRate with the Date Criteria specified by the user. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | System Plan | Business Rules | Attached Rules | CreateAdditionalRates** and click on the **XML Source** pane. The key configuration for this business rule is explained below.
 - The <RateGroup> element encloses the configuration required for the creation of a new set of rates.
 - The <RateDescription> element specifies the name of the math

variable--RateDescriptionMV in this prototype configuration--that holds the rate description.

- The <CreateRate> element encloses the configuration required for adding a new record to the rate group and AsRate with the specified rate description. It can be used multiple times to create multiple records in AsRate. The element has several child elements:
 - The <DateCriteria> element contains a date value used to add to the DateCriteria column in the AsRate table.
 - The <Criteria> element contains additional criteria values that need to be considered when adding the new rate. It contains <Criterion> child elements that specify the name and value of a single criterion.
 - The <RateCollection> element specifies the name of the collection math variable that contains the new rate value, which is mapped to the Integer Criteria as the key.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password.
2. Select **Tables > Rates** from the main menu at the top of the screen.
3. Enter "CreateAdditionalRates" into the **Rate Group** field and click **Find**.
4. Select one of the rows that appears in the **CREATEADDITIONALRATES** section of the screen.
5. Enter criteria of your choosing in the **RATECRITERIA** section of the screen and click **Filter**.
6. The search results will appear in the **RATE SEARCH RESULTS** section of the screen. Make a note of the rates that correspond to a given date criterion.
7. Navigate to the Company Activity screen.
8. Add the PlanLevelCreateAdditionalRates activity.
9. Enter information in the Rate Effective Date and Rate Factor fields.
10. Process the PlanLevelCreateAdditionalRates activity.
11. View the updated rate for the previously noted date criterion.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > CreateRates
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

CreateRates Prototype

Two components make up this prototype. First, a transaction was created that references an initial set of rates and then allows a user to enter an adjustment factor that is applied to those rates to create new rate values. The CreateRates business rule was attached to the transaction and is used to insert a new rate group and rate records into the database. The CreateRates rule may be attached to Plan level transactions under a Primary or Subsidiary Company.

Scenario

A CSR adds the insurance rate activity to a plan or company in OIPA. He enters a new effective date (EffectiveDate in AsRateGroup) for the rate and a date when the rates will be available to activities (ActiveToDate in AsRateGroup). An adjustment factor will be applied to the existing rates to create new rate values. The adjustment factor may be a positive or negative decimal value. When the activity is processed, the new rates are created and rates may be expired based on the rate effective date the user has entered. Even when expired, rates are not shadowed as they are still effective for back dating during the time the rate was active.

Configuration Examples

- A set of rates have been initially created for use by the new transaction. The rate description is CreateRates. The initial Effective and Active to Dates are 1/1/2000. There is one criterion, Gender, with two possible values: Male (01), Female (02). There are six rates for each gender.
- A **CompanyLevelCreateRates** transaction was created on the plan of a Primary company. The user enters a new rate effective date (EffectiveDate in AsRateGroup) for the rate and a date when first available to transactions (ActiveToDate in AsRateGroup). An adjustment factor will be applied to the existing rates to create new rate values. The adjustment factor may be a positive or negative decimal value. Navigate to **Global Explorer | Transactions | CompanyLevelCreateRates**

View New Feature in OIPA

To view the prototype example in OIPA:

1. Login and select the Prototype Company and Functional Prototype plan.
2. Create a new policy, add roles, and add at least one segment.
3. Click the **Activities** link to open the Activity screen.
4. Click **Add Activity** on the Secondary menu and select the **CompanyLevelCreateRates** activity.
5. Enter new rate effective date for the rate and a date when first available to activities.
6. Enter an adjustment factor (positive or negative decimal value).
7. Click **OK** to add the activity to the Activity screen.
8. Click the lightning bolt icon to process the activity.
9. Click the **Tables** menu in the Main menu.
10. Click the **Rates** option.
11. Search for the new rates that were just added. They should display in the Rates table.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > CreateSegment
Business Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

CreateSegments Business Rule

The CreateSegments business rule allows an activity in OIPA to add a segment to a policy. This business rule can be attached to a transaction, then once the transaction is added to a policy and processed as an activity in OIPA, a segment (or coverage) is added to the policy.

As a part of this enhancement, all SegmentFields with a TYPE of Money are assigned a default CurrencyCode. The AsPlan table has a new column named DefaultCurrencyCode and the AsSegmentField table has a new column named CurrencyCode. The currency value at the segment level will be the Plan default unless it is overridden in AsSegmentField.

Scenario

A CSR needs to add a coverage to an existing policy. The ContractAddCoverage activity is added to the policy and processed. A Base Coverage segment is added to the policy.

Configuration Requirements

A segment can be added to an existing policy if the following items are configured:

- a transaction configured to add a segment
- the CreateSegments business rule overridden at the transaction level

Prototype Samples

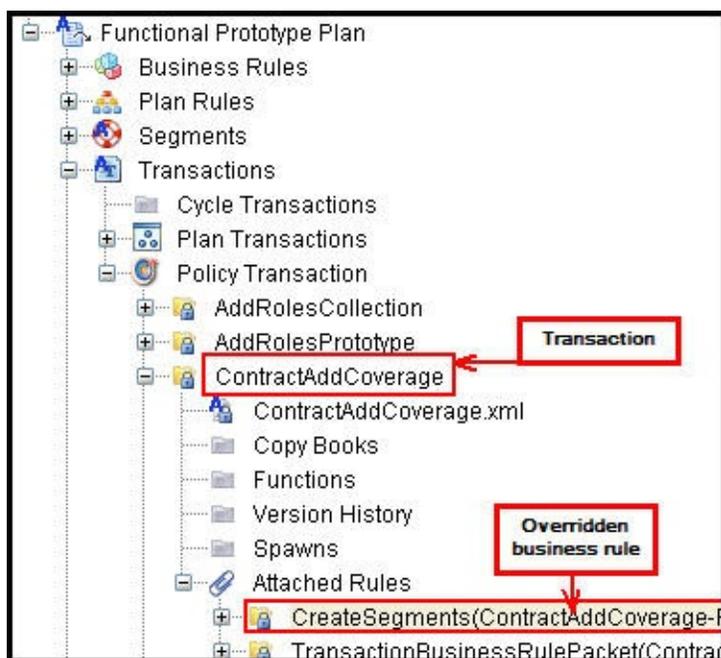
There is one transaction created to demonstrate this functionality:

- **ContractAddCoverage:** This transaction is configured to add the Base Coverage segment to an existing policy. The CreateSegments business rule is overridden at the transaction level. Currency is commented out at this time, but will be an enhancement in a future release.

There is one business rule configured to demonstrate this functionality:

- **CreateSegments (ContractAddCoverage-Functional Prototype Plan):** This business rule was overridden at the transaction level and attached to the transaction in the TransactionBusinessRulePacket.

To see both configuration samples, navigate through the following folders in the Main Explorer: **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | ContractAddCoverage.**



CreateSegment Transaction and Business Rule in Main Explorer

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Create Unique Identifier Number

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create Unique Identifier Number

The Identifier field data type can be used when a company needs to generate an ID that is built from characters and/or numeric parts. The numeric parts may be incremented and therefore provide a unique characteristic to the IDs constructed with the same elements. An example would be the generation of a claim number for a death claim process that is unique among other claim numbers.

Identifier fields can also be called in transaction configuration. A transaction can contain a math variable that is configured to pull a value from an Identifier field, then pass that number to a spawned transaction. The scenarios below provide examples of the various ways Identifier fields are used in OIPA.

 The new Activity Confirmation Screen uses [Identifier fields](#) and [Identifier math variables](#) to meet specific requirements for unique confirmation numbers for activities and spawned activities. For more information, see [Activity Confirmation Screen](#).

Scenario 1:

Use the Identifier field type to create a field on a screen where the Identifier number will be used. When a new policy is created and saved for the first time, this identifier number will populate the Identifier field on the screen. For example, the Policy screen may have an Identifier field to capture the unique number. When a new policy is saved, the field is automatically populated with the unique ID.

Scenario 2:

- Create a parent transaction that uses a math variable to pull a value from an identifier field. When the transaction processes as an activity in OIPA, the Identifier fields configured in the transaction are populated and the value of the math variable is

listed in the Math link of the Activity Detail window.

- Use the MultiFields business rule to set the number of identifier fields that should be populated when the activity processes.
- Create a child transaction that is spawned from the first transaction and pulls a value from an identifier field of the original transaction. When the transaction processes as an activity in OIPA, the value of the math variable is listed in the Math link of the Activity Detail window.

Scenario 1

A configurator creates a field on the Policy screen that will generate an ID. When the policy is created and saved for the first time, the identifier number will automatically populate the field on the Policy screen.

Add an Identifier Field to the Policy Screen

1. Open a **Company | Plan** in the Main Explorer.
2. Open the **Plan Rules** folder and check out the PolicyScreen business rule.
3. Click the **Fields** pane and drag an Identifier field onto the Configuration area.
4. Click the new field. This will open the [Field Properties](#) Window.
5. Enter a Field name and Display name.
6. Scroll down to the Parts section and click the ellipsis button next to Parts. This will open the Parts window.
7. Select a Part type from the Type drop down box.
8. Enter the [corresponding information](#) for the type of Identifier chosen.

 To generate a unique Identifier number, the Identifier field must contain a SEQUENCE Part type.

9. Click **Add**. The information will be listed in the Parts tree at the top of the window.
10. Click **OK** when all Identifier information has been listed.
11. Save the changes by clicking the **Save** button on the menu bar.

Prototype Samples in Rules Palette

The PolicyScreen business rule was overridden and configured to demonstrate the identifier field. Navigate through the following folders in the Main Explorer to locate the configuration sample: **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Dynamic Prototype Plan | Plan Rules | PolicyScreen**. Open the PolicyScreen and click the Fields pane. The identifier field is there and the properties can be viewed from the Field Properties Window.

View New Feature in OIPA

View this configuration in OIPA by logging into the **Company | Plan** you identified in the steps above. Create a new policy. Add policy details and click the **Save** button. The identifier field should be disabled and should show the unique ID generated by the system.

The screenshot displays a web form titled "POLICY" and "POLICY DETAIL". The form contains the following fields and values:

- Company: Prototype Child Company
- Plan: Dynamic Prototype Company
- Plan Date: 1/29/2009
- Issue State: ID
- Policy Name: Identifier
- Policy Number: (empty)
- Policy Status: Pending
- Interest Bonus Qualified: No
- Policy Identifier: 0000000002 (highlighted with a red box)

A "Save" button is located at the bottom right of the form.

Identifier Field on Policy Screen

Scenario 2

A configurator creates a transaction that uses a math variable to pull a value from an Identifier field. This transaction also references the MultiField business rule and spawns a second Identifier transaction. Create the spawned transaction first so that it can be referenced in the main transaction.

Create the Spawned Transaction

1. Open the **Company | Plan** in the Main Explorer for the new transaction.
2. Right-click on the Transaction node and select New Transaction.
3. Select the Transaction Type from the drop down box.
4. Enter a Processing Order. It should be after the processing order of the transaction that will spawn this one.
5. Click **Finish**. The new transaction will display in the Configuration area.
6. Add a [text](#) field to the Fields pane.
7. Click the Math pane.
8. Drag a Field math variable onto the screen.
9. Enter a name for the math variable.
10. Scroll down to the Prefixes field, click in the field then hold down CTRL and press the space bar. This will bring up a list of all available Prefixes. Select **Activity**.

 If you cannot see the Prefix field, then grab the horizontal scroll bar at the bottom of the Math Pane section and drag it to the right. A vertical scroll bar will appear. Drag the vertical scroll bar down until the fields are visible.
11. Scroll down to the Value field, click in the field then hold down CTRL and press the space bar. This will bring up a list of available values. Select the value you want to use.
12. Check in the transaction to save the configuration.
13. Add a translation value when the message window appears.
14. Add [security](#) for the transaction.
15. Add the transaction to the [EligibleTransactionByPolicyStatus](#) business rule.

Create the Parent Transaction

1. Open the **Company | Plan** in the Main Explorer for the new transaction.
2. Right-click on the Transaction node and select New Transaction.
3. Select the Transaction Type from the drop down box.
4. Enter a Processing Order. It should be before the processing order of the spawned transaction created above.
5. Click **Finish**. The new transaction will display in the Configuration area.
6. Click the Fields pane.
7. Drag an Identifier field onto the screen.
8. Click the new field. This will open the [Field Properties](#) Window.
9. Enter a Field name and Display name.
10. Set the Clear on Recycle element according to the way the identifier number should be handled if the activity is recycled:
 - To keep the same confirmation number for the recycled activity, set it to No.
 - To generate a new confirmation number each time the activity is recycled, set it to Yes.
11. Scroll down to the Parts section and click the ellipsis button next to Parts. This will open the Parts window.
12. Select a Part type from the Type drop down box.
13. Enter the [corresponding information](#) for the type of Identifier chosen.

 To generate a unique Identifier number, the Identifier field must contain a SEQUENCE Part type.
14. Click **Add**. The information will be listed in the Parts tree at the top of the window.
15. Click **OK** when all Identifier information has been listed.
16. Save the changes by clicking the **Save** button on the menu bar.
17. Drag a Multifields element onto the screen and click on it.

18. Click the ellipsis button on the Multifield line in the Field Properties window. A Multifield window opens.
19. Select the Multifield rule to associate with this Multifields element and click OK.
20. Click on the Math pane.
21. Drag an Identifier math variable onto the screen.
22. Enter a name for the Identifier math variable.
23. Scroll down to the Parts section and click the Parts node.
24. Select a Part type from the Type drop down box.
25. Enter the [corresponding information](#) for the type of Identifier chosen.

 To generate a unique Identifier number, the Identifier field must contain a SEQUENCE Part type.

26. Click **Add**. The information will be listed in the Parts tree below the Parts node.
27. Click on the Spawn pane.
28. Select the child transaction from the Transactions list and click Add.
29. Configure the Spawn Fields to map the Identifier math variable in the parent to the text field in the child transaction.
30. Check in the parent transaction.
31. Add a translation value when the message window appears.
32. Add [security](#) for the transaction.
33. Add the transaction to the [EligibleTransactionByPolicyStatus](#) business rule.

Prototype Samples in Rules Palette

There are two transactions configured to demonstrate identifier field functionality. Navigate through the following folders in the Main Explorer to locate the configuration sample: **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Dynamic Prototype Plan | Transactions | Policy Transactions**.

- **IdentifierFields**: uses a mathvariable to pull a value from an identifier field. When the transaction processes as an activity in OIPA, the Identifier fields configured in the transaction are populated and the value of the mathvariable is listed in the Math link of the Activity Detail window. This transaction also uses the MultiFields business rule to set the number of identifier fields that should be populated when the activity processes.
- **IdentifierSpawn**: spawned from the first transaction and pulls a value from an identifier field of the original transaction. When the transaction processes as an activity in OIPA, the value of the mathvariable is listed in the Math link of the Activity Detail window.
- The following sequences have been added to the AsSequence table for use in generating identifiers:
 - ClientID
 - EffectiveDateID
 - MultiFieldID
 - PolicyFieldID

View New Feature in OIPA

View the prototype example in OIPA by logging in and selecting the **Company | Plan** you used when creating the screen and transactions above. Create a new policy and click the Activities link. Select the Identifier Fields transaction first and click **OK**. Process it from the Activity screen. The Identifier Spawn activity will appear on the Activity screen. Process it. After it processes, click the Activity Detail icon to the left of the activity. Check the Entry Fields link and the Math link.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Currency Code Validation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Currency Code Validation

Before writing to the Target Field, OIPA will confirm that the currency code associated with the source value exists in the Currency element of the Target Field. If the Target Field has no defined currency, the default currency for the plan or company is assumed.

Business errors are generated if the Source Currency code is not found in the Currency element of the Target Field, or if the Target Field is not a Money data type.

Configuration Requirements

There are 16 distinct rules and one set of elements affected by currency validation functionality. The affected rules may be broken into the following four categories:

1. CopyTo (update existing fields) rules attached to transactions
2. Create / Add / Generate (insert new fields) rules attached to transactions
3. Transaction elements that update existing fields
4. Rules that are imbedded into Screen business rules, and create new, or update existing fields

For the purposes of demonstrating the currency validation feature, six rules/elements were selected from the four categories above.

Business Rules

- **CopyToPolicyFields:** This rule is attached to the Premium transaction in the International Holding Company in the Unit Linked template. Target field is LastPremiumAmount in PolicyScreen business rule in the Unit Linked template.

Navigate to **Main Explorer | International Holding Company | Subsidiary Companies | International Child Company | Plans | Unit Linked Template | Transactions | Policy Transactions | Premium | Attached Rules | CopyToPolicyFields.**

- **MaintainSuspense:** This rule can be found in the Prototype Company in the Functional Prototype Plan. The target field is the PremiumAmount in the SuspenseScreen business rule in the Functional Prototype Plan.

Navigate to **Main Explorer | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | MaintainSuspense | Attached Rules | MaintainSuspense-(FunctionalPrototypePlan).**

- **CopyToSegmentFields:** This transaction can be found in the Prototype Company in the Functional Prototype Plan under Policy Transactions. The target field is the SegmentAmount in the Base Coverage segment in the Functional Prototype Plan.

Navigate to **Main Explorer | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | CopyToSegmentFields | Attached Rules | CopyToSegmentFields-(FunctionalPrototypePlan).**

- **CalculateGeneral:** This rule can be found in the Global Rules

explorer under **Calculate | CalculateGenearlLifeBaseCov | Plan Overrides for Guaranteed Level Premium Term**. The target fields are the money fields in the Guaranteed Level Premium Term Base Coverage segment and PolicyScreen business rule.

Transactions

- **AddRolesPrototype**: This transaction can be found in the Prototype Company in the Functional Prototype Plan. The target field is the Amount field in the **Cont Bene** Role Screen business rule in the Functional Prototype Plan.
- **AddRolesCollection**: This transaction can be found in the Prototype Company in the Functional Prototype Plan. The target field is the Amount field in the **Cont Bene** Role Screen business rule in the Functional Prototype Plan.

Elements

- **Disbursement**: This element can be found in the **Holding Company | Primary Company** in the Variable Deferred Annuity plan. The Target fields are TotalFederalWithholding and TotalStateWithholding in the Variable Deferred Annuity plan.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Currency Exchange

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Currency Exchange

Currency Exchange functionality was enhanced to replace the existing NUVOffset rule with a defined MathStatement (which operates much like a Function) to return the GuaranteedExchangeDate, GainLossExchangeDate, and an indicator that exchange rates exist on the calculated date (RatesExist). An input offset integer will tell the MathStatement how many business days to add to the currency activity effective date.

The exchange of one currency to the next is done using the currency exchange rates for the Market Maker record attached to the Plan or Company. The base valuation currency is defined at the Plan level and all amounts passed into Activity assignment processing are presumed to be in this base currency. The system will apply the Market Maker's exchange rate as of the Guaranteed Exchange Date calculated by the configured MathStatement to convert from the base currency to the target currency of each fund in the activity allocation. Each currency will have its own exchange rate but the rates used will all share the same effective date as calculated by the MathStatement.

The exchange date calculated by the MathStatement will always be driven by the activity effective date plus any applicable offset business days based on the Plan Market Maker's calendar. Activity assignment will then find the guaranteed Price Date for each allocated fund based on that fund's calendar and business day offset record. Each fund's offset is applied to the Guaranteed Exchange Date based on the fund's calendar.

In some transactions it may be necessary to use the last known exchange rates or prices for the valuation and assignment processes. In these cases, the MathStatement used has no input offsets and returns the last known exchange date for the Market Maker as of the activity effective date. Additionally, the activity assignment will use the last known prices of the allocated funds for valuation with no business day offsets.

```
Withdrawal(Unit Linked Template) x PostAssignment.ValidateExpressions(Withdrawal - Unit Linked Template) x
General Allocations Fields Events Math Assignment Spawn Xml Source Debug
</MathIF>
<MathVariable VARIABLENAME="TotalSurrenderValue" TYPE="EXPRESSION" DATATYPE="DECIMAL" ROUND="2">NetMoneyOut + SurrenderCharge</MathVari
<MathVariable VARIABLENAME="NegativeNetMoneyOut" TYPE="EXPRESSION" DATATYPE="DECIMAL" ROUND="2">NetMoneyOut * -1</MathVariable>
<MathVariable VARIABLENAME="NegativeSurrenderCharge" TYPE="EXPRESSION" DATATYPE="DECIMAL" ROUND="2">SurrenderCharge * -1</MathVariable>
<MathVariable VARIABLENAME="CurrentActivityEffectiveDate" TYPE="FIELD" DATATYPE="DATE">Activity:EffectiveDate</MathVariable>
<MathVariable VARIABLENAME="CurrentActivityEffectiveDate" TYPE="FIELD" DATATYPE="DATE">Activity:EffectiveDate</MathVariable>
<MathVariable VARIABLENAME="InputOffset" TYPE="VALUE" DATATYPE="INTEGER">0</MathVariable>
<MathVariable VARIABLENAME="CurrencyExchangeDate" TYPE="FIELD" DATATYPE="DATE">Activity:EffectiveDate</MathVariable>
<MathVariable VARIABLENAME="GainLossCurrencyExchangeDate" TYPE="FIELD" DATATYPE="DATE">Activity:EffectiveDate</MathVariable>
<MathVariable VARIABLENAME="RatesExist" TYPE="EXPRESSION" DATATYPE="BOOLEAN">>false</MathVariable>
<MathIF IF="Activity:PricingType = 'Forward'">
  <MathStatement TYPE="ACTIVITYFUNCTION" FUNCTIONNAME="FindNextExchangeDatePricing">
    <Parameters>
      <Parameter NAME="ExchangeDateOffset" TYPE="INPUT">InputOffset</Parameter>
      <Parameter NAME="GuaranteedExchangeDate" TYPE="OUTPUT">CurrencyExchangeDate</Parameter>
      <Parameter NAME="GainLossExchangeDate" TYPE="OUTPUT">GainLossCurrencyExchangeDate</Parameter>
      <Parameter NAME="RatesExist" TYPE="OUTPUT">RatesExist</Parameter>
    </Parameters>
  </MathStatement>
</MathIF>
<MathIF IF="Activity:PricingType = 'Last'">
  <MathStatement TYPE="ACTIVITYFUNCTION" FUNCTIONNAME="FindLastExchangeDatePricing">
    <Parameters>
      <Parameter NAME="GuaranteedExchangeDate" TYPE="OUTPUT">CurrencyExchangeDate</Parameter>
      <Parameter NAME="GainLossExchangeDate" TYPE="OUTPUT">GainLossCurrencyExchangeDate</Parameter>
    </Parameters>
  </MathStatement>
</MathIF>
</MathVariables>
<Assignment TYPE="Withdrawal" REMOVALMETHOD="Net" MONEYTYPE="32" SURRENDERCHARGERATE="PartialSurrenderPercentCharge" SURRENDERCHARGEMONEYTYPE="
<Assignment TYPE="Withdrawal" REMOVALMETHOD="Gross" MONEYTYPE="32" SURRENDERCHARGERATE="PartialSurrenderPercentCharge" SURRENDERCHARGEMONEYTYPE="
<Assignment TYPE="GrossWithdrawal" IF="MVWithdrawalType = 'Amount'">
  <MoneyType NAME="NegativeNetMoneyOut">32</MoneyType>
  <MoneyType NAME="NegativeSurrenderCharge">07</MoneyType>
</Assignment>
h>
nsitions>
  <Queue METHOD="VALUATION" ADVANCETOSYSTEMDATE="Yes"></Queue>
nsitions>
>
```

XML Example Showing the Call to MathStatement

Configuration Requirements

The number of offset days should be held at the company level and overridable down to the transaction level. The Market Maker calendar will use the offset integer to determine the number of business days from the activity's effective date to look for the guaranteed exchange date. The offset integer is a parameter passed into the Math Statement function.

Currency conversion will use only one exchange date for assignment (can be different than the activity effective date). The exchange date is validated by:

1. Confirming the activity effective date is a business day on the custodian bank's calendar.
2. If the activity effective date (day) is a business day, add the number of offset days to the date.
3. If the activity effective date (day) is not a business day, move the calendar to the next business day and add the number of offset days.

The default offset day(s) will be 0 if not specified in the Math. However, a transaction may need to determine the currency exchange offset to be used as of the time of day locally when the activity is entered.

TransactionTimes business rule can assist with cut off time frames.

Business Rules

- **PriceOffset:** This rule will manage the Net Asset Value lookup method called "NextPrice" and the implementation of exchange date and price offsets.
- **TransactionTimes:** This rule can be used in conjunction with the MathStatement for determining the number of offset day(s) when the activity effective date needs to be determined locally. Transactions added prior to a cutoff time may use (0) days while those added after may use a different number of offset day(s).
- **PostAssignmentValidateExpressions:** This rule can be used to pull values from Funds and confirm that the activity and/or accounting can go forward, the values in the funds meet minimum guidelines, or that funds have a sufficient amount for the activity being processed.

Navigate to **Main Explorer | International Company | Subsidiary Companies | International Child Company | Plans | Unit Linked Template | Transactions | Policy Transactions | Withdrawal | Attached Rules | PostAssignmentValidateExpressions-(UnitLinkedTemplate).**

Transactions

- **Withdrawal:** This transaction can be found in the International Child Company in the Unit Linked Template. The offset integer used within the MathStatement (FindNextExchangeDatePricing) in determining the GuaranteedExchangeDate and GainLossExchangeDate.
- **MonthlyProcessing:** This transaction can be found in the International Child Company in the Unit Linked Template. The Math Statement uses FindLastExchangeDatePricing in determining the GuaranteedExchangeDate and GainLossExchangeDate.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Currency With Events

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Currency With Events

Money and Currency data types are now supported in transaction events in the ScreenMath and Action sections. Screen Math performs arithmetic and comparison operations on currency values and Actions compares currency values to make sure that currency types match.

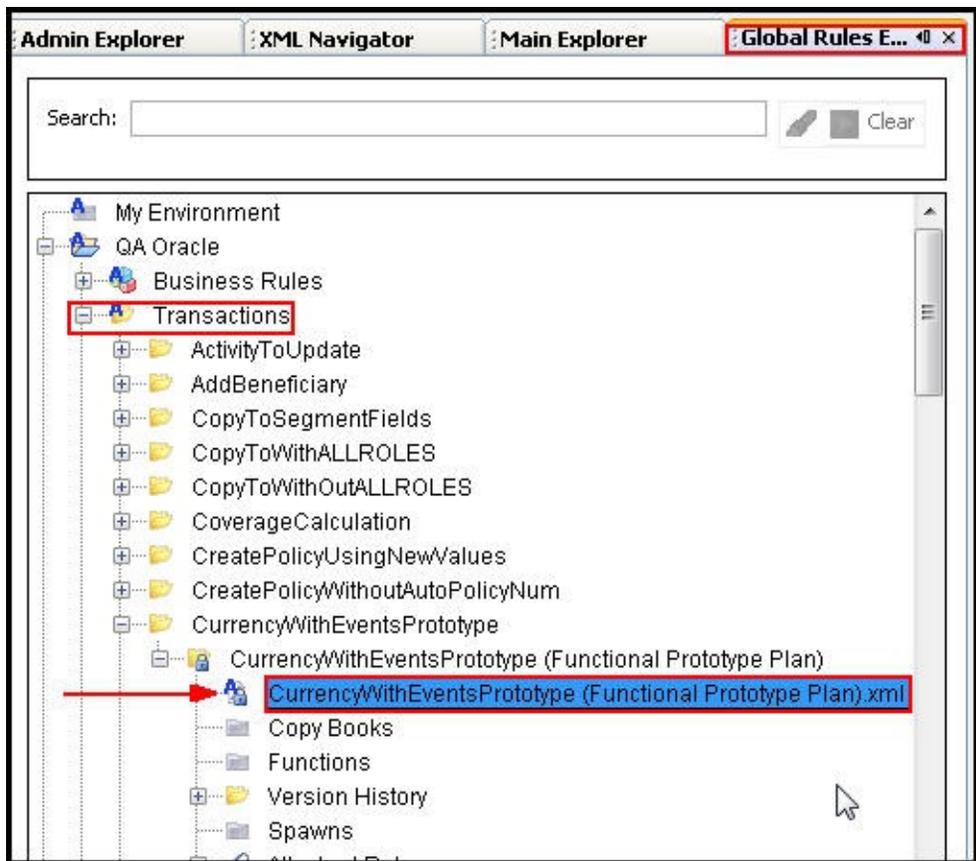
Scenario

A CSR processes a transaction that compares two currency values and validates that the currency types are the same.

Configuration Requirements

The following prototype samples are used to demonstrate this functionality.

- The **CurrencyWithEventsPrototype** transaction uses Screen Math to perform arithmetic and comparison operations on currency values. The Actions section compares currency values makes sure the currency types are the same. Navigate through the following folders in the Global Explorer to find the transaction: **Transactions | CurrencyWithEventsPrototype**.
 - The Math section performs the following:
 - Comparison of math variables in the Math section
 - Addition of like currency codes
 - Multiplication of like currency codes
 - The Actions section performs the following:
 - Comparison of money to money fields using less than expression
 - Comparison of currency math variables
 - Assign currency code of math variable to field
 - Assign value from one money field to another money field with like currency code
 - Assign value Currency <ScreenMath > MathVariable
 - Assign a math variable with a different currency code to a money field that allows that currency code
 - Tested for IsEmpty



Currency Prototype in Global Explorer

View Configuration in OIPA

To view the prototype example in OIPA:

1. Login and select the Prototype Company and Functional Prototype plan.
2. Create a new policy and add roles, and at least one segment.
3. Click the **Activities** link to open the Activity screen.
4. Click **Add Activity** on the Secondary menu and select the **Currency With Events Prototype** activity.
5. Enter an amount in each field.
6. Click **OK** to add the activity to the Activity screen.
7. Click the lightening bolt icon to process the activity.
8. Click the Activity Detail icon to the left of the processed activity. This opens the activity details.
9. Review the Math tab.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Original Currency Detail

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Original Currency Detail

OIPA stores the original currency code, currency amount, exchange rate effective date and exchange rate for activities and suspense. This new functionality includes configuration enhancements to allow this feature to be turned on or off and also allows a user to request what Market Maker should be used for the exchange. In addition the system records values used in these accounting currency exchanges for auditing purposes.

The ChartOfAccountsCriteria business rule is now renamed **ChartOfAccountsSpecifications** and a new element is supported that allows accounting records to be written in the default plan or company currency while retaining the information on the original currency amount entered.

The Unit Linked template uses a parent company override of the rule to write the original currency of the entered suspense record and captures the currency exchange factors (rates and dates) per the effective date and market maker entered on the Suspense screen.

The Unit Linked template Premium transaction has the COA set up to report on the Math Variable (MoneyIn). The transaction uses a ChartOfAccountsSpecifications child company override since that is the company the transaction is associated to. The accounting should capture the original currency of the MoneyIn math variable and the exchange factors (rates and dates) per the exchange effective date (math variable) calculated in the transaction and the default plan market maker.

Configuration Requirements

The following prototype samples are provided to demonstrate this functionality. They reside in the **Admin Explorer | Administration | Chart of Accounts** folder.

- **Suspense:** A suspense entity was created that uses a parent company override of the ChartofAccountSpecifications rule to write the original currency of the entered suspense record and capture the currency exchange factors (rates and dates) per the effective date and market maker entered on the screen.
- **Premium:** Premium transaction has the Chart of Accounts set up to report on the Math Variable (MoneyIn), formerly the Chart of Accounts was by fund. The transaction uses a ChartOfAccountsSpecifications child company override since that is at the company the transaction is associated to. The accounting should capture the original currency of the MoneyIn math variable and the exchange factors (rates and dates) per the exchange effective date (math variable) calculated in the transaction and the default plan market maker.

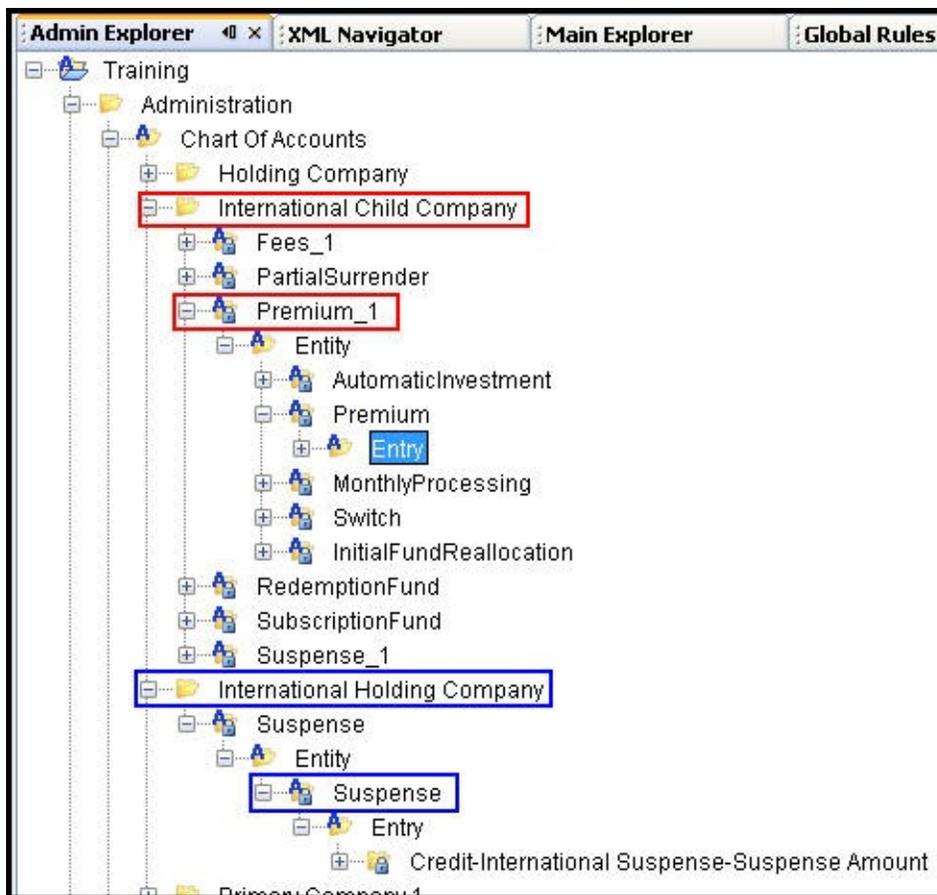


Chart of Accounts in Admin Explorer

- **ChartofAccountsSpecifications:** Overrides for this business rule are created at the company level (Parent company for suspense, Subsidiary company for transactions). **ExchangeToBaseCurrency** is given a value of **Yes**. If set to Yes, then the system will write the accounting detail in the base currency and write the original entry currency and its exchange detail to another table (AsAccountingDetailExchange). By default, the system will use a value of No and write accounting details in the entered currency. **MarketMaker** is an optional attribute. By default the plan or company MarketMaker will be used (if ExchangeToBaseCurrency is Yes). **EffectiveDate** is an optional attribute. By default, the suspense or activity effective date will be used (if ExchangeToBaseCurrency is Yes). Navigate to **Global Explorer | Business Rules | System | ChartofAccountsSpecifications | ChartofAccountsSpecifications (International Holding Company)**.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Use Specific Exchange Rate Date for Currency Exchange

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Use Specific Exchange Rate Date for Currency Exchange

The OIPA system can perform currency conversion inside of the math using a specified date to determine the exchange rate that should be used. The **ExchangeDate** attribute added to math variables of TYPE=CURRENCY should be configured to accept a field (Fixed or Dynamic) or math variable containing a date value, which will be used to determine the exchange rate used in the currency conversion.

Acceptable values include a field or math variable holding a date value. The specified date should allow for any date (past, present or future) as long as the rates that correspond to the configured date are present on the AsExchangeRate table for the currencies involved.

Configuration Requirements

- **MonthlyProcessing** transaction: This transaction contains a math variable called **SumAssuredInBaseCurrency** that converts the SegmentAmount field from USD to base currency (THB) using rates as of the specified effective date (Last Known exchange rate for the MarketMaker). Navigate to **Global Explorer | Transactions | Monthly Processing (Unit Linked Template)**.

```
<EffectiveDateNUVMustExist>No</EffectiveDateNUVMustExist>
<SystemDateNUVMustExist>No</SystemDateNUVMustExist>
</Valuation>
<Math>
  <MathVariables>
    <MathVariable VARIABLENAME="ActivityEffectiveDate" TYPE="
    <MathVariable VARIABLENAME="MonthlyProcessingSpawnDate"
    <MathVariable VARIABLENAME="CurrencyExchangeDate" TYPE="
    <MathVariable VARIABLENAME="GainLossCurrencyExchangeDate"
    <MathStatement TYPE="ACTIVITYFUNCTION" FUNCTIONNAME="F
      <Parameters>
        <Parameter NAME="GuaranteedExchangeDate"
        <Parameter NAME="GainLossExchangeDate"
      </Parameters>
    </MathStatement>
    <MathLoop TYPE="SEGMENT" VARIABLENAME="BaseLoop" POLICY
    <MathVariable VARIABLENAME="TypeCode" TYPE="SEG
    <MathIF IF="TypeCode = '04'">
      <MathVariable VARIABLENAME="SegmentGUID"
      <MathVariable VARIABLENAME="SumAssured"
      <MathVariable VARIABLENAME="SumAssuredC
      <MathVariable VARIABLENAME="Tobacco" TY
    </MathIF>
    </MathLoop>
    <MathVariable VARIABLENAME="PolicyCashValue" TYPE="EXPP
    <MathVariable VARIABLENAME="SumAssuredInBaseCurrency"
    <MathVariable VARIABLENAME="PolicyCashValueInSumAssured
    <MathVariable VARIABLENAME="DBOut" TYPE="POLICYFIELD" >
```

MonthlyProcessing Transaction in Global Explorer

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases >
DisbursementUpdate Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

DisbursementUpdate Prototype

OIPA allows certain activities to update the status of disbursement records to an active status by configuring the records that will be selected for update. Disbursement accounting can be triggered through the processing of this update activity as well.

Accounting on Suspense field values allows the accounting of amounts in dynamic fields on the Suspense Screen when the screen is saved.

Scenario

Two activities are processed that generate pending disbursements for a particular business day. An UpdatePendingDisbursement activity is pending, which was spawned from the previous cycle. The nightly batch cycle is run. The disbursement records are selected and the disbursement status is updated to active. Accounting records are generated for each disbursement.

Prototype Examples

There are two transactions, one business rule, five CoA accounts and two CoA Entities that were configured to demonstrate this functionality.

- The **DisbursementUpdate transaction** was developed in the Functional Prototype Plan using the existing policy level transaction PrototypeDisbursement as the source activity for the disbursement records. Navigate to **Global Explorer | Transactions | DisbursementUpdate (Functional Prototype Plan)**.
 - The [DisbursementUpdate](#) rule was attached to the transaction to demonstrate the use of the <UpdateStatus> attribute ACTIVITYARRAY. The <UpdateStatus> attribute STATUS with a value of Active (i.e., the only valid value), was used to set the disbursement status to active following successful execution of the rule. The rule element <COAEntity/> points to the Chart of Accounts entity defining the accounting. Navigate to the **Attached Rules** folder under the transaction to see the rule XML.
- The **DisbursementUpdateClient** transaction was developed in the SystemPlan using the existing client level transaction ClientLevelDisbursement as the source activity for the client disbursement records. Navigate to **Global Explorer | Transactions | DisbursementUpdateClient (System Plan)**.
 - The [DisbursementUpdate](#) rule was attached to the transaction to demonstrate the use of the <UpdateStatus> attribute DISBURSEMENTARRAY. The <UpdateStatus> attribute STATUS with a value of Active (i.e., the only valid value), was used to set the disbursement status to active following successful execution of the rule. The rule element <COAEntity/> points to the Chart of Accounts entity defining the accounting. Navigate to the **Attached Rules** folder under the transaction to see the rule XML.
- Under the Primary Company **Prototype Company**, Chart of Accounts were established for each status needed for disbursements. Navigate to **Admin Explorer | Administration |**

Chart of Accounts | Prototype Company.

- **Disbursement_01** with a description of Active Disbursement
 - **Disbursement_02** with a description of Pending Disbursement
 - **Disbursement_12** with a description of Shadow Disbursement
 - **Disbursement_27** with a description of Recovered Disbursement
 - **Disbursement_44** with a description of Recovered Shadow Disbursement
- Two Chart of Account Entities were created for each account: PrototypeDisbursement and ClientLevelDisbursement . Navigate to **Admin Explorer | Administration | Chart of Accounts | Prototype Company | Disbursement_01 | Entity** . Repeat for each status listed in the bullets above.
 - The ChartOfAccount entry is created for the dynamic field 'PremiumAmount' from the suspense screen.

Result: When the user saves with a value for the dynamic field to which COA is linked in the SuspenseScreen, user should be able to see the entry in AccountingDetailScreen.

View in OIPA

1. Log on to the Prototype Company in OIPA.
2. Click **Company** and **Company Activity** on the Main menu.
3. Select **System Plan** from the Plan drop down box.
4. Click the **Add Activity** link on the Secondary menu and select the **DisbursementUpdateClient** activity from the drop down box.
5. Enter an effective date and click **OK**.
6. Click **Activity** from the Left Navigation menu. This will open the Activity screen.
7. Click the lightning bolt icon to the right of the activity on the Activity screen. This will process the activity.
8. Click the Activity Detail icon to the right of the processed activity. This will open the Activity Results window.
9. Click the **Accounting** link at the top of the window. All disbursements are listed in active status.
10. Click **Plan** on the Main Menu and select **Plan Activity**.
11. Select **Prototype Child Company** and **Functional Prototype Plan**.
12. Click the **Add Activity** link on the Secondary menu and select the **DisbursementUpdate** activity from the drop down box.
13. Click **Activity** from the Left Navigation menu. This will open the Activity screen.
14. Click the lightning bolt icon to the right of the activity to process it.
15. Click the Activity Detail icon to the left of the processed activity. This will open the Activity Results window.
16. Click the **Accounting** link at the top of the window. All disbursements are listed in active status.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Enhancements to Activity & Allocation Field in Screen Math

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Enhancements to Activity & Allocation Field in Screen Math

This is a new feature for processing of allocation details in the transaction with the ability to fetch the context variable in the allocation screens and transaction math as a part of configuration. The Plan considered for the prototype is Model prototype plan, and by using the verification screen in the transaction, the prototype generates some warning messages to show how to use context variable in the screen math section. Clients number is used along with Relative GUID and Activity GUID.

Prototype Prerequisites

PAS user should have access to "Model Prototype Plan". The Plan must have funds attached to it.

Prototype Explanation

This prototype configuration showcases the access to allocation details and activity details in the screen math and regular math sections of configuration.

Transactions

This configuration showcases the access to allocation details and activity details into screen math and regular math section. The transaction name is AllocationAndActivityDetailInScreenMath under the Model Prototype Plan.

Prototype Configuration

The configuration fetches the Allocation Detail in both the screen math and regular math configuration sections with the help of two math loops: **FILLBY-ALLOCATIONFROM** and **FILLBY-ALLOCATIONTO**.

Two warning messages were configured to satisfy the following cases:

- Substitution of the screen math variable that holds the ContextVariable value.
- Substitution of the screen math variable that holds the AllocationDetail value.

The context below describes accessing the activity fields. The configuration will generate a warning message upon OnLoad of this transaction, based on the value of RelatedGUID.

Two different warning messages are configured. If the RelatedGUID is null then a warning message "This is a Natural Activity" will appear. Otherwise, if the RelatedGUID is not null, a warning message "This is an Undo/Redo Activity" will appear.

The RelatedGUID, which is generated when the activity is reprocessed, used in the above context, is retrieved in the GlobalScreenMath section. Like RelatedGUID, ActivityGUID is also retrieved in GlobalScreenMath section, and the system checks if it is empty.

- If ActivityGUID is empty, then the warning message “This is the first time the activity is getting loaded” is shown on the screen upon Onload of the screen.
- If ActivityGUID is not empty, then the warning message “This is not the first time the activity is getting loaded” is shown on the screen upon OnLoad of the screen.

Along with the warning message, an Assign operation is also configured, which will assign the TransactionName retrieved in the GlobalScreenMath section to the activity field TransactionName.

The Warning Message uses substitutions for ClientNumber & ClientName: “The Client \$\$\$ClientNameSMV\$\$\$ has ClientNumber \$\$\$ClientNumberSMV\$\$\$” is shown to the user.

The following activity data is being accessed within the ScreenMath and placed in the associated variable:

- RelatedGUID - RelatedGUIDSMV.
- ActivityGUID - ActivityGUIDSMV.
- TransactionName - TransactionNameSMV.
- ClientNumber - ClientNumberSMV.
- ClientNameSMV - Retrieves ClientName with the help of ClientNumber using SQL.

The context below describes the accessing of the **AllocationFields**.

As part of the Onsubmit event, the transaction will display a warning that shows the TotalAllocationAmount of the “From” Fund along with the message, “The Total Allocation amount of 'From' Fund is XXX”.

Database Configuration

The following allocation data being accessed within the ScreenMath to produce the variable SumOfFromFundValueSMV is “AllocationAmount.”

The AllocationAmounts of each of the “From” and “To” allocations are summed and assigned to transaction fields. These fields are displayed in the verification screen. The verification screen also displays the total of the “From” allocation from an activity math variable.

The screen math and activity math variables below were created in the math section.

ScreenMath:

- SumOfFromFundValueSMV
- SumOfToFundValueSMV

Activity Math:

- SumOfFromFundValueMV.

The fields that are displayed in these ScreenMath and Activity math variables in the Verification screen are "From AllocationAmount In Screenmath," "To AllocationAmount In Screenmath" and "From AllocationAmount In TransactionMath."

Business Rules:

The ValidateExpressions business rule will check whether the total allocation amount of the “From” fund is equal to the total allocation amount of the “To” fund. If the values are not equal, then a message "Allocation amount of FromFund should be equal to allocation amount of ToFund" will be displayed. Since the OVERRIDE attribute is set to "Yes," the user is able to override that error. If the values are equal, then this error message will not be displayed.

The following activity math variables were created to execute this feature rule:

- SumOfFromFundValueMV.
- SumOfToFundValueMV.

View Prototype in OIPA

1. Log in OIPA using the Prototype Company user ID and password.
2. Click **Policy | New** from the Main menu.
3. Add a shell policy to test the prototype.
4. Click **Add Activity** on the Secondary menu.
5. Select the AllocationAndActivityDetailsScreenmath activity from the **Activity** drop down box.
6. View the Warning messages displayed as given in the prototype examples.
7. Select the **Allocations** tab and fill in the allocation details in the various fields.
8. Process the "AllocationAndActivityDetailsScreenmath" transaction as explained above to view the functionality.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Enhancements to Allocation Screens

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Enhancements to Allocation Screens

OIPA now supports the ability to filter the funds available on the Allocation screen using SQL queries. Criteria that must be satisfied for the funds to be available on the Allocation screen can be grouped with logical "and" and "or" relationships.

Prototype Prerequisites

- The policy must belong to a plan with the Plan Allocation Method set to “Model Supported.”
- The funds in the table below must be attached to the plan.
- The policy must be in pending status in the Model Prototype Plan.
- The default "Dollar Cost Averaging Program" should exist for the policy in order to process all the transactions

Fund Name	Fund Code	Redemption Factor	Redemption Fee	Redemption Duration
BigPlus Equity Income	320			8
Finlux Diversified Bond	230	3	Yes	4
GS Dynamic Equity	800	5	No	6
BigPlus China Fund	100	1	Yes	2
Money Market Fund	900	7	No	8
GS Midcap Value	700			

Prototype Explanation

The following business rules and transactions were configured in the Model Prototype Plan, located within the Prototype Company, to demonstrate this new functionality.

Business Rules

- **PolicyAllocationScreen:** This business rule controls the available allocation types on the policy-level Allocation screen. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Model Prototype Plan | Plan Rules | PolicyAllocationScreen** and click on the **XML Source** pane. The key configuration for this transaction is explained below.
 - The following allocation types and their supporting configuration are made available by this rule:
 - AIP From Fund (Allocation TypeCode = 13)
 - An ONLOAD event is configured to show or hide the "From" section of the Allocation screen based on the value of the PlanDate policy fixed field. If PlanDate's value is before SystemDate's value, the "From" section will display.
 - Systematic Withdrawal Program (Allocation TypeCode = 11)
 - Both the "From" and "To" sections of the Allocation screen are available for user entry.
 - A SQL query (configured in the <Query> element) is configured to return a single fund for both the "From" and "To" sections. The fund that has the following fund dynamic field values will display:
 - FundCode with value of "100"
 - RedemptionFactor with value of "1"
 - RedemptionFee with value of "Yes"
 - Dollar Cost Averaging Program (Allocation TypeCode = 14)
 - Both the "From" and "To" sections of the Allocation screen

are available for user entry.

- For the "From" allocation, the list of funds is filtered based on the fund fields and their values listed below. The individual criteria are joined together using an <And> element.
 - FundName with a value of "BigPlus China Fund," "Finlux Diversified Bond," "GS Dynamic Equity" or "Money Market Fund"
 - FundCode with a value of "100"
 - RedemptionFee with a value of "No"
 - RedemptionDuration with a value of "4"
- For the "To" allocation, the list of funds is first filtered based on the criteria in the <FundFields> element, then on the criteria in the <FundStatuses> element. Funds that satisfy the following criteria will be returned:
 - FundName with a value of "BigPlus China Fund," "Finlux Diversified Bond," "GS Dynamic Equity" or "Money Market Fund"
 - Active funds
- Automatic Investment Plan (Allocation TypeCode = 16)
 - Both the "From" and "To" sections of the Allocation screen are available for user entry.
 - For the "From" allocation, the list of funds is filtered based on the fund fields and their values listed below. The individual criteria are joined together using an <Or> element.
 - FundName with a value of "Finlux Emerging Market Fund"
 - TypeCode with a value of "01"
 - Redemption Factor with a value of "3"
 - Redemption Fee with a value of "Yes"
 - Redemption Duration with a value of "8"
 - For the "To" allocation, the list of funds is first filtered based

on the criteria in the <FundFields> element (which contains a combination of <And> and <Or> elements), then on the criteria in the <FundStatuses> element. Funds that satisfy the following criteria will be returned:

- Within the <And> element of configuration, FundName with a value of "BigPlus China Fund," "Finlux Diversified Bond," "GS Dynamic Equity" or "Money Market Fund"
 - Within the <Or> element of configuration, FundName with a value of "GS MidCap Value" or "GS Large Value"
 - FundCode with a value of "700"
 - Active funds
- **TransactionAllocationScreen:** This business rule drives the Allocation tab for activity allocations to capture the functionality for allocation models and funds. To view the prototype configuration, navigate in the Main Explorer to **Global Rules Explorer | Business Rules | Screen | TransactionAllocationScreen | Transaction Overrides** and click on the **XML Source** pane. This business rule is attached to the following transactions for the purposes of this prototype:
 - AllocationsTransferPrototype
 - AllocationSqlFilterPrototype
 - AllocationsFieldFilterAndPrototype
 - AllocationsFieldFilterOrPrototype

Transactions

- **AllocationsTransferPrototype:** This transaction will display either both allocation sections ("From" and "To") on the Allocation tab of the activity, or the "To" section only depending on the value of the SourceAllocationType activity field, and is configured to return multiple funds for both "From" and "To" allocations. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype**

Company | Subsidiary Companies | Prototype Child Company | Plans | Model Prototype Plan | Transactions | Policy Transactions | AllocationsTransferPrototype. The key configuration for this transaction is described below.

- The SourceAllocationType combo box field is configured to have the values "Auto Transfer" and "Manual Transfer" available for selection.
- The TransactionAllocationScreen business rule is attached, and is configured to perform the following functions:
 - Populate the details of the default policy-level Dollar Cost Averaging allocation to the Allocation tab of the activity.
 - Display or hide the "From" allocation section on the Allocation tab based on the value of the SourceAllocationType field. If the field has a value of "Auto Transfer," only the "To" allocation section will display; if the field has a value of "Manual Transfer," both the "From" and "To" sections will display.
 - Retrieve multiple funds for both the "From" and "To" allocations' fund lists.
- The ReassignAllocations business rule is attached, and is configured to provide the data for the "From" allocation section of an activity. The rule is additionally configured as follows:
 - The rule is only invoked when the SourceAllocationType activity field is set to "Auto Transfer" using the TransactionBusinessRulePacket business rule.
 - The <From> section of configuration is configured to retrieve the default allocation (i.e. Dollar Cost Averaging) details that exist at the policy level.
 - The <To> section of configuration is configured to retrieve the fund details that exist at the activity level.
- **AllocationsSqlFilterPrototype:** This transaction is configured to filter its lists of available funds based on a SQL query defined in the attached TransactionAllocationScreen business rule. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Model Prototype Plan | Transactions | Policy Transactions | AllocationsSqlFilterPrototype.** The key configuration for this transaction

is described below.

- Both the "From" and "To" allocation sections on the Allocation tab of the activity will be populated according the <Query> element of the attached TransactionAllocationScreen business rule.
- The TransactionAllocationScreen business rule is attached, and is configured to perform the following functions:
 - Populate the details of the default policy-level Dollar Cost Averaging allocation to the Allocation tab of the activity.
 - Filter the list of funds for both the "From" and "To" allocations available to the activity. Only funds that satisfy the following criteria will be available:
 - FundCode with a value of "100"
 - RedemptionFactor with a value of "1"
 - RedemptionFee with a value of "Yes"
- **AllocationsFieldFilterAndPrototype:** This transaction is configured as described below. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Model Prototype Plan | Transactions | Policy Transactions | AllocationsFieldFilterAndPrototype.**
 - The "From" allocation fund list is populated according to the criteria configured in the <FundFields> element of the attached TransactionAllocationScreen business rule.
 - The "To" allocation fund list is populated according to the <Query>, <FundFields> and <FundStatuses> elements of the attached TransactionAllocationScreen business rule.
 - Only one fund will be retrieved for both the "From" and "To" allocations' fund lists.
 - The TransactionAllocationScreen business rule is attached, and is configured to perform the following functions:
 - Populate the details of the default policy-level Dollar Cost Averaging allocation to the Allocation tab of the activity.
 - Add/modify the funds available within the "From" and "To" allocation sections, and filter them as described below:

configured to perform the following functions:

- Populate the details of the default policy-level Dollar Cost Averaging allocation to the Allocation tab of the activity.
- Add/modify the funds available within the "From" and "To" allocation sections, and filter them as described below:
 - "From" allocation:
 - Fund filtering will be performed based on the fund field values listed below. These criteria are configured in the <FundFields> element, and are combined by an <Or> element.
 - FundName with a value of "Finlux Emerging Market Fund"
 - TypeCode with a value of "01"
 - RedemptionFactor with a value of "3"
 - RedemptionFee with a value of "Yes"
 - RedemptionDuration with a value of "8"
 - "To" allocation:
 - The fund list is first filtered based on the criteria configured in the <FundFields> element as described below.
 - The following fund field criteria are configured in the <And> element:
 - FundName with a value of "BigPlus China Fund"/"Finlux Diversified Bond"/"GS Dynamic Equity"/"Money Market Fund"
 - The following fund field criteria are configured in the <Or> element:
 - FundName with a value of "GS Large Value"/"GS MidCap Value"
 - FundCode with a value of "700"
 - The fund list is then filtered again based on the <FundStatuses> element—only active funds will be available.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password.
2. Open a policy in "Pending" status or create a new policy. The policy should belong to a plan that uses the Model Supported Plan allocation type. If creating a new policy, the PremiumWithAllocation activity can be used to apply the premium to multiple funds.
3. View the functionality configured for the above types of fund allocation types in two sets of screens, i.e. Policy Allocation Screen & Transaction Allocation Screen.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Enhancements to Verification Screen

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Enhancements to Verification Screen

The Verification Screen business rule was enhanced to provide more robust support of activity errors and warnings. Error numbers presented on the screen in OIPA are now paired with error messages to provide a better understanding of the issues encountered during processing.

Refer to the XML Configuration Guide for a complete explanation of the elements and attributes supported in the VerificationScreen business rule.

Prototype Explanation

The following transactions and supporting configuration were created in the Functional Prototype Plan to demonstrate this new functionality.

Transactions

There are three transactions configured. Navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Transactions|** to view the configuration for each.

- **DefineActivityButtonsPrototype:** this transaction defines the presence of the Verify button through the <Button> configuration section. The OK button is not defined on Activity Detail screen. There are two attached rules:
 - **VerificationScreen:** the <ErrorNumberStartsWith> element has a value of G inside <DisableOkButton> elements. This disables the Ok button when an error beginning with G occurs. There is also a <Warnings> element with a Yes value used to display the Warnings section in the Verification Screen grid.
 - **ValidateExpressions:** this rule uses the new attribute WARNING such that ERRORNUMBER="W035" is defined as a warning (Premium amount is less than minimum plan premium).
- **VerifyAllErrorsPrototype:** this transaction is used to verify any errors that occur during processing. There are three attached rules:
 - **VerificationScreen:** this rule uses the attribute ALLERRORS with a value of Yes to disable the Ok button when any nonoverridable error occurs.
 - **ValidateExpressions:** this rule defines G001 as a non-overridable error, indicating the withdrawal amount is greater than available cash.
 - **ConfirmationScreen:** this rule contains one field confirmation number.
- **VerifyWithdrawal:** this transaction is used to verify a withdrawal activity before it processes. There are three attached rules:
 - **ValidateExpressions:** this rule is configured with an overridable error W034, which says, "Withdrawal amount is less than plan

minimum withdrawal." It demonstrates the ability to override the error directly from the Verification screen when a withdrawal is less than \$10.00.

- **VerificationScreen:** this rule uses the element <ErrorNumberIs> with a value of G001 within <DisableOkButton> elements resulting in the disabling of the Ok button when withdrawal amount is greater than available cash.
- **ConfirmationScreen:** this rule contains one field confirmation number.

Plan Fields

Three plan fields were added to the Functional Prototype Plan. Navigate to **Main Explorer | Prototype Company | Subsidiary Company | Prototype Child Company | Plans | Functional Prototype Plan | Plan Data** to view the new fields.

- MinimumWithdrawalAmount: set at \$10
- MinimumPremiumAmount: set at \$10
- MaximumPremiumAmount: set at \$100000.00

Error Catalog

Three errors were added to the Error Catalog in the Admin Explorer. Navigate to **Admin Explorer | Administration | Error Catalog** to view the new errors.

- Added G001: Withdrawal amount is greater than available amount
- Added G002: Premium amount is greater than maximum amount
- Added W035: Premium amount is less than minimum plan premium

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password. Make sure the user has privileges to view the Client screen and client activities.
2. Open a policy or create a shell policy and add each of the three prototype transactions described above, one at a time.
3. When adding each activity, click the **Verify** button before adding the activity. This will open the Verification screen and display any errors or warnings. An override option will be available, if supported in configuration.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Error Messages with Substitutions

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Error Messages with Substitutions

OIPA supports user defined error messages that contain substitutions. These substitutions can be math variables, fields or system values using prefixes, which display additional error details to the user.

The areas of the system that allow for user defined error messages are [ValidateExpressions](#), [PostAssignmentValidateExpressions](#), [Calculate Segment](#) rule messages and messages within [Actions and Events](#).

If a message is configured in the ValidateExpressions or PostAssignmentValidateExpressions rule, then it will display when the lightning bolt icon is clicked on the Activity screen and the activity attempts to process. If a message is configured in a Calculate Segment rule, then it will display when the segment is added to the policy. If a message is configured in Actions and Events, it will display during the process of entering the activity information on the Add Activity screen.

Scenario

A premium activity has a field for Premium Amount. There is also a plan minimum that is required to be paid or the premium is not allowed. A user has a Universal Life plan. The plan premium minimum is \$5,000. A user attempts to enter a premium of \$3,000. When the user tabs out of the field, OIPA displays a message saying, "The premium amount entered of \$3000 is less than the \$5000 minimum for this plan."

Prototype Examples

- An **ErrorMessageWithSubstitutions** transaction was configured to show how messages can be configured in Actions and Events. The new transaction demonstrates an Actions validation using Money, Text and Integer field values along with Currency, Text and Integer ScreenMath values in an <Actions> validation message. The validation will appear when field Money1 is less than field Money2. This will generate an error that uses the values from the fields and ScreenMath variables as substitutions. Navigate to **Global Explorer | Transactions | ErrorMessageWithSubstitutions | ErrorMessageWithSubstitutions (Functional Prototype Plan)**.
- A **ValidateExpressions** rule was attached to the ErrorMessageWithSubstitutions transaction. This rule demonstrates the use of field values along with multiple datatype math variables in the validation message. A message will automatically generate when the transaction is processed. It can be overridden, however, to allow the transaction to process. The message simply displays multiple data type field values and math variables. Navigate to **Global Explorer | Transactions | ErrorMessageWithSubstitutions | ErrorMessageWithSubstitutions (Functional Prototype Plan) | Attached Rules | ValidateExpressions (ErrorMessageWithSubstitutions-Functional Prototype Plan)**.
- A **Calculate General** rule was attached to the Base Coverage segment in the Functional Prototype Plan. This rule tests substitutions in the Validations section of the SegmentAmount field from the Base Coverage segment along with a currency code of 'USD'. The error message will display the numeric field value from the SegmentAmount and a math variable text value of 'USD'. Navigate to **Global Explorer | Segments | Base Coverage | Base Coverage (Functional Prototype Plan) | Calculate Rule | Calculate General**.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Field Level Security

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Field Level Security

Two new security groups were created to demonstrate the three levels of field security supported in OIPA: visible and editable, hide field value and disable field value.

Configuration Requirements

- **Security Groups:** Two security groups were created and assigned various levels of field security.
 - **Prototype Analyst** – Primary Company: Prototype Company.
 - Access has been granted to all company pages for the Prototype Company unless noted otherwise.
 - Access has been granted to all plan pages for the Functional Prototype Plan unless noted otherwise.
 - Access has been granted to all plan pages for the Dynamic Prototype Plan unless noted otherwise.
 - Access has been granted to all transactions for the Functional Prototype Plan unless noted otherwise.
 - Access has been granted to all transactions for the Dynamic Prototype Plan unless noted otherwise.
 - **Prototype Tester** – Primary Company: Prototype Company
 - Access has been granted to all company pages for the Prototype Company unless noted otherwise.
 - Access has been granted to all plan pages for the Functional Prototype Plan unless noted otherwise.
 - Access has been granted to all plan pages for the Dynamic Prototype Plan unless noted otherwise.
 - Access has been granted to all transactions for the Functional Prototype Plan unless noted otherwise.
 - Access has been granted to all transactions for the Dynamic Prototype Plan unless noted otherwise.

Each security group has security applied to specific fields to demonstrate the various levels of field security. An explanation of the field security is provided below.

Prototype Analyst

- Prototype Company - ClientScreen. Navigate to **Admin Explorer | Security | Application Security | Prototype Analyst | Company**

Security | Company Pages | Prototype Company | Client.

- TaxID: Hide field value. Click the Fields pane and scroll down to the TaxID field. Hide field value is selected in the Security drop down box for that field.

- **Prototype Company - ClientAddressScreen. Navigate to Admin Explorer | Security | Application Security | Prototype Analyst | Company Security | Company Pages | Prototype Company | ClientAddress.**
 - ExpirationDate: Disable field value. The calendar button will not be displayed in OIPA.
 - FaxNumber: Hide field value.

- **Prototype Company - SuspenseScreen. Navigate to Admin Explorer | Security | Application Security | Prototype Analyst | Company Security | Company Pages | Prototype Company | SuspenseRecord.**
 - BankName: Disable field value. This is a configured fixed field.
 - BankNumber: Hide field value. This a configured fixed field.

- **DynamicPrototype Plan - PolicyScreen. Navigate to Admin Explorer | Security | Application Security | Prototype Analyst | Plan Security | Prototype Child Company| Dynamic Prototype Plan | Plan Pages | Policy.**
 - PolicyName: Hide field value. This is a fixed field.
 - PlanDate: Disable field value. This is a fixed field.

- **DynamicPrototype Plan - SegmentScreen. Navigate to Admin Explorer | Security | Application Security | Prototype Analyst | Plan Security | Prototype Child Company| Dynamic Prototype Plan | Plan Pages | Segment.**
 - SegmentAmount: Disable field value. The currency drop down box will be disabled in OIPA as well.

- **FunctionalPrototype Plan - IdentifierFields transaction. Navigate to Admin Explorer | Security | Application Security | Prototype Analyst | Transaction Security | Prototype Child Company| Functional Prototype Plan | IdentifierFields**
 - FieldID: Hide field value.
 - ClientID: Disable field value.

- **FunctionalPrototype Plan - ContractSign transaction. Navigate to Admin Explorer | Security | Application Security | Prototype Analyst | Transaction Security | Prototype Child Company| Functional Prototype Plan | ContractSign**
 - OpenDate: Disable field value.
 - DueDate: Hide field value.

- **FunctionalPrototype Plan - StatusChange transaction. Navigate to Admin Explorer | Security | Application Security | Prototype Analyst | Transaction Security | Prototype Child Company| Functional Prototype Plan | StatusChange**
 - NewStatus: Disable field value.

Prototype Tester

- **FunctionalPrototype Plan - StatusChange transaction. Navigate to Admin Explorer | Security | Application Security | Prototype Tester | Transaction Security | Prototype Child Company| Functional Prototype Plan | StatusChange**
 - NewStatus: Disable field value.

- **FunctionalPrototype Plan - ContractSign transaction. Navigate to Admin Explorer | Security | Application Security | Prototype Tester | Transaction Security | Prototype Child Company| Functional Prototype Plan | ContractSign**
 - Access was not granted for this transaction.

- **PrototypeCompany - Client. Navigate to Admin Explorer | Security | Application Security | Prototype Tester | Company Security | Company Pages | Prototype Company | Client**
 - Access was not granted for this page.

- **Dynamic Prototype Plan - PolicySegment. Navigate to Admin Explorer | Security | Application Security | Prototype Tester | Plan Security | Prototype Child Company | Dynamic Prototype Plan | Plan Pages | PolicySegment.**
 - Access was not granted for this page.

You are here: Configuration > Valuation > Valuation Business Rules > Fixed Interest Rate Calculation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Fixed Interest Rate Calculation

When performing valuation, OIPA must be able to accurately calculate and then credit interest to a fixed fund. There are many factors that come into play, including various calculation methods, fixed, variable or additive rates and multiple definitions of what constitutes a year.

Scenario

Multiple scenarios were constructed to demonstrate this functionality. Each of the ten funds created for the Prototype Company were created to demonstrate a particular aspect of the functionality. The types of situations that the funds were configured to represent are listed below. Refer to the fund descriptions below to see which fund demonstrates which aspect of fixed interest rate calculation.

- Recognize leap year
- simple interest
- guaranteed rates
- additive rates
- varying definitions of a year

Configuration Requirements

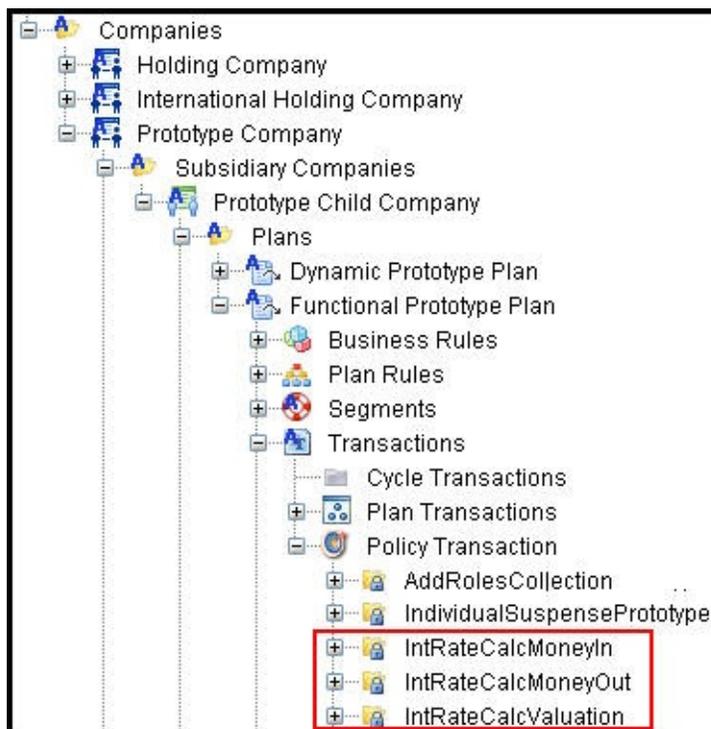
The following configuration requirements are necessary to implement fixed interest rate calculation.

- funds must be configured with InterestRateCalculation fund level overrides for each fund.
- rates used by the funds should be uploaded through the Admin Explorer, using the Rate Groups editor or SQL statements.
- A table must exist to hold additive rate codes if needed: AsCodeAdditiveRate.
- Components to execute valuation like transactions, ValuesScreen and InquiryScreen.

Prototype Samples

There are three transactions configured to demonstrate fixed interest rate calculation. Navigate through the following folders in the Main Explorer to locate the configuration sample: **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions.**

- IntRateCalcMoneyIn: Demonstrates the ability to invest in one or multiple funds with the calculation of interest performed independently of each other.
- IntRateCalcValuation: Calculates interest to the point in time that is selected.
- IntRateCalcMoneyOut: Calculates simple interest. Principle and interest are tracked separately. When money is taken out, the interest is taken first followed by the principle. This activity demonstrates how the principle and interest columns are affected.



Fixed Interest Rate Calculation Transactions in Main Explorer

There are two fields added to the PolicyScreen. This can be seen in the PolicyScreen plan level override. Navigate in the Main Explorer to

Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Plan Rules | PolicyScreen (Functional Prototype Plan).

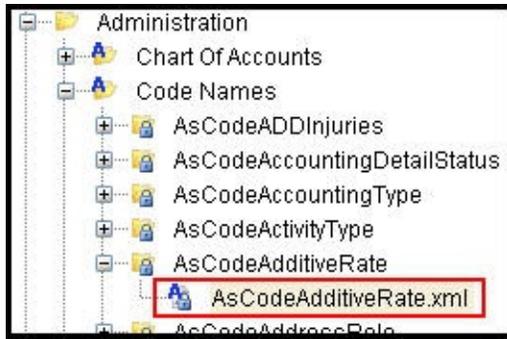
- InterestBonusQual
- EffectiveDate

```
<PolicyScreen>
  <Fields>
    <Field>
      <Name>InterestBonusQual</Name>
      <Display>Interest Bonus Qualified</Display>
      <DataType>Combo</DataType>
      <Query TYPE="FIXED">
        <Options>
          <Option>
            <OptionValue>Yes</OptionValue>
            <OptionText>Yes</OptionText>
          </Option>
          <Option>
            <OptionValue>No</OptionValue>
            <OptionText>No</OptionText>
          </Option>
        </Options>
      </Query>
    </Field>
    <Field>
      <Name>EffectiveDate</Name>
      <Display>Policy Effective Date</Display>
      <DataType>Date</DataType>
    </Field>
  </Fields>
</PolicyScreen>
```

PolicyScreen Business Rule with New Interest Fields

There is one new AsCode table. The values in this table can be edited through the Code Names folder in Admin Explorer. Navigate to Admin Explorer and open **Administration | Code Names | AsCodeAdditiveRate**.

- AsCodeAdditiveRate



AsCodeAdditiveRate Editor in Admin Explorer

There are ten funds and each demonstrates specific calculations. The details of each fund's configuration are explained below. They are located in the Funds folder. Navigate to the Main Explorer and open **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Funds**.

- Interest Rate Calc Fund 1
 - Static decimal rate
 - Current interest with one additive interest that sometimes adds to the current interest rate. The additive interest is turned on or off based on whether the InterestBonusQual PolicyScreen field is set to Yes. If set to No, the additive rate = 0.
 - Compound interest
 - Policy year (365)
 - Interest period end
 - Helped to determine when the rate changed; end of day or beginning of day.
 - Adjust date calculation: no
 - Fund level tracking

- Interest Rate Calc Fund 2
 - Lookup a percentage rate
 - Immediate change upon declaration of new rate
 - Current interest only
 - Compound interest

- Calendar year (365)
- Interest period begins at the beginning of the day
- Adjust date calculation: no
- Fund level tracking

- Interest Rate Calc Fund 3
 - Lookup a decimal rate
 - Changes on anniversary of deposit
 - Last known rate; the rate will not change until anniversary
 - Current interest only
 - Compound interest
 - 360 day year
 - Interest period begin
 - Adjust date calculation: no
 - Deposit level tracking

- Interest Rate Calc Fund 4
 - Look up a decimal rate
 - Changes on anniversary of deposit
 - Age of deposit (For example, on the day of deposit, the deposit date age = 0)
 - Current interest only
 - Compound interest
 - Policy year
 - Interest period begin
 - Adjust date calculation: no
 - Deposit level tracking

- Interest Rate Calc Fund 5
 - Look up a decimal rate
 - Changes on anniversary of policy

- Age of policy
 - Current interest
 - Compound interest
 - 365 day year with leap day
 - Interest period begin
 - Adjust date calculation: no
 - Guaranteed interest rate
 - Static rate: compares current rate against guaranteed rate to determine which rate is greater
 - Fund level tracking

- Interest Rate Calc Fund 6
 - Look up a percentage rate
 - Changes on the anniversary of the policy, biennially guaranteed for two years
 - Age of policy
 - Current interest
 - Compound interest
 - 365 day year with no leap day
 - Interest period begin
 - Adjust date calculation: no
 - Guaranteed interest rate
 - Lookup rate
 - Fund level tracking

- Interest Rate Calc Fund 7
 - Look up a decimal rate
 - Changes on anniversary of policy
 - Current interest
 - Simple interest
 - Policy year

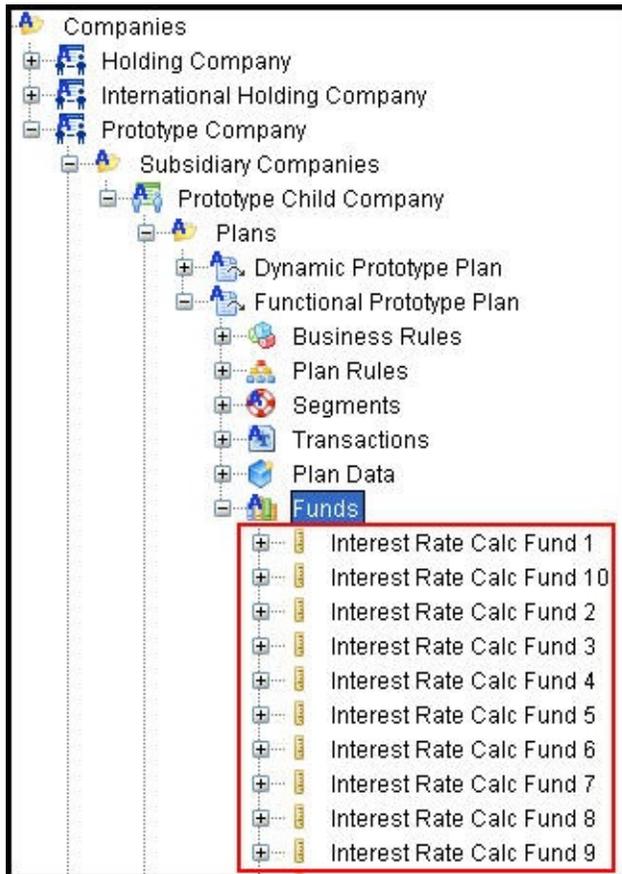
- Interest period begin
- Adjust date calculation: no
- Guaranteed interest rate
 - Static
- Fund level tracking

- Interest Rate Calc Fund 8
 - Lookup a decimal rate
 - Immediate change
 - Current interest only
 - Compound interest
 - Calendar year
 - Interest period end
 - Adjust date calculation: no
 - Fund level tracking

- Interest Rate Calc Fund 9
 - Lookup a decimal rate
 - Immediate change
 - Current interest only
 - Compound interest
 - Calendar year
 - Interest period begin
 - Adjust date calculation, yes
 - Fund level tracking

- Interest Rate Calc Fund 10
 - Lookup decimal rate
 - Never changes
 - Current interest only
 - Compound interest

- Policy year
- Interest period begin
- Adjust date calculation, no
- Fund level tracking

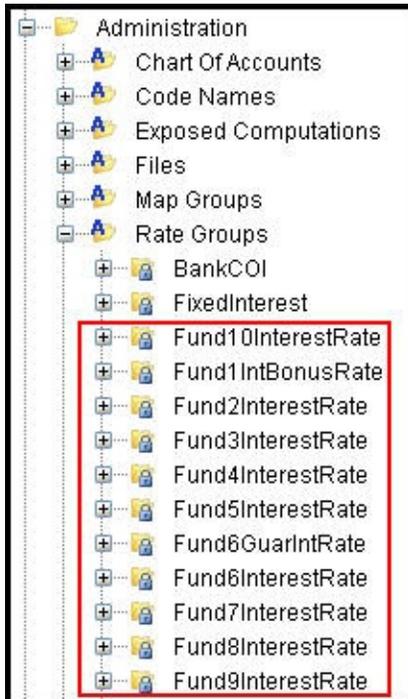


Interest Rate Funds in Main Explorer

Rates were added for each fund and can be found in the Admin Explorer in the Rate Groups folder. The Rate Descriptions are as follows:

- Fund 1: Fund1IntBonusRate or ZeroRate
- Fund 2: Fund2InterestRate
- Fund 3: Fund3InterestRate
- Fund 4: Fund4InterestRate
- Fund 5: Fund5InterestRate
- Fund 6: Fund6InterestRate and Fund6GuarIntRate
- Fund 7: Fund7InterestRate

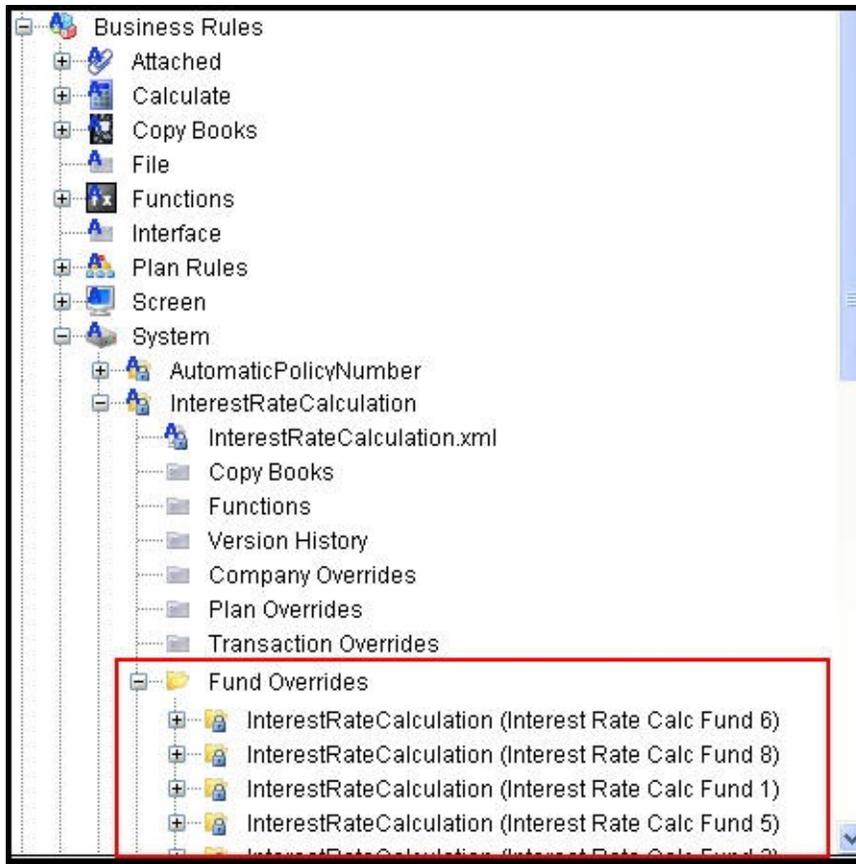
- Fund 8: Fund8InterestRate
- Fund 9: Fund9InterestRate
- Fund 10: Fund10InterestRate



Rates for Funds in Prototype Company

There is one new business rule created to expose the components that describe the attributes of the interest calculation of fixed funds. This new rule will support deposit and fund valuations. This system rule exists globally, void of any configuration, with Fund Overrides for each of the ten funds. Navigate in the Global Rules Explorer to **Business Rules | System | InterestRateCalculation | Fund Overrides**.

- InterestRateCalculation



InterestRateCalculation Fund Level Overrides in Global Explorer

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Imperial Calendar Support

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Imperial Calendar Support

The Imperial Date Calendar is supported by extension in the OIPA system so that users can enter a date in a date field using the imperial date calendar. Currently when a Imperial date is entered in the system the date is stored in the database on save as a Gregorian date. In order to support use of the Imperial calendar a configuration setting has been introduced that allows users to enter an Imperial date on the screen, and stores it as a Gregorian date in the database. Upon re-display of that date the Gregorian date from the database is converted to an Imperial date.

Configuration Requirements

There are two steps involved in adding support for the Imperial calendar. First, the configuration must be added to the screen supporting the calendar type. Second, translations must be added to the AsTranslation table. This can be done using the [Localization Editor](#) tool in the Rules Palette or by adding the translations manually to the database.

- ClientScreen business rule: Configuration was added to this rule. It can be viewed from **Global Explorer | Business Rules | Screen | ClientScreen | ClientScreen (International Holding Company)** .
 - The DateofBirth and DateofDeath fixed fields use the new FORMAT and CALENDAR attributes.
 - The GregorianDate dynamic field uses the Gregorian translation key.
 - The JapaneseDate dynamic field uses the Japanese translation key.

- Several translations were added to support the Imperial and Gregorian calendar configuration:
 - **en-us**
 - TranslationKey: Gregorian
 - TranslationValue: Gregorian
 - TranslationKey: Japanese
 - TranslationValue: Japanese
 - TranslationKey: LocalCalendar
 - TranslationValue: Gregorian
 - **ja-jp**
 - TranslationKey: Gregorian
 - TranslationValue: Gregorian
 - TranslationKey: Japanese
 - TranslationValue: Japanese

- TranslationKey: LocalCalendar
- TranslationValue: Japanese

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Foreign Address Support

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Foreign Address Support

The Address screen was enhanced to support foreign addresses. New address format definitions are driven by the combination of address type and [country](#).

Address Type + Country = Address screen format

Configuration Requirements

The following configuration considerations apply when configuring the Address screen to support foreign addresses.

- **<AllowExpiration>**: This element dictates whether or not the EffectiveDate and ExpirationDate Fixed Fields are visible and used on the Address Screen.
- **<DisplayFormat>**: This element will dictate how the address is displayed in the AddressScreen summary table, ClientSearchScreen Results table and DisbursementDisplay screen.
- **Fields with DataType = Date**: Specific date formats can now be configured using the FORMAT attribute of the <DataType> element, and translation records configured in AsTranslation table.
- **CopyToAddressScreen business rule (attached to a transaction)**: This validates whether or not fields configured within the <To> element actually exist in the AddressScreen business rule configuration. If not, a system error will be generated.

The following existing features will persist:

- <Expanded> element for Fixed and Dynamic Fields
- Default address in Address Summary table cannot be deleted
- CopyBooks are functional
- Events are functional

Note: A complete explanation of the elements and attributes used to configure the Address screen is provided in the XML Configuration Guide. Select **Help | XML Configuration Guide | Business Rules | Screen Rules**.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Large Text Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Large Text Prototype

OIPA supports fields that allow a user to enter large amounts of text. This text can be viewed in its entirety on the screen and in the Activity Results tab after an activity is processed.

Scenario

The Issue transaction has been configured with a TextArea field to enter free-form information. The character limit is 260. This field is used by the CSR to enter free-form information that is fed to the document system used to create policy pages. The CSR enters 230 characters and is able to see all information that was entered. An update is made to the text field that pushes the character amount to 270. The first 260 characters are visible to the CSR.

Prototype Examples

- A **LargeTextPrototype** transaction was configured to demonstrate the new TextArea field datatype, the BIGTEXT math variable datatype, and the new StringLength() function. Navigate to **Global Explorer | Transactions | LargeTextPrototype | LargeTextPrototype (Functional Prototype Plan)**.
 - Two MathVariables were created:
 1. BigTextMV1: the BIGTEXT datatype that assigns the field values of TextArea1 to a MathVariable.
 2. BigTextMV2: the BIGTEXT datatype that assigns the field values of TextArea2 to a MathVariable.
 - The two other MathVariables are TYPE=FUNCTION math variables. These math variables demonstrate the new StringLength() function.
 1. TextArea1Length: INTEGER datatype that uses the TextArea1 field as the parameter in the function. When processed this math variable will display the length of the TextArea1 field.
 2. BigTextMV2Length: INTEGER datatype that uses the BigTextMV2 as the parameter in the function. When processed this math variable will display the length of the BigTextMV2 math variable. This value is originally derived from the TextArea2 field.
 - An action using the IsEmpty() function is used to verify that screen functions work properly with the new TextArea datatype. If the TextArea1 field is empty, an OnSubmit validation message should display. Once TextArea1 is populated, the activity should allow processing.
 - The LargeTextPrototype transaction, when processed, will spawn itself twice to demonstrate using fields and math variable values in the <SpawnFields> section. The first spawn will use the TextArea1 and TextArea2 fields as the <From> spawn fields. The second spawn will use BigTextMV1 and BigTextMV2 as the <From> spawn fields. Both spawned activities use TextArea1

and TextArea2 as the <To> spawn fields.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Maintain Suspense

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Maintain Suspense Updates

The MaintainSuspense business rule supports the following activity features (Plan and Policy levels):

1. The ability to update any field on suspense
2. The ability to update multiple suspense tickets within the same activity
3. The ability to generate accounting entries using a suspense field as a trigger, the accounting must be visible on the associated suspense ticket.

This configuration document deals with items (1) and (2): ability to update one or multiple StatusCode and/or EffectiveToDate Suspense record fields, within multiple (or single) Suspense records.

Two transactions are configured to test CopyToSuspenseFields feature: Policy level "r;SuspenseCopyTo4945", and Plan level "r;SuspenseCopyToPlan4945" (configuration files attached to the JIRA are "r;SuspenseCopyTo", and "r;SuspenseCopyToPlan", respectively). Although there are minor configuration differences, both transactions perform essentially the same task: gather Suspense Record GUIDs, and then pass new StatusCode and EffectiveToDate values to the corresponding Suspense records. The following three MathVariables are contained within each of the two transactions:

1. "r;SuspenseArray" MathVariable is the "r;bounding" or "r;limiting" array of Suspense record GUIDS;
2. "r;SuspenseCollection2" MathVariable is the collection that contains a set of Suspense record GUIDs/new StatusCode pairs which will be used to update Suspense records' StatusCode fields. The set of collection Suspense GUIDs is a sub-set of the array Suspense GUIDs, created in "r;SuspenseArray" MathVariable;
3. "r;SuspenseCollection3" MathVariable is the collection that contains a set of Suspense record GUIDs / new EffectiveToDate pairs which will be used to update Suspense records' EffectiveToDate fields. The set of collection Suspense GUIDs is a sub-set of the array Suspense GUIDs, created in "r;SuspenseArray" MathVariable



A suspense record with a StatusCode of 05 ("MakeUp") must exist in order for SuspenseCollection3 to process successfully.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Manage Duplicate Clients

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Manage Duplicate Clients

The new **DuplicateClient** business rule was created to check for duplicate clients based on parameters that are configured in the business rule. Fields can be used together to determine if a duplicate client exists. The rule may be configured as a warning that the user can override or as a restriction that prevents the user from adding the duplicate client.

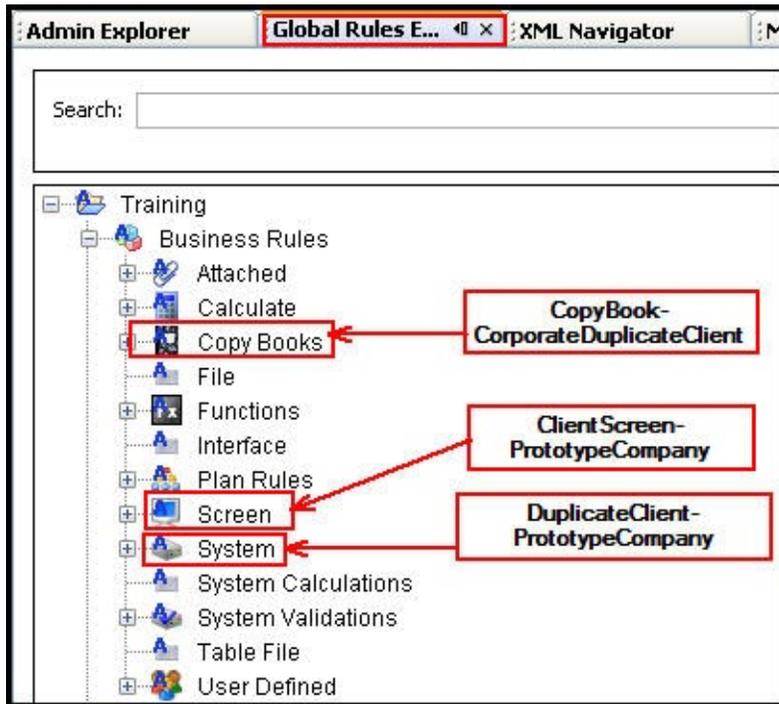
Configuration Requirements

The following configuration examples were created to demonstrate this new feature.

- A new Global Business rule called Duplicate Client was created at the System level. An override was created for the Prototype Company. Navigate to **Global Explorer | Business Rules | System | DuplicateClient | Company Overrides | DuplicateClient-PrototypeCompany**
 - **Individual Clients:** The following will be used to check duplicates for Individual clients: First Name, Last Name, Tax ID, Sex. The restriction for Individual clients was set to Denied. This means that duplicate clients cannot be added.
 - **Corporate Clients:** The following will be used to check duplicates for Corporate clients: Company Name and Tax ID. The restriction was set to Warning. This means that a pop-up warning will alert the user that a duplicate record exists. The user can click OK to add the record or Cancel to either change the entry date or remove the clients addition.
 - **Producer Clients:** The following will be used to check duplicates for Producer clients: First Name and Last Name. The restriction for Producer clients was set to Warning. This means that a pop-up warning will alert the user that a duplicate record exists. The user can click OK to add the record or Cancel to change the client or remove it.
- The CopyBook-CorporateDuplicateClient was created and placed in the duplicate rule for the corporate record type to demonstrate a potential use of a CopyBook in this rule. Navigate to **Global Explorer | Business Rules | CopyBooks | CopyBook-CorporateDuplicateClient**. This rule is referenced in the Prototype Company override.
- An override of the ClientScreen business rule was created at the Prototype Company level. Navigate to **Global Explorer | Business Rules | Screen | ClientScreen | Company Overrides | ClientScreen-PrototypeCompany**. Three client types were added

to this rule so they could be referenced by the DuplicateClient rule.

- **Individual:** typecode "02"
- **Corporate:** typecode "01"
- **Producer:** typecode "17"



Rules in Global Explorer for DuplicateClient Configuration

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Masking

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

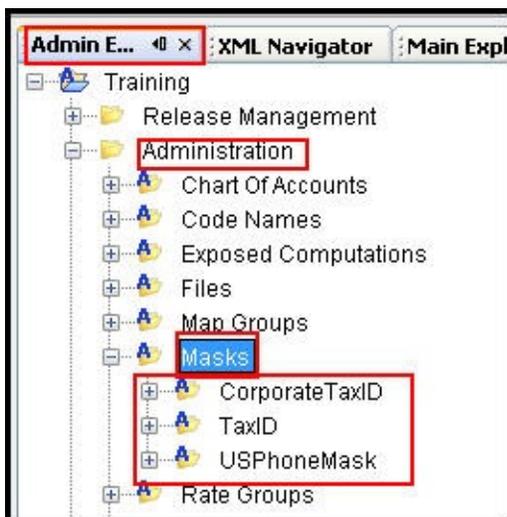
Masking

Masks can be used to control the visual format of configurable fields and fixed fields by modifying the entered and persisted data on the screen for visual purposes only. Parts of the data may be replaced with characters. Validations can be performed on the entered values to ensure they meet field requirements.

Configuration Requirements

The following configuration was created to demonstrate the use of masks in OIPA.

- **Define Security Levels for Masks:** There are four security levels defined in AsCodeMaskSecurityLevel: Restrictive, Medium, High and No Security. Navigate to **Admin Explorer | Administration | Code Names | AsCodeMaskSecurityLevel** to view these levels.
- **Create Masks:** There are three masks configured in the Admin Explorer [Mask Editor](#): CorporateTaxID, TaxID, USPhoneMask. Navigate to **Admin Explorer | Administration | Masks** to view these masks.

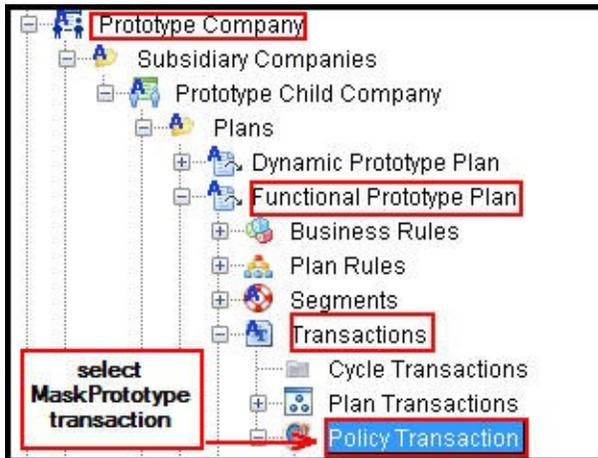


Masks in Admin Explorer

- **Create security levels for each mask.** A mask can have multiple security levels. Navigate to **Admin Explorer | Administration | Masks** and expand the Mask folder. Right-click on a mask name and select **New Mask Security Level**.
- **Create a transaction with fields using masks.** The MaskPrototype transaction has two fields with masks defined: an individual tax ID and a phone number. Open the transaction and click the [Fields](#) pane. Click on a field to open the FieldProperties window and view

the Mask property information.

- **Create a screen with fields using masks.** The ClientScreen for the Prototype Company has placed a mask on all tax ID fields, whether an individual or corporate client.



Prototype Company Navigation

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Multifield Events

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Multifield Events

This prototype demonstrates event functionality within the MultiFields section of a transaction. You will see how transaction field and math is used to trigger changes to multifields and also how action/events are configured within a MultiField business rule to trigger changes to a specific multifield section.

The Multifield action and event processing in screen business rules ensures that the functionality is extended to all other screens BR such as client, role etc.

Configuration Requirements

1. There are two pieces of configuration required to demonstrate this prototype. First, a Multifield rule must be configured to define the characteristics of the multifield. Then a transaction must be configured to display the multifields and allow user selections.
2. User should have access to "Functional Prototype Plan".

Prototype Configuration

- The Multifields business rule called **MultiField-MultiFieldEvents** demonstrates the ability to use Events within a MultiField section. Navigate to **Global Explorer | Business Rules | System | MultiField-MultiFieldEvents** to view the prototype configuration. The following event functionality is included in the prototype:
 - In the <Actions> section, OnLoad, OnChange and OnSubmit are configured.
 - In the <ActionSets> section, Assign, Hide, Disable, Enable and Show are configured. These actions are applied in three scenarios. First, to impact an entire multifield, second, to impact an index from a multifield. Third, to impact a single field in a single row of the multifield.
 - In the **ComboTestMF** multifield, the <Actions> section has a QuerySet that uses the action types OPTIONS and SQLQUERY.
 - In the **TextTestMF** multifield, the <ActionSet> **TestTextActionOnChange** uses MultiValueFieldIndex in Conditions.
 - In the **TextTestMF** multifield, the <ActionSet> **TestTextActionOnChange** uses \$\$\$MultiValueFieldIndex\$\$\$ substitution in an error message.
- The MultiFieldEvents transaction demonstrates the ability for transaction configuration to trigger events on multifields. Navigate to **Global Explorer | Transactions | MultiFieldEvents (Functional Prototype Plan)** to view the prototype configuration. The following event functionality is included in the prototype:
 - <Multifields> element identifies the multifield rule that should be called.
 - In the <Actions> section, OnLoad, OnChange and OnSubmit are configured.

- In the <ActionSets> section, Assign, Hide, Disable, Enable and Show are configured. These actions are applied in three scenarios. First, to impact an entire multifield, second, to impact an index from a multifield. Third, to impact a single field in a single row of the multifield.
 - In the <ActionSet> **TestTextActionOnLoad**, ScreenMath from the transaction is pulled into a multifield.
 - In the <ActionSet> **TestTextActionOnChange**, a transaction field value is assigned to a multifield.
- A set of actions and event is added to AddressScreen, PolicyScreen, ClientScreen, MultiFields-Phones, MultiFields-PrototypePolicyScreen and MultiFields-AccountNumber which is present at the 'PrototypeCompany' level and a new field 'Some Integer' field is added to 'MultiFields-PrototypePolicyScreen' business rule.
 - **AddressScreen:** This business rule is modified by adding 'CallExternalEvent' and 'CalledEvent'. When the user does not add a value to 'PhoneNumber' field which is present in 'MultiFields-Phones' business rule, user will be prompted with a warning 'PhoneNumber is a required field'.
 - **PolicyScreen:** This business rule is modified by adding 'CallExternalEvent' and 'CalledEvent'. When the user does not add a value to 'Some Integer' field which is present in 'MultiFields-PrototypePolicyScreen' business rule, user will be prompted with a warning 'Some Integer' is a required field'.
 - **ClientScreen:** This business rule is modified by adding 'CallExternalEvent' and 'CalledEvent'. When the user does not add a value to 'BankAccountNumber' field which is present in 'MultiFields-AccountNumber' business rule, user will be prompted with a warning 'BankAccountNumber is a required field'.

View Prototype in OIPA

1. Log in OIPA using the Prototype Company user ID and password.
2. Click **Policy | New** from the Main menu.
3. Add a shell policy to test the prototype.
4. Click **Add Activity** on the Secondary menu.
5. Select the MultifieldEvents activity from the Activity drop down box.
6. Make changes to the various fields to view the action event functionality.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Multifield Value in Math and ScreenMath Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Multifield Value in Math and ScreenMath Prototype

The MultifieldEvents prototype demonstrates how OIPA allows picking up the values of Multifield values from specific instances in math variables in the ScreenMath and Math sections in transactions. In addition to the UI Screen, the multifield data can also be updated via transactions.

Prototype Requirements

1. In a specific transaction, the multifield value entered in the second instance of the Multifield is required to be used for validation in screen math and display in the verification screen. The value will need to be picked up in math variables in ScreenMath and Math to perform the same.
2. User should have multifields saved on the screen in order to view the copied value from transaction on to the respective screen like Policy, Role etc.
3. User should be able to add the transaction 'MultifieldEnhancement' on the policy.

Changes to Existing Items

1. RoleScreen present at 'FucntionalPrototypePlan' level is altered by adding the 'Multifield-PrototypeRoleScreen' multifield business rule for the insured role.
2. Existing Multifield-PrototypePolicyScreen is altered by adding the Name attribute into it.

New Items

- A new business rule Multifield-PrototypeRoleScreen is added to showcase this feature.
- A new transaction called "MultifieldEnhancement" is added at the "Functional Prototype Plan" level.

Transaction name: MultifieldEnhancement

Company name: Prototype Individual Child Company

Product Name: Individual Prototype Product

Plan Name: Functional Prototype Plan

Attached BusinessRules: CopyToPolicy, CopyToRole, CreatePolicy

Scenario

In a specific transaction, the multifield value entered in the second instance of the Multifield is required to be used for validation in screen math and display in the verification screen. The value will need to be picked up in math variables in ScreenMath and Math to perform the same.

Prototype Configuration

- The existing transaction 'MultifieldEvents' in the Functional Prototype Plan is used for demonstrating the functionality associated with this feature. The following features are configured to demonstrate the same functionality.
 - A math variable "TextTestMV" of type MultiField is added in the Math section of the transaction which fetches the value from the associated Multifield rule "Multifield-MultifieldEvents" from Multifield "TextTestMF" at index 1. This can be viewed in the Math Variables section after processing the activity.
 - A VerificationScreen BR is attached to this transaction where the above created math variable "TextTestMV" is accessed as a field. This can be viewed during Verify.
 - A ValidateExpression BR is also attached to this transaction which will perform validation based on the math variable "TextTestMV" and display errors as appropriate while processing the transaction.
- A transaction will be given an option to select 'CopyToPolicyFields', 'CopyToRoleFields' and 'CreatePolicy' BR. Depending on the selection the respective business rules will be invoked. The transaction will have a collection of multifield index and a value attached to it. This collection will be copied to the respective screens.
 - **CopyToPolicyFields:** When the user selects 'CopyToPolicyFields' option from the 'SelectOperation' field in the transaction, the transaction copies the collection value to 'PolicyScreen' via CopyToPolicyFields business rule.
 - **CopyToRoleFields:** When the user selects 'CopyToRoleFields' option from the 'SelectOperation' field in the transaction, the transaction copies the collection value to 'RoleScreen' for the Insured Role since the multifields are present for Insured role in the 'RoleScreen.'
 - **CreatePolicy:** When the user selects 'CreatePolicy' option from the 'SelectOperation' field in the transaction, the transaction copies the

collection value to 'PolicyScreen' via CreatePolicy business rule along with the details which is required for creating a policy. Policy will be created at the Functional Prototype Plan Level.

View Prototype in OIPA

1. Log in OIPA using the Prototype Company user ID and password.
2. Click **Policy | New** from the Main menu.
3. Add a shell policy to test the prototype.
4. Click **Add Activity** on the Secondary menu.
5. Select the MultifieldEvents activity from the Activity drop down box.
6. Make changes to the various fields and press verify. The verification screen will display the value entered in the multifield TextTestMF at index 1. Enter any text other than 'New Text' in TextTestMF at index 1 and an error message will also be displayed on the verification screen.
7. Enter value 'New Text' in TextTestMF at index 1 and process the transaction. Click on the activity results and go to the Math tab to see the multifield value from TextTestMF at index 1 being available in the math variable TextTestMV.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > New Allocation Methods Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

New Allocation Methods Prototype

OIPA users now have the option to employ several new allocation methods.

Prototype Prerequisites

- The policy whose allocations are being configured must be in "Pending" status.
- The policy must exist with an Account Value allocated in at least two funds.
- The policy must belong to a plan with the Plan Allocation Method set to "Model Supported."

Prototype Explanation

The following business rules and transactions were configured in the Model Prototype Plan, located within the Prototype Company, to demonstrate this new functionality.

Business Rules

- **PolicyAllocationScreen:** This business rule controls the allocation methods available to a policy. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Model Prototype Plan | Plan Rules | PolicyAllocationScreen** and click on the **XML Source** pane. The key configuration for this business rule is explained below.
 - The list of available funds--configured within the <Funds> element--is configured to make the following funds available:
 - BigPlus Large Cap Growth
 - Finlux Short Treasury Bond Fund
 - GS SmallCap Value
 - Money Market Fund
 - The list of available models--configured within the <Models> element--is configured to make the following models available:
 - LifeStyle Aggressive Portfolio
 - LifeStyle Custom Portfolio
 - The <Allocation> element has the ALLOWMIXEDMETHODS attribute set to "Yes" to allow multiple allocation methods to be used.
 - The <AllocationMethods> element contains <AllocationMethod> elements that make the following allocation methods available:
 - Percent
 - Amount
 - Units
 - Mixed
- **TransactionAllocationScreen:** This business rule controls the allocation methods available to a transaction, and is attached to the

AllocationMethodSupport transaction. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Model Prototype Plan | Plan Rules | PolicyAllocationScreen** and click on the **XML Source** pane. The key configuration for this business rule is explained below.

- The list of available funds--configured within the <Funds> element--is configured to make only funds that have a fund value on the policy available for selection.
- The list of available models--configured within the <Models> element--is configured to make the following models available:
 - LifeStyle Aggressive Portfolio
 - LifeStyle Custom Portfolio
- The <Allocation> element has the ALLOWMIXEDMETHODS attribute set to "Yes" to allow multiple allocation methods to be used.
- The <AllocationMethods> element contains <AllocationMethod> elements that make the following allocation methods available:
 - Percent
 - Amount
 - Units
 - Mixed

Transactions

- **AllocationMethodSupport:** This policy-level financial transaction has the TransactionAllocationScreen rule attached. The attached rule is configured to display an Allocation tab for this transaction, where a user can select either mixed allocation methods or a specific allocation method. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Transactions | ConfirmationPremium** and click on the **Fields** pane. The key configuration for this transaction is explained below.
 - The <Assignment> element has the TYPE attribute set to "Withdrawal," which will cause the transaction to withdraw fund values based on the transaction allocation details.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password.
2. Open a policy in "Pending" status or create a new policy. The policy should belong to a plan that uses the Model Supported allocation method. If creating a new policy, the PremiumWithAllocation activity can be used to apply the premium to multiple funds.
3. Select **Allocations** from the menu on the left side of the screen.
4. Note that each allocation's **Allocation Method** drop-down box has Percent, Amount, Units and Mixed available for selection.
5. Note that if **Mixed** is chosen from the Allocation Method drop-down box, the **Method** drop-down box has Percent, Amount and Units available for selection. If Percent, Amount or Units is selected, note that each field in the **Mixed** column changes to reflect the allocation method chosen.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > New Business Underwriting Requirements Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

New Business Underwriting Requirements Prototype

As part of the New Business Underwriting process, OIPA now supports enhanced requirement functionality in order to facilitate the processing of applications for policies.

Prototype Explanation

Underwriting Guideline

A sample underwriting guideline was created for the purposes of this prototype. See the Requirements section below for an explanation of each requirement used in the guideline.

Initial Underwriting Requirement

Age-Amount	50,000-99,999	100,000-199,999	200,000-999,999	1,000,000-10,000,000
18-30	Non-Medical	Non-Medical Blood and Urinalysis Motor Vehicle Report Paramed Exam	Paramed Exam Blood and Urinalysis Motor Vehicle Report APS Report	Paramed Exam Blood and Urinalysis Motor Vehicle Report Inspection Report APS Report
31-35	Non-Medical	Non-Medical Blood and Urinalysis Motor Vehicle Report Paramed Exam	Paramed Exam Blood and Urinalysis Motor Vehicle Report APS Report	Paramed Exam Blood and Urinalysis Motor Vehicle Report Inspection Report APS Report
36-45	Non-Medical	Non-Medical Blood and Urinalysis Motor Vehicle Report Paramed Exam	Paramed Exam Blood and Urinalysis Motor Vehicle Report APS Report	Paramed Exam Blood and Urinalysis Motor Vehicle Report Inspection Report APS Report
46-55	Non-Medical	Paramed Exam Blood and Urinalysis Motor Vehicle Report APS Report	Paramed Exam Blood and Urinalysis Motor Vehicle Report APS Report EKG Exam	Paramed Exam Blood and Urinalysis Motor Vehicle Report APS Report EKG Exam Inspection Report
56-60	Non-Medical APS Report	Paramed Exam Blood and Urinalysis Motor Vehicle Report APS Report	Paramed Exam Blood and Urinalysis Motor Vehicle Report APS Report EKG Exam	Paramed Exam Blood and Urinalysis Motor Vehicle Report APS Report EKG Exam Inspection Report
61-65	Non-Medical APS Report	Paramed Exam Blood and Urinalysis Motor Vehicle Report APS Report	Paramed Exam Blood and Urinalysis Motor Vehicle Report APS Report EKG Exam	Paramed Exam Blood and Urinalysis Motor Vehicle Report APS Report EKG Exam Inspection Report
	Non-Medical	Paramed Exam Blood and Urinalysis	Paramed Exam Blood and Urinalysis	Paramed Exam Blood and Urinalysis Motor Vehicle Report

66-70	APS Report	Motor Vehicle Report APS Report	Motor Vehicle Report APS Report EKG Exam	APS Report EKG Exam Inspection Report
-------	------------	------------------------------------	--	---

Requirements

The following Policy-Client requirements were configured in Primary Company 1, a subsidiary company to the Holding Company, in order to demonstrate this new functionality.

- **Blood and Urinalysis:** This requirement dictates that a Blood and Urinalysis report is needed for the potential insured. To view the prototype configuration, navigate in the Admin Explorer to **Administration | Requirements | Blood and Urinalysis** and view the XML Source, XML Definition and XML Results panes. The key configuration for this requirement is explained below:
 - The requirement has the MatchRequirementResult screen attached to search for a requirement result with the RequirementResultGUID defined in the RequirementResultGuidMV math variable.
- **Motor Vehicle Report:** This requirement dictates that a Motor Vehicle Report is needed for the potential insured. To view the prototype configuration, navigate in the Admin Explorer to **Administration | Requirements | Motor Vehicle Report** and view the XML Source, XML Definition and XML Results panes. The key configuration for this requirement is explained below:
 - The requirement uses the CopyToRequirementFields to retrieve data for the potential insured, such as the client GUID and driver's license number
 - The requirement has the MatchRequirementResult screen attached to search for a requirement result with the RequirementResultGUID defined in the RequirementResultGuidMV math variable.
- **Paramed Exam:** This requirement dictates that Paramed Exam results are needed for the potential insured. To view the prototype configuration, navigate in the Admin Explorer to **Administration | Requirements | Paramed Exam** and view the XML Source, XML Definition and XML Results panes. The key configuration for this requirement is

explained below:

- The requirement has two fields configured on the XML Results pane:
 - An Information Received checkbox, used to indicate whether the potential insured's Paramed Exam results have been received.
 - A Date Received field.
- **APS Report:** This requirement contains the minimum configuration needed for a requirement to process. To view the prototype configuration, navigate in the Admin Explorer to **Administration | Requirements | APS Report** and view the XML Source, XML Definition and XML Results panes. The key configuration for this requirement is explained below:
 - The requirement is configured with opening and closing `<RequirementDefinition>`, `<Requirement>` and `<RequirementResults>` parent tags on the XML Source, XML Definition and XML Results panes, respectively.
- **Inspection Report:** This requirement dictates that an Inspection Report is needed for the potential insured. To view the prototype configuration, navigate in the Admin Explorer to **Administration | Requirements | Inspection Report** and view the XML Source, XML Definition and XML Results panes. The key configuration for this requirement is explained below:
 - The requirement has two fields configured on the XML Results pane:
 - A Received Date field, used to indicate the date that the potential insured's Inspection Report was received.
 - A Fulfilled By Date date, used to indicate the date that the Inspection Report was fulfilled.
- **EKG Exam:** This requirement dictates that EKG Exam results are needed for the potential insured. To view the prototype configuration, navigate in the Admin Explorer to **Administration | Requirements | EKG Exam** and view the XML Source, XML Definition and XML Results panes. The key configuration for this requirement is explained below:
 - The requirement has the MatchRequirementResult screen attached to search for a requirement result with the RequirementResultGUID defined in the RequirementResultGuidMV math variable.

Transactions

The following transactions were configured in the NBU Term Life Plan, belonging to Primary Company 1, to demonstrate this new functionality:

- **AutoGenerateRequirements:** This transaction automatically adds the Age-Amount requirements, as well as additional requirements, based on the configured criteria. To view the prototype example, navigate in the Main Explorer to **Companies | Holding Company | Subsidiary Company | Primary Company 1 | Plans | NBU Term Life Plan | Transactions | Policy Transactions | AutoGenerateRequirements**. The key configuration for this transaction is explained below:
 - The transaction has the AddRequirements business rule attached, which is configured to add the Age-Amount requirements explained above, as well as several additional requirements, to the application.

Business Rules

- **AddRequirements:** This business rule is attached to the AutoGenerateRequirements transactions in order to add a specified set of requirements to an application. To view the prototype example, navigate in the Main Explorer to **Companies | Holding Company | Subsidiary Company | Primary Company 1 | Plans | NBU Term Life Plan | Transactions | Policy Transactions | AutoGenerateRequirements | Attached Rules | AddRequirements**. The key configuration for this business rule is explained below:
 - The business rule is configured to add the following requirements to an application:
 - APS Report
 - Motor Vehicle Report
 - Inspection Report
 - Blood and Urinalysis
 - Missing Drivers License Information
 - Illustration Details
 - MIB Inquiry

- Missing MIB Authorization
- Prescription Verification
- Aviation Questionnaire
- Travel Questionnaire

View Prototype in OIPA

1. Log in OIPA using the Holding Company user ID and password.
2. Open an existing application or create a new application.
3. Add and process the AutoGenerateRequirements activity.
4. View the application's requirements by clicking on **Requirements** in the navigational menu on the left side of the Application screen.
5. Note that the requirements listed above were added to the application.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Nonreversible-Nonreversing Transactions

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Nonreversible-Nonreversing Transactions

In OIPA, an activity that does not cause the reversal of any other activities and cannot be reversed by another activity, is called a Nonreversible-Nonreversing activity.

These activities are unaffected by the activities that process around them. A Non-reversible-Nonreversing activity can be processed in the midst of multiple activities and if those activities are reversed, it will remain unaffected.

These activities can be deleted from the Activity screen before they are processed, but once processed, the user will have no delete or recycle options. The only way to delete a Nonreversible/Nonreversing activity is to use the **Delete checkbox** on the Activity screen. A user will only be able to click the Delete checkbox if security privileges are assigned to that user.

Scenario

An End of Quarter report needs to be generated. The document needs to be processed on the actual End of Quarter date, but that date occurs during a holiday. OIPA must wait to process the policy information on the next business day. It then backdates the End of Quarter Report for the actual date of the end of the quarter. Backdating the activity does not cause any undo/redo processing.

Configuration Overview

Six transactions were configured to demonstrate how a Nonreversible-Nonreversing activity can spawn a variety of other activity types. When the **PolicyFinancialNonreversible-Nonreversing** activity is processed , it will spawn three additional activities: itself, **PolicyDocumentNonReversibleNonReversing** and **PolicyReversibleReversing**. Keep in mind that even if a spawned activity is a type that can be reversed, since it is a spawn, the delete and recycle icons will not be available.

Navigate to **Global Explorer | Transactions | PolicyFinancialNonreversible-Nonreversing | PolicyFinancialNonreversible-Nonreversing (Functional Prototype Plan)** to view all of the configured transactions.

Explanation of Activity Names

The activity names describe the type and behavior of the activity. There are three parts to each name. **Example:**

PolicyFinancialNonreversible/Nonreversing

1. **PolicyFinancial:** This denotes the transaction type.
 2. **Nonreversible:** This indicates that the activity cannot be reversed by any other activity. It describes the behavior of other activities toward the activity.
 3. **Nonreversing:** This indicates that the activity cannot reverse any other activity. It describes the behavior of the activity toward other activities.
- **PolicyFinancialNonreversible-Nonreversing :** This is the main activity that is configured to spawn itself, a document nonreversible and a policy reversible activity.
 - **PolicyDocumentNonreversible-Nonreversing:** This is a policy document activity, which cannot be deleted or recycled once it is processed.
 - **PlanFinancialNonreversible-Nonreversing:** This is a plan financial activity, which cannot be deleted or recycled once it is processed.
 - **PlanDocumentNonreversible-Nonreversing:** This is a plan document activity, which cannot be deleted or recycled once it is processed.
 - **ClientFinancialNonreversible-Nonreversing :** This is a client financial activity, which cannot be deleted or recycled once it is processed.
 - **ClientDocumentNonreversible-Nonreversing:** This is a client document activity, which cannot be deleted or recycled once it is processed.

Three additional transactions were added to the Functional Prototype Plan to help with the spawning of a regular Reversible/Reversing transaction from the Non-Reversible/Non-Reversing.

- **PolicyReversibleReversing:** This is a policy level activity, which when reversed will reverse other activities and can be reversed by other activities.

If this activity is spawned the delete and recycle icons are no longer available.

- **PlanReversibleReversing:** This is a plan level activity, which when reversed will reverse other activities and can be reversed by other activities. If this activity is spawned the delete and recycle icons are no longer available.
- **ClientReversibleReversing:** This is a client level activity, which when reversed will reverse other activities and can be reversed by other activities. If this activity is spawned the delete and recycle icons are no longer available.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Original Activity Allocation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Original Activity Allocation

OIPA supports the ability to display the original and final allocations for a processed activity on the Activity Results screen. The ActivityResultsScreen business rule contains the configuration that determines whether the original allocations will display. Final allocations will always display.

Scenario

An exchange activity is added in OIPA with amounts to be transferred from one fund to another specified in the activity allocations by the CSR. The activity is processed and the ReassignAllocations rule is triggered. The CSR views the Activity Results screen and is able to see the original allocation amount entered as well as the final allocations after the ReassignAllocation rule has been executed.

Configuration Examples

- An **ActivityResultsMoneyIn** transaction was configured to demonstrate the use of the SHOWORIGINAL attribute in the ActivityResultsScreen business rule, which tells OIPA to show the original allocations. The attribute value is set to Yes to display original allocations. The ActivityResultsScreen is overridden and attached to this transaction. Navigate to **Global Explorer | Transactions | ActivityResultsMoneyIn | ActivityResultsMoneyIn (Functional Prototype Plan)**.
- An **ActivityResultsMoneyOut** transaction was configured to demonstrate the use of the SHOWORIGINAL attribute in the ActivityResultsScreen business rule, which tells OIPA to show the original allocations. The attribute value is set to No to display only the final allocations. The ActivityResultsScreen is overridden and attached to this transaction. Navigate to **Global Explorer | Transactions | ActivityResultsMoneyOut | ActivityResultsMoneyOut (Functional Prototype Plan)**.
- A **ReassignAllocations** rule was attached to the ActivityResultsMoneyIn transaction. This rule drives the Allocations tab of the processed activity and displays the activity allocations as the future allocations. Navigate to **Global Explorer | BusinessRules | Attached | ReassignAllocations | Transaction Overrides | ReassignAllocations (ActivityResultsMoneyIn-Functional Prototype Plan)** .
- A **ReassignAllocations** rule was attached to the ActivityResultsMoneyOut transaction. This rule drives the Allocations tab of the processed activity and displays the activity allocations as the future allocations. Navigate to **Global Explorer | BusinessRules | Attached | ReassignAllocations | Transaction Overrides | ReassignAllocations (ActivityResultsMoneyOut-Functional Prototype Plan)** .

View New Feature in OIPA

To view the prototype example in OIPA:

1. Login and select the Prototype Company and Functional Prototype plan.
2. Create a new policy and add roles, and at least one segment.
3. Click the **Activities** link to open the Activity screen.
4. Click **Add Activity** on the Secondary menu and select the **ActivityResultsMoneyIn** activity.
5. Enter an amount.
6. Click the **Allocations** tab in the Activity window and enter the allocation information.
7. Click **OK** to add the activity to the Activity screen.
8. Click the lightning bolt icon to process the activity.
9. Click the Activity Detail icon to the left of the processed activity. This opens the activity details.
10. Click the **Allocations** tab and see that original allocations and final allocations are listed.
11. Repeat steps 3-10, but use the **ActivityResultsMoneyOut** activity.
12. Notice on the second activity that final allocations are displayed in the Allocation tab of the Activity Results screen.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Overridable Error Authorization Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Overridable Error Authorization Prototype

OIPA's `ValidateExpressions` and `PostAssignmentValidateExpressions` business rules now support the ability to configure the security roles that are able to override specific errors.

Prototype Explanation

The following transactions and business rules were configured in the Model Prototype Plan, located within the Prototype Company, to demonstrate this new functionality.

Transactions

- **SecurityOverrideForPAVEErrors:** This transaction is configured to return errors based on the value entered in the "Amount" field. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Transactions | SecurityOverrideForPAVEErrors**. The key configuration for this transaction is explained below.
 - An "Amount" field is configured to accept an integer value.
- **SecurityOverrideForVEErrors:** This transaction is configured to return errors based on the value entered in the "Amount" field. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Transactions | SecurityOverrideForPAVEErrors**. The key configuration for this transaction is explained below.
 - An "Amount" field is configured to accept an integer value.

Business Rules

- **ValidateExpressions:** This business rule is attached to the SecurityOverrideForVEErrors transaction. It contains configuration that specifies whether errors returned from the transaction to which it is attached are able to be overridden, as well as the specific errors that are able to be returned from the transaction. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Model Prototype Plan | Business Rules | Attached Rules | ValidateExpressions**. The key configuration for this business rule is explained below.
 - The <Expressions> element has its OVERRIDABLE attribute set to "Yes," which designates that errors returned from the transaction are able to be overridden.

- The <Expressions> element's ERRORNUMBER attribute contains two error numbers. These error numbers will appear in the Overridable Errors section of the security group's Transaction Security page.
- **PostAssignmentValidateExpressions:** This business rule is attached to the SecurityOverrideForPAVEErrors transaction. It contains configuration that specifies whether errors returned from the transaction to which it is attached are able to be overridden, as well as the specific errors that are able to be returned from the transaction. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Model Prototype Plan | Business Rules | Attached Rules | PostAssignmentValidateExpressions**. The key configuration for this business rule is explained below.
 - The <Expressions> element has its OVERRIDABLE attribute set to "Yes," which designates that errors returned from the transaction are able to be overridden.
 - The <Expressions> element's ERRORNUMBER attribute contains two error numbers. These error numbers will appear in the Overridable Errors section of the security group's Transaction Security page.

Security Groups

- **Prototype Super:** This Security Group is configured to be able to override ORY001, but not ORY002 (see the Error Numbers section for an explanation of each error number).
- **Prototype Tester:** This Security Group is configured to be able to override both errors (see the Error Numbers section for an explanation of each error number).

Error Numbers

- **ORY001:** This error will be returned if the user enters a value less than 1000 in the "Amount" field.
- **ORY002:** This error will be returned if the user enters a value less than 500 in the "Amount" field.

View Prototype in OIPA

1. Log in to OIPA using credentials for a user belonging to the Prototype Super security group.
2. Open a policy belonging to the Model Prototype Plan.
3. Add the SecurityOverrideForVEErrors activity.
4. Enter a value under 500 in the Amount field. The system will display error number ORY002 without the ability to override.
5. Enter a value under 1000 in the Amount field. The system will display error number ORY001 with the ability to override.
6. Log out of OIPA and log back in using credentials for a user belonging to the Prototype Tester security group.
7. Repeat steps 2 through 5 above. This time, both errors will be overridable.

You are here: Configuration > Valuation > Point in Time Valuation > Point In Time Configuration

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Point In Time Valuation Prototype

Point In Time valuation allows users to control how valuation is calculated during transaction processing. This prototype demonstrates how Point In Time valuation can be configured. There are two business rules used: PointInTimeValuation and WriteValuationElements.

At the plan level, the PointInTimeValuation business rule is configured to indicate whether the beginning of the activity saves valuation records and the condition upon which the end of the activity saves valuation records. This rule also directs the display of a beginning valuation area on the Activity Results screen of an activity. If needed, the beginning valuation can be set to **No** and the ending valuation can be set to only save when there is a monetary change in any fund.

The WriteValuationElements business rule has the ability to write to the database extra data associated to the valuation. Configuration is divided into four sections: Policy, Fund, Deposit, and PolicyValues. In these sections, a set list of elements are the only elements that can be written. The one exception to this is the Policy Values section, which can be configured to access any variables from the PolicyValues business rule. CopyBooks may be used in this rule.

Scenarios

Two scenarios can be created to demonstrate this functionality.

- User adds an activity to a policy that applies money to a fixed fund and performs valuation.
- User adds an activity to a policy that values the policy, but does not add money.

Configuration Requirements

On the AsPlan table the PointInTimeValuation column must be set to a value of Y(or T if the plan is transitioning from Traditional valuation) in order to use the PointInTimeValuation business rule. The additional configuration required includes:

- PointInTimeValuation business rule
- WriteValuationElements business rule
- transactions to demonstrate functionality

If a plan is being converted from Traditional to Point-in-Time valuation, the setting for the PointInTimeValuation column of AsPlan depends on the specific type of conversion. For details, see Point in Time Conversion.

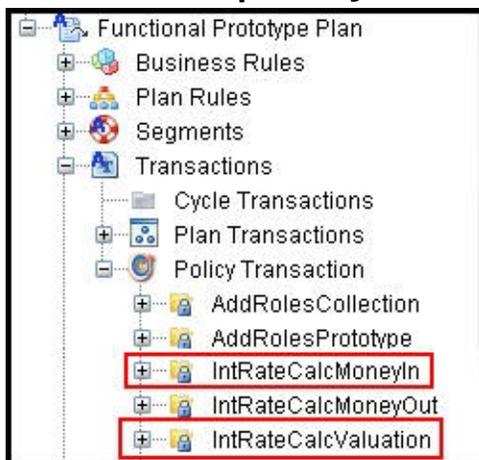


Prior to configuring Point-in-Time valuation, ensure that basic fund valuation is already fully working in your environment. Funds, allocations, and assignments must already be setup and configured before attempting to enable Point-in-Time valuation.

Prototype Samples

Two transactions have been configured to demonstrate the enhancements to the system.

- IntRateCalcMoneyIn transaction: This transaction allows the user to apply money to variable or fixed funds and perform valuation. Slight changes can be made to the plan level overrides of the PointInTimeValuation business rule to limit the amount of valuation data being saved to the database. Navigate through the following folders in the Main Explorer to locate the configuration sample: **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | IntRateCalcMoneyIn.**
- IntRateCalcValuation transaction: This transaction does not apply any money, but does value the policy. Navigate through the following folders in the Main Explorer to locate the configuration sample: **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | IntRateCalcValuation.**

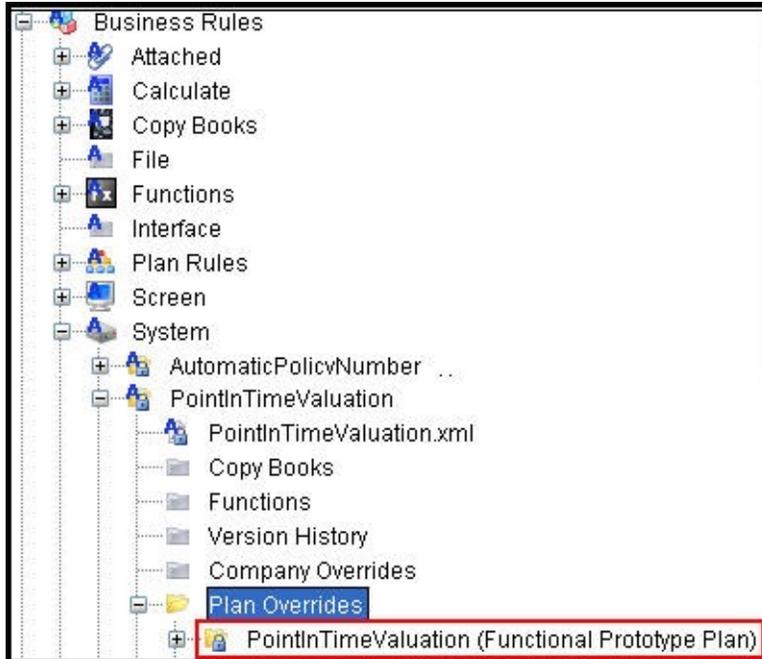


PointInTime Transactions in Main Explorer

Two business rules have been configured to demonstrate the enhancements to the system.

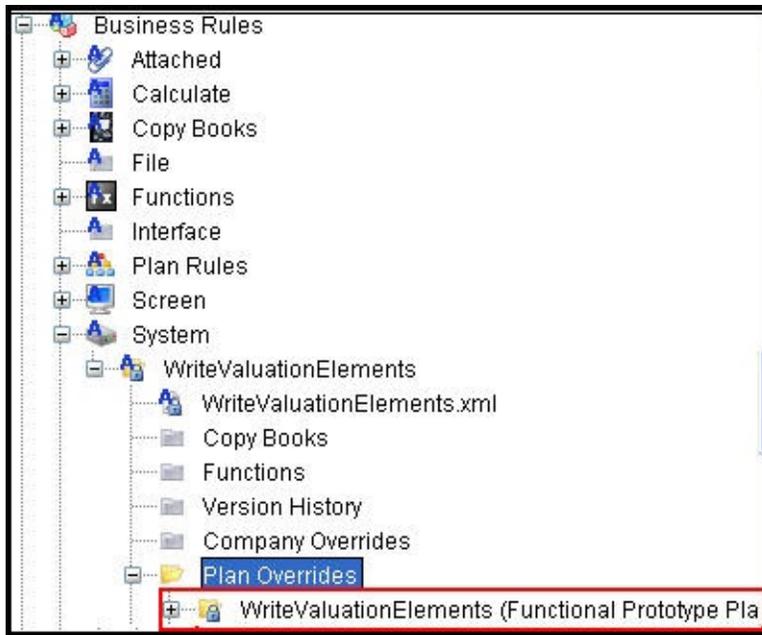
- PointInTimeValuation: This rule is configured as a plan level override and indicates when and which valuation records are saved to the database. In addition, the rule directs the display of a beginning valuation area in the Valuation tab of Activity Results. Navigate

through the following folders in the Global Rules Explorer to locate the configuration sample: **Business Rules | System | PointInTimeValuation | Plan Overrides.**



PointInTimeValuation Business Rule in Global Rules Explorer

- WriteValuationElements: This rule can write additional valuation data to the database. Navigate through the following folders in the Global Rules Explorer to locate the configuration sample: **Business Rules | System | WriteValuationElements | Plan Overrides.**



WriteValuationElements Business Rule in Global Rules Explorer

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Policy Overview
Screen Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Policy Overview Screen Prototype

The Policy Overview screen provides a read only summary of policy details such as policy roles, segments, segment roles and policy values. The prototype configuration demonstrates the use of CopyBooks in the PolicyOverviewScreen rule and the new PolicyScreen button for PolicyOverviewScreen. It also demonstrates the use of field masking in the Policy Details section.

The "PolicySummary" section present at the end of the PolicyScreen provides the consistency between all the Policy related screens.

Scenario

The PolicyScreen <Button> element **PolicyOverview** is present in configuration and the Policy Overview screen has been configured. As a result, the Policy Overview screen is the first item in the Policy Left Navigation menu and is the default view when a user opens a policy.

Prototype Examples

- An plan level override of the **PolicyScreen** business rule was created for the Functional Prototype Plan. In the <Buttons> section, a button was added for Policy Overview. Navigate to **Global Explorer | Business Rules | Plan Rules | PolicyScreen | Plan Overrides |PolicyScreen(Functional Prototype Plan)**.
- A plan level override was created for the **PolicyOverviewScreen** rule. Navigate to **Global Explorer | Business Rules | Screen Rules | PolicyOverviewScreen | Plan Overrides |PolicyOverviewScreen(Functional Prototype Plan)**.
 - **Policy Details:** CopyBook functionality is demonstrated in the Policy Details section. A field mask is applied to the Tax ID field. OnLoad configuration enables the field displayed as Bonus Need if the field Interest Bonus Qualified has a value of Yes. The field displayed as Bonus Reason is presented when the field displays as Interest Bonus Qualified has a value of Yes. The field displayed as Bonus Reason is hidden when the field displays as Interest Bonus Qualified has a value of No.
 - **Policy Roles:** Within <PolicyDetails>, <PolicyRoles SHOW="Yes"></PolicyRoles> is configured. This syntax results in a grid displaying all policy roles associated with the policy. This grid is located in the Policy Roles screen section.
 - **Segments:** Using the SHOW="Yes" attribute in the <Segments> element, segments are visible in Policy Segments screen section.
 - **Segment Roles:** Using the SHOW="Yes" attribute in the <SegmentRoles> element, segment roles are visible in the Segment Roles screen section. Segment roles are associated with their respective segment via an index number displayed in both the segment and segment role grids.
 - **Values:** Using the SHOW="Yes" attribute in the <Values> element, the Values section of the screen is visible with a Calculate button.
- The **PolicySummarySection** is added into the configuration of

PolicyScreen override present at the Functional Prototype Plan Level. The PolicySummary section will have the following fields configured to provide consistency between all the Policy screens.

Company: This field will not have 'Expandedattribute.

Product: This field will not have 'Expandedattribute.

Plan: This field will not have 'Expandedattribute.

PolicyNumber: This field will not have 'Expandedattribute.

PolicyName: This field will not have 'Expandedattribute.

IssueStateCode: This field will not have 'Expandedattribute.

The label for this field is modified in the fixedfields section.

Result : User should be able to see PolicySummarySection in the Policy Overview, Policy Role, Policy Activity, Policy Requirement Screen, Policy Segment, Policy Values, Policy Allocations, Policy Withholding, Policy Programs and Policy Comments.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Post Assignment Math Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Post Assignment Math Prototype

This prototype demonstrates the ability to log post assignment math and the ability to use an Identifier field type in the PostAssignmentValidateExpressions rule. A math variable in the transaction will be logged to validate that the ability to log activity math was not impacted by the changes to post assignment math.

Configuration Explanation

- A **RateRetrieval** transaction has the attached **PostAssignmentValidateExpressions** business rule. Using **LOG="YES"** syntax, the value of a math variable named **TobaccoMV** will be logged. Navigate to **Global Explorer | Transactions | RateRetrieval | RateRetrieval (Functional Prototype Plan)**.
- A **PostAssignmentValidateExpressions** business rule was attached to the **RateRetrieval** transaction. This rule demonstrates that math variables can be logged, using the **PAVERateDescriptionMV** math variable. It also demonstrates that an Identifier math variable named **SMMBDayMV** was created and logged. Navigate to **Global Explorer | Transactions | RateRetrieval | RateRetrieval (Functional Prototype Plan) | Attached Rules | PostAssignmentValidateExpressions**.

View Prototype in OIPA

1. Login and select the Prototype Company and Functional Prototype plan.
2. Create a new policy. No additional policy information, such as roles or segments, is needed.
3. Click the **Activities** link to open the Activity screen.
4. Click **Add Activity** on the Secondary menu and select the **RateRetrieval** activity.
5. Enter an age greater than 18 and less than 65 in the IssueAgeFLD field.
6. Click **OK**.
7. Click the lightning bolt icon next to the activity on the Activity screen. A warning icon will stop processing.
8. Click the red flag icon to override the warning. When the window opens, click both Override boxes and click OK.
9. Click the lightning bolt icon again to process the activity.
10. Click the Activity Detail icon to the right of the processed activity.
11. Click the Math link at the top of the window.
12. In the Calculated Fields section, notice that the **TobaccoMV** is listed.
13. Scroll down to the Post Assignment Calculated Fields section and notice the **SMMBDayMV** is also listed.

The Activity Results Math tab of a processed activity now shows both Activity math and Post Assignment math.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > PostAssignmentValidateExpressions Enhancements Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

PostAssignmentValidateExpressions Enhancements Prototype

New functionality has been added to allow values from the PostAssignmentValidateExpressions business rule to be accessible on the Verification screen, Activity Summary and Transaction Cosmetics.

Prototype Prerequisites

- NUVs must exist for the funds that are selected for allocation for the processing of the transaction. Otherwise, the transaction will go into NUV Pending status upon processing.
- The error number DCM666 must exist in the AsErrorCatalog database table.

Prototype Explanation

The following transactions and business rules were configured in the Model Prototype Plan in the Prototype Company to demonstrate this new functionality.

Transactions

- **PremiumWithAllocation:** This transaction is configured to allow the user to enter a premium amount and then to select the funds to which this amount should be allocated. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Model Prototype Plan | Transactions | PremiumWithAllocation**. The key configuration for this transaction is explained below.
 - The <Field> section is configured for the Premium Amount field, which allows the user to enter the amount of the policy's premium.

Business Rules

- **PostAssignmentValidateExpressions:** This business rule is attached to the PremiumWithAllocation transaction. It contains configuration to calculate the policy's existing cash value using a math variable, as well as to generate error or warning messages in certain situations. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Model Prototype Plan | Business Rules | Attached Rules | PostAssignmentValidateExpressions**. The key configuration for this business rule is explained below.
 - The <MathVariables> section contains configuration for a math variable called CurrentCashValue, which is calculated by adding the policy's existing cash value to the premium amount entered by the user.
 - The <ValidateExpressions> section contains configuration for two validations:
 - The first expression throws an error if the policy has no existing cash value. The OVERRIDABLE attribute is set to "Auto",

meaning that the error will be automatically overridden. The client number of the user who views the error window and presses "OK" will be recorded against the overriding user in the activity XML.

- The second expression displays a warning message if the premium amount entered by the user is greater than 5000. This expression includes an error number, on which the error message and Error Fix Tip shown on the Verification screen are based.

Note: The following business rules have been configured to show the accessibility of the PostAssignmentValidateExpressions math variable across these business rules.

- **VerificationScreen:** This business rule is attached to the PremiumWithAllocation transaction To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Model Prototype Plan | Business Rules | Attached Rules | VerificationScreen**. The key configuration for this business rule is explained below.
 - A field is configured to access the math variable CurrentCashValue, which is configured in the PostAssignmentValidateExpressions attached rule.
- **ActivitySummary:** This business rule is attached to the PremiumWithAllocation transaction To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Model Prototype Plan | Business Rules | Attached Rules | ActivitySummary**. The key configuration for this business rule is explained below.
 - A field is configured in the <MathandFields> section to access the math variable CurrentCashValue, which is configured in the PostAssignmentValidateExpressions attached rule.
- **TransactionCosmetics:** This business rule is attached to the PremiumWithAllocation transaction To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Model Prototype Plan | Business Rules | Attached Rules | TransactionCosmetics**. The key configuration for this business rule is

explained below.

- A field is configured in the <AmountField> section to access the math variable CurrentCashValue, which is configured in the PostAssignmentValidateExpressions attached rule.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password.
2. Open a policy or create a shell policy.
3. Add the PremiumWithAllocation activity and enter a premium amount over 5000.
4. Process the activity. Note that a warning appears alerting the user to the amount of the premium.
5. View the Verification screen, Activity Summary and/or Transaction Cosmetics for the activity and note that the value from the CurrentCashValue math variable in the ValidateExpressions rule is utilized.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Precision Values
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Precision Values Prototype

The PrecisionValues business rule provides rounding precision convention (i.e. number of digits allowed after the decimal point) for the display of Unit Values, and for the calculation and display of Number of Units. Unit Values and Number of Units are displayed on the following OIPA screens:

- Policy Screen's Values link
- Activity Result Screen's Allocation link
- Activity Result Screen's Valuation link PrecisionValue business rule

Scenario

A CSR processes an activity that performs money movement in unit capable funds. After the activity processes, the CSR clicks on the Activity Detail icon to open the Activity Results window. The Allocation link is clicked to see the display of units and unit values. The Valuation link is also clicked within the window to see the display of units and unit values. The CSR then navigates to the Values screen to see the display of units and unit values. All values are displaying as defined in the PrecisionValues rule overrides.

Prototype Examples

Each of the components below was configured to determine how units or unit values display in OIPA.

- The **IntRateCalcMoneyIn** transaction was configured and added to the Dynamic Prototype Plan and the Functional Prototype Plan. The reason for choosing two separate transactions, within two separate plans, is to better demonstrate how the PrecisionValues rule operates across plans and companies, showcasing its ability to provide different rounding and display conventions. Navigate to **Global Explorer | Transactions | IntRateCalcMoneyIn** .
- The **Variable Fund 1** fund was associated to the Functional Prototype Plan and the Dynamic Prototype Plan. Navigate to **Main Explorer | Companies | Subsidiary Companies | Prototype Child Company | Plans | Dynamic Prototype Plan (or Functional Prototype Plan) | Funds**.
- The **PrecisionValues** rule was overridden at two levels: Functional Prototype Plan and Prototype Child Company. The Functional Prototype Plan override establishes precision for Unit Values. The Prototype Child Company override establishes precision for Units. Navigate to **Global Explorer | Business Rules | System | Precision Values | Company Overrides | PrecisionValues (Prototype Child Company)** or navigate to **Global Explorer | Business Rules | System | Precision Values | Plan Overrides | PrecisionValues (Functional Prototype Plan)**.
- The **ValueScreen** rule was overridden at two levels to display various valuation items. The overrides can be found in the Dynamic Prototype Plan and the Functional Prototype Plan. Navigate to **Global Explorer | Business Rules | Plan Rules | ValuesScreen | Plan Overrides | ValuesScreen (Functional Prototype Plan)** or navigate to **Global Explorer | Business Rules | Plan Rules | ValuesScreen | Plan Overrides | ValuesScreen (Dynamic Prototype Plan)**.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Priors and Suspects Inquiry Screen Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Priors and Suspects Inquiry Screen Prototype

Prototype Explanation

The following entities were configured to demonstrate this new functionality.

Inquiry Screens

Priors Suspects: This inquiry screen will search across all companies for prior or suspect policies. The results of the search will display in a table, where they are identified and ordered by type: PRIOR or SUSPECT. An attribute 'Total' is added to the output column specification in the Inquiry screen, which will be the sum of the results of specific numeric data columns.

To view the prototype configuration, navigate in the Main Explorer to **Companies | Holding Company | Inquiry Screens | Main Menu | Priors Suspects.**

Configuration Details

The key configuration for this inquiry screen is described below.

- Four searchable fields are configured: First Name, Last Name, Tax ID and Date of Birth.
 - Values are required for all four fields. A wildcard ("%") may be used in the First Name, Last Name and Tax ID fields.
 - If the search finds a prior or suspect policy with a matching First Name, Last Name or Date of Birth, the policy must also have a match on a second field to be returned from the search. If a policy is found with a matching Tax ID, a match on a second field is not required for the policy to be returned.
- The search results table is configured with the following columns:
 - **Type**—PRIOR or SUSPECT
 - **Company**—Holding Company, Prototype Company, etc.
 - **Policy Number**—The policy number of the prior or suspect policy
 - **Insured Name**—The individual's First Name and Last Name
 - **Tax ID**—Social Security Number or Tax ID
 - **Date of Birth**—The individual's date of birth
 - **Policy Role**—Insured, Primary Beneficiary, etc.
 - **Issue Date**—The date of policy issue
 - **Status**—Active, Pending, etc.
- Face Amount InquiryScreen, which is present at the ClientLevel of Prototype Company is altered. A New attribute "Total="yes" is added to the column 'Segment Amount' in the table element of the output section.

Result: User should be able to see the total of the entire segment amount at the end of the column.

View Prototype in OIPA

1. Log in to OIPA with a username and password for the Holding Company.
2. Select **Inquiry** from the Main Menu, then click on **Priors Suspects**.
3. Enter search criteria in all four of the configured fields and click **Ok**.
4. Note that the search results are displayed in a table containing the configured columns.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Programs Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Programs Prototype

A program can be set up to run on a policy or specific segment for a predetermined period of time. It includes attributes that define the program instance and as a result, activities can be run over time according to a specific schedule. Examples of such programs include asset rebalancing, automatic investment plans, systematic withdrawals and cost of living adjustments.

Program Configuration

Three programs were configured on the Functional Prototype Plan to demonstrate program functionality.

- **Program A:** This is a policy program linked to the Functional Prototype Plan and the Rider with Roles segment. Navigate to the **Global Explorer | Programs | Program A | ProgramDefinition.xml**. The ProgramDefinition contains the following configuration.
 - Fields: In addition to the fixed Program fields, the following fields have also been configured. ProgramStartDate (Fixed field), ProgramEndDate (Fixed Field), ProgramEffectiveDate (FixedField), ClientName (dynamic field), ProgramFrequency (dynamic field), ProgramAmount (dynamic field), NextScheduledDate (Disabled field). This date is disabled and updated through CopyToProgram Fields on certain activities).
 - Validations: there are three validations added to the validations section.
 - Transactions: there are four transactions configured for Program A.
 - **SetUpProgram:** This transaction is created when the green Start icon is clicked on the Program screen for a new program. All fields are pre-populated with the exception of the TestText field. This transaction will spawn the PolicyProgramAScheduledActivity activity, with a spawn date of the next business day. There is one attached business rule, ResetProgram, which will move the program status back to PendingReady upon reversal of the SetUpProgram activity.
 - **PolicyProgramAScheduledActivity:** This activity is created and spawned from SetUpProgram. All fields are disabled with the exception of the TestText field. Math will determine if the activity effective date is greater than or equal to the Program End Date. If it is greater, then the TerminateProgram activity will generate and the program will be terminated upon the successful processing of the TerminateProgram activity. The TerminateProgram attached business rule causes this to occur. If the date is not greater, then the PolicyProgramAScheduledActivity will spawn on the next business day. The next spawn date is copied to this field.

- **TerminateProgram:** This activity is added to a program when the PolicyProgramAScheduledActivity effective date is greater than or equal to the Program End date. The ACTION=Terminate attribute must be present in configuration. Processing the activity will end the program. The Grace activity will also cause the TerminateProgram activity to generate.
- **ReinstateProgram:** When the IntRateCalcMoneyIn activity is processed, a program can be reinstated if the program was suspended (08). The ReinstateProgram rule must be attached to the activity. In Program A, the ReinstateProgramSkip activity is generated. This activity determines through configuration that the activity PolicyProgramAScheduledActivity will not be added and processed during the period of suspension. ReinstateProgramSkip will spawn PolicyProgramAScheduledActivity on the next business day.

There is one plan and one segment linked to this program.

- **PolicyProgram:** The Add and Maintain operations are both available for this program. One test condition also exists for this program. If a policy's Issue State Code is 03, for Arizona, then programs cannot be added. Navigate to the **Global Explorer | Programs | Program A | Plan Programs | Functional Prototype Plan | PolicyPrograms.xml**.
- **SegmentProgram:** The Both operation identifies that the Add and Save buttons will be available for programs. One test condition also exists for this program. If a policy's Issue State Code is 05, for California, then programs cannot be added. Navigate to the **Global Explorer | Programs | Program A | Segment Programs | Rider With Roles | SegmentPrograms.xml**
- **Program B:** This is a policy program linked to the Functional Prototype Plan only. The Both operation identifies that the Add and Save buttons will be available for programs. One test condition also exists for this program. If a policy's Issue State Code is 05, for California, then programs cannot be added. Navigate to the **Global Explorer | Programs | Program B | ProgramDefinition.xml**.

- Fields: In addition to the fixed Program fields, the following fields have also been configured. ProgramStartDate (Fixed field), ProgramEndDate (FixedField), ProgramEffectiveDate (FixedField), ClientName (dynamic field), ProgramFrequency (dynamic field), ProgramAmount (dynamic field), NextScheduledDate (Disabled field. This date is disabled and updated through CopyToProgram Fields on certain activities).
- Validations: there are three validations added to the validations section.
- Transactions: there are four transactions configured for Program B.
 - **SetUpProgram:** This transaction is created when the green Start icon is clicked on the Program screen for a new program. All fields are pre-populated with the exception of the TestText field. This transaction will spawn the PolicyProgramAScheduledActivity activity, with a spawn date of the next business day. There is one attached business rule, ResetProgram, which will move the program status back to PendingReady.
 - **PolicyProgramBScheduledActivity:** This activity is created and spawned from SetUpProgram. All fields are disabled with the exception of the TestText field. Math will determine if the activity effective date is greater than or equal to the Program End Date. If it is greater, then the TerminateProgram activity will generate and the program will be terminated. The TerminateProgram attached business rule causes this to occur. If the date is not greater, then the PolicyProgramBScheduledActivity will spawn on the next business day. The spawn date is copied from the CopyToProgramFields attached rule.
 - **TerminateProgram:** This activity is added to a program when the PolicyProgramBScheduledActivity effective date is greater than or equal to the Program End date. The ACTION=Terminate attribute must be present in configuration. Processing the activity will end the program. The Grace activity will also cause the TerminateProgram activity to generate.
 - **ReinstateProgram:** When the IntRateCalcMoneyIn activity is processed, a program can be reinstated if the program was suspended (08). The ReinstateProgram rule must be attached to

the activity. In Program B, the ReinstatementProgramProcess activity is generated. This activity determines through configuration that the activity PolicyProgramBScheduledActivity will be added and processed during the period of suspension. There is the possibility that multiple PolicyProgramBScheduledActivity activities generate. For example if the Program suspended on 1/1/2012 and the ReinstatementProgramProcess transaction is processed on 1/10/2012, 10 PolicyProgramBScheduledActivity activities will be generated. This will make up for the time the Program was in suspension. They all will have the same effective date. The ReinstatementProgramProcess activity will spawn PolicyProgramBScheduledActivity activity/activities with an effective date on the next business day.

There is one plan linked to this program.

- **PolicyProgram:** The Add and Maintain operations are both available for this program. One test condition also exists for this program. If a policy's Issue State Code is 05, for California, then programs cannot be added. Navigate to the **Global Explorer | Programs | Program B | Plan Programs | Functional Prototype Plan | PolicyPrograms.xml**
- **Program C:** This is a segment program linked to the Rider with Roles segment only. The Both operation identifies that the Add and Save buttons will be available for programs. One test condition also exists for this program. If a policy's Issue State Code is 05, for California, then programs cannot be added. Navigate to the **Global Explorer | Programs | Program C | ProgramDefinition.xml**.
 - **Fields:** In addition to the fixed Program fields, the following fields have also been configured. ProgramStartDate (Fixed field), ProgramEndDate (FixedField), ProgramEffectiveDate (FixedField), ClientName (dynamic field), ProgramFrequency (dynamic field), ProgramAmount (dynamic field), NextScheduledDate (Disabled field. This date is disabled and updated through CopyToProgram Fields on certain activities).
 - **Validations:** there are three validations added to the validations section.
 - **Transactions:** there are three transactions configured for Program C.

- **SetUpProgram:** This transaction is created when the green Start icon is clicked on the Program screen for a new program. All fields are pre-populated with the exception of the TestText field. This transaction will spawn the SegmentProgramCScheduledActivity activity, with a spawn date of the next business day. There is one attached business rule, ResetProgram, which will move the program status back to PendingReady.
- **SegmentProgramCScheduledActivity:** This activity is created and spawned from SetUpProgram. All fields are disabled with the exception of the TestText field. Math will determine if the activity effective date is greater than or equal to the Program End Date. If it is greater, then the TerminateProgram activity will generate and the program will be terminated. The TerminateProgram attached business rule causes this to occur. If the date is not greater, then the SegmentProgramCScheduledActivity will spawn on the next business day. The spawn date is copied from the CopyToProgramFields attached rule.
- **TerminateProgram:** This activity is added to a program when the SegmentProgramCScheduledActivity effective date is greater than or equal to the Program End date. The ACTION=Terminate attribute must be present in configuration. Processing the activity will end the program. The Grace activity will also cause the TerminateProgram activity to generate.

There is one segment linked to this plan.

- **SegmentProgram (Rider With Roles):** The Both operation identifies that the Add and Save buttons will be available for programs. One test condition also exists for this program. If a policy's Issue State Code is 05, for California, then programs cannot be added. Navigate to the **Global Explorer | Programs | Program C | Segment Programs | Rider With Roles | SegmentPrograms.xml**.

Two new code names were added to AsCode.

- **AsCodeProgramType:** this user defined code identifies the types of

programs. The TYPE attribute in the [ProgramScreen](#) business rule references this code name and its values.

- **AsCodeProgramStatus:** this system code identifies program statuses. It is referenced in the <ProgramPriorStatusCode> element in the [ProgramDefinition](#) rule, when identifying a program's status prior to suspension.

Several attached rules were configured and attached to the transactions referenced in the programs above.

- **CopyToProgramFields:** this attached rule updates dynamic disabled program fields. It can only be attached to a program transaction.
- **SuspendProgram:** this attached rule creates the SuspendProgram activity, which suspends the program when successfully processed.
- **ResetProgram:** this attached rule returns the program to a Pending Ready status when the SetUpProgram activity is reversed.
- **TerminateProgram:** this attached rule creates the TerminateProgram activity, which terminates the program when successfully processed.
- **ReinstateProgram:** this attached rule creates the ReinstateProgram activity, as defined in ProgramDefinition. A program can be reinstated from Suspend status.

Transaction Configuration

The Program Configuration section above shows how the transactions are associated to the program. The information in this section explains the configuration driving the transactions.

- **SetUpProgram:** This transaction was used in all three programs. It spawns the scheduled activity for each program. The configuration demonstrates how scheduling is controlled. The Function - NYOKWithDirection is used to trigger the spawning of the scheduled activities on the next business day. The ResetProgram rule is attached to this transaction.
- **PolicyProgramAScheduledActivity, PolicyProgramBScheduledActivity and SegmentProgramCScheduledActivity:** Using Function-NYOKWithDirection, each scheduled activity is configured to spawn on the next business day. The attached CopyToProgramFields rule copies the NextBusinessDay math variable to the NextScheduledDate on the Program screen. The TransactionBusinessRulePacket uses rule IF syntax to determine when the TerminateProgram rule should execute.
- **TerminateProgram:** this transaction will terminate any program on the policy after it is processed.
- **Grace:** this existing transaction was modified to spawn TerminateProgram. It adds a TerminateProgram transaction override to Grace demonstrating the use of the rule attached to a non-program activity. TerminateProgram will terminate any programs on the policy.
- **Reinstate:** this transaction was configured in two different ways to demonstrate how a program is reinstated and what happens to program activities that were suspended. This transaction spawns the scheduled program activity. The transaction has the ReinstateProgram attached rule overridden at the transaction level. The ReinstateProgram rule will reinstate the program and generate a ReinstateProgramSkip or ReinstateProgramProcess based on the behavior defined in the ProgramDefinition.

View Prototype Example in OIPA

1. Log in OIPA using the Prototype Company user ID and password.
2. Click **Policy | New** from the Main menu.
3. Type the new policy information and save the record.
4. Click **Programs** from the Left Navigation menu.
5. Add a program and save.
6. Click the start icon in order to create the SetUpProgram activity.
7. Click **Activities** to see the program activities.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Required Fields

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Required Fields

Required fields are now visually identified in OIPA. This allows a user to easily see the fields that must be filled in when adding information on a screen or in an activity. If a required field is not populated, then an error message will display.

Scenario

A user creates a new policy. The configuration dictates that the Tax ID field is required. The Tax ID field on the Policy screen has an asterisk next to it so the user can identify it as required. The user enters the policy information but leaves this field blank. After the user clicks Save, OIPA generates an error stating that the required field must be completed.

Prototype Example

- A **RequiredFields** transaction was configured with an example of each type of required field. Each field has the <Required> element with a value of **Yes**. When the user tries to add this activity, OIPA will make sure each required field contains a value. If the user did not enter a value in a required field when adding the activity, then an error message will appear. Navigate to **Global Explorer | Transactions | RequiredFields | RequiredFields (Functional Prototype Plan)** to view this configuration example.
- A translation was added to the Translation Table for GlobalContext.RequiredFieldError with a value of **Required field(s) cannot be blank**.



When configuring required fields, refer to the **XML Configuration Guide**, which is available from the **Help** menu. Navigate to **Configuration Overview | Configuration Standards | General Structure and Best Practices**. This section lists all special considerations that should be reviewed before configuring required fields.

REQUIRED FIELDS - 1112

Details

NEW ACTIVITY

Activity: Required Fields

ACTIVITY DETAILS

Effective Date: 1/29/2031

Required Combo Field *		Required Date Field *	
Required Money Field *	USD	Required Text Field *	
Required Percent *		Required Decimal Field *	
Required Client Field *		Required Integer Field *	

Required fields identified with an asterisk.

Ok Cancel

Required Fields Identified in OIPA

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > AddRoles Business Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Add Roles Business Rule

In the course of policy processing there may be a need to add new roles to a policy while the Role screen's functionality is disabled. In such cases, a transaction may be used to perform the task. The AddRoles business rule can be attached to a policy-level transaction in order to add an existing client in the database to a new role on an existing policy.

The AddRoles rule can only be attached to a policy-level transaction and can only be used to add roles to the policy where the activity is being processed. Multiple role records can be added to the policy using this business rule.

Scenario

A request to add a new beneficiary needs to be processed. Since the policy is in Active status, the Role screen has been disabled. To add a new role, the CSR selects the AddRolePrototype activity after clicking the Add Activity link. The new beneficiary information is added to the activity window and the activity is added to the Activity screen. When the activity is processed the new beneficiary information is added to the policy.

.

Configuration Requirements

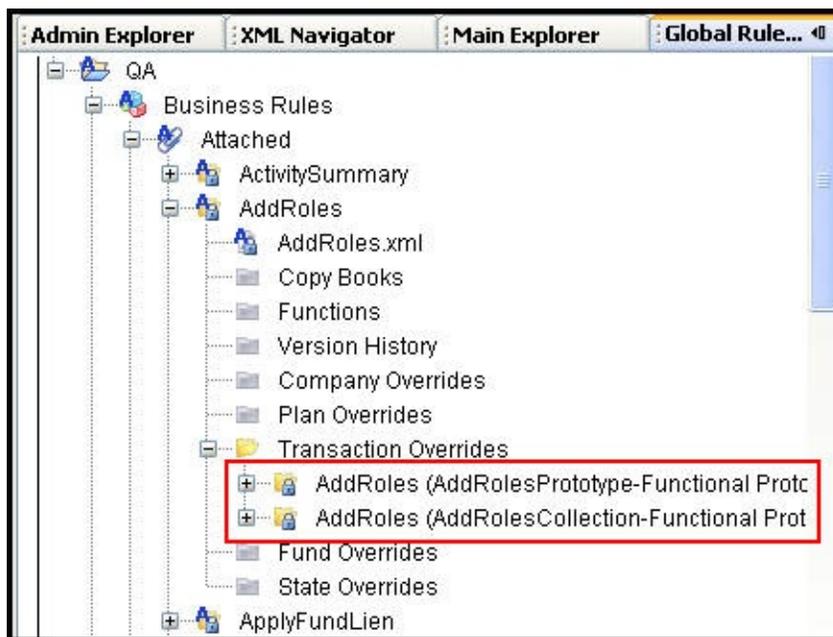
In order to use this business rule to perform the scenario listed above, two pieces of configuration are required:

- configure the AddRoles business rule
- create the transaction that will add roles to a policy

Prototype Examples

The Prototype Company in the Rules Palette contains three sample configurations for AddRoles.

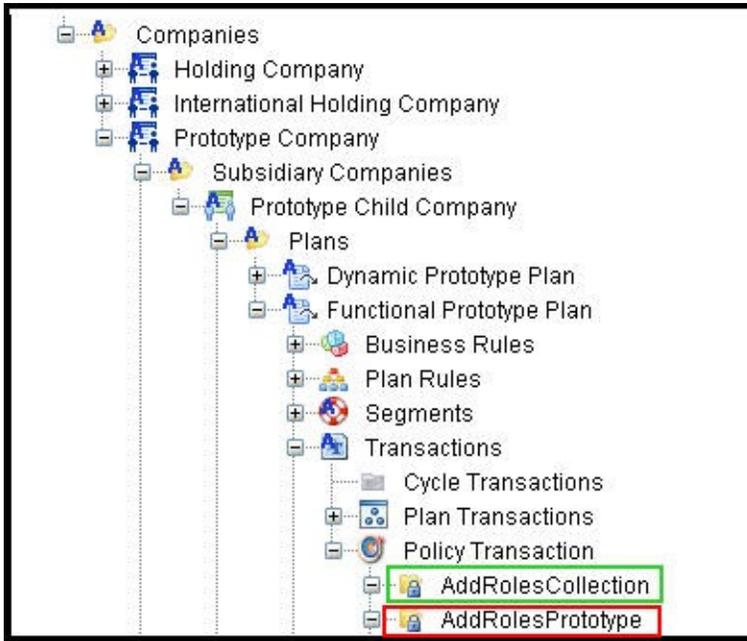
- AddRoles business rule: When attached to a policy level transaction, this business rule will add an existing client to a policy. Navigate through the following folders in the Global Rules Explorer to locate the configuration sample: **Business Rules | Attached | Add Roles | Transaction Overrides**. There are two overrides, one for each of the transactions (AddRolesPrototype and AddRolesCollection)



AddRoles Rule Overrides

- AddRolesPrototype transaction: This transaction adds a single existing client to a policy and assigns the client a role. Navigate through the following folders in the Main Explorer to locate the configuration sample: **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | AddRolesPrototype**.
- AddRolesCollection transaction: This transaction adds multiple

existing clients to a policy and assigns them the same role. Navigate through the following folders in the Main Explorer to locate the configuration sample: **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | AddRolesCollection.**



AddRoles Transactions in Main Explorer

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > CopyToRoleFields Rule Enhancements

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

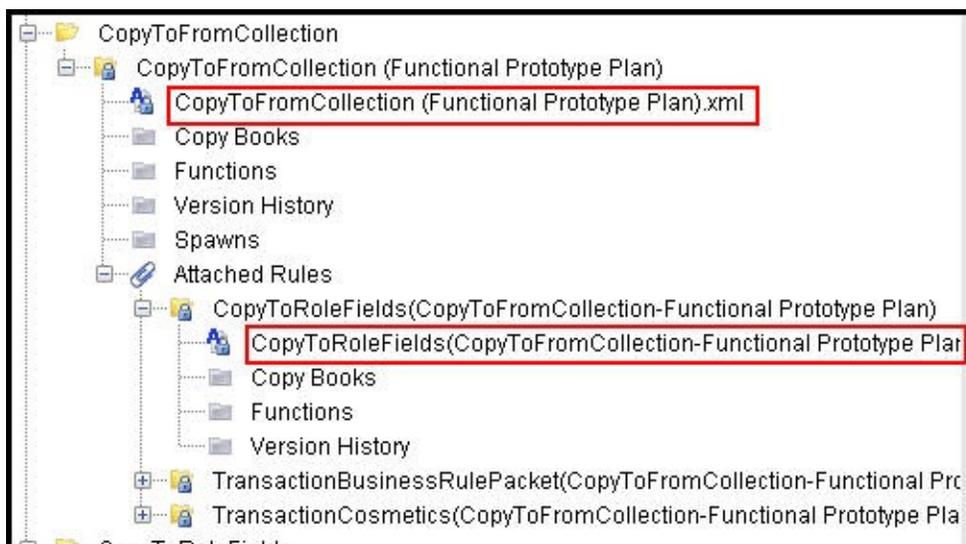
CopyToRoleFields Rule Enhancements

This prototype example was created to demonstrate the update of a field on multiple roles.

Navigate to **Global Explorer | Transactions | CopyToFromCollection (Functional Prototype Plan)**. The CopyToRoleFields rule is in the Attached Rules folder for that transaction.

Configuration Requirements

- **CopyToFromCollection:** This transaction configuration used a math variable named RoleFieldCollection to demonstrate the use of the TYPE=COLLECTION attribute. The value will be copied to the field identified in the CopyToRoleFields attached rule.
- **CopyToRoleFields:** This attached business rule identifies the collection that should be copied and the field it should be copied to, which is CommonRoleField.



CopyToFromCollection Transaction in Global Explorer

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Segment Roles

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Segment Roles Prototype

Segment functionality was enhanced to allow a user to assign and view roles assigned to a segment.

The SegmentRoleScreen business rule is a new rule that allows the user to configure the Segment Role screen and define the dynamic fields that can be displayed and updated on the specific role detail window.

When the [segment is configured](#), the role characteristics are defined, such as the minimum/maximum number of roles allowed and cloning data from existing roles. The role detail can vary by segment and role combination.

Each segment will have distinct roles assigned to it, while the same client(s) may be assigned to multiple segments in the same role or in different roles.

The CopyToRoleFields business rule is used to update fields for multiple roles. Roles assigned to the Segment may also be created by transactions.

Segment Configuration Requirements

- Create the [Segment](#) and configure it to establish and maintain roles as well as supply role views. Segments behave much like the [PolicyScreen business rule](#) in terms of Role creation, editing information provided at the detail level (minimum/maximum number of roles, percentages, disabling role fields), and client type designation. Segment fields, for example, **DisableStatus** and **DisableRoleFields** can be locked down and disabled based on the configuration pointing to the overall policy status. When the entire segment is locked down, disabled, no role deletions, additions, or changes are allowed. When the role fields are disabled **Find Client** and **New Client** are also disabled. Role Views are provided on the Segment to allow for the update of a field on multiple roles. Navigate to **Global Explorer | Segments**. The three segments used to demonstrate this functionality are **RiderWithRoles**, **RiderWithClientAssignmentII**, and **RiderWithoutClientAssignmentII**.
 - **RiderWithRoles**: This segment demonstrates the use of Role Views associated with the display of information on the Role grid in OIPA. The Roles to be included in the view are configurable and roles can be repeated among different views. An optional attribute **ALLROLES** allows for view of all roles, otherwise, the default is displayed. The default view houses the following information for display on the role grid: Client Name, Role, Tax ID, Action, and Percent. In terms of the Percent, **AllowZeroPercent** or **AllowPercent** will be based on the element configured within the role for the particular Segment.
 - **RiderWithClientAssignmentII**: This segment demonstrates the use of Client Assignment configuration. Client Assignment configuration was developed to satisfy the business need to clone existing roles from existing segments or to exclude specific clients already associated with a role on an existing segment. **ClientAssignment**, when checked, will limit the search result set during a 'Find Client' request to those clients defined by the role's ClientAssignment. ClientAssignment specifies information about cloning role detail and excluding

assigned clients. **CLONEROLEDETAIL** stipulates whether the common role detail is cloned/copied to the new role.

EXCLUDEASSIGNED generates a list of available clients not already assigned in previous acts of role designation to the same role code on the same segment. When assigned **Yes**, clients already assigned are removed from the list. The **Source Segment** provides the information for Client Assignment in terms of the Segment to use for copying of role detail and which roles to include/exclude. When cloning Role Detail, there must be a match in the field name between the source and destination segment.

- **RiderWithoutClientAssignmentII**: This segment demonstrates no Segment Role Views or the presentation of default views when configuration depicting another view is absent.
- The **SegmentRoleScreen** business rule is used to define the role detail at the field level. There is specific syntax within the SegmentRoleScreen business rule that allows dynamic role fields to be shared across multiple roles for segments.
- There are eight transactions configured to demonstrate the segment role functionality. The purpose of each transaction is provided below.
 - **CopyToWithALLROLES**: This transaction demonstrates the use of the attribute ALLROLES associated with the element SegmentRoles used in the CopyToRoleFields rule configuration. It demonstrates the use of Test elements in the CopyToRoleFields rule. Configuration stipulates that the RoleRelationship field will be updated only for those segments whose type code is not 72 (72 is Rider Without Client Assignment). Navigate to **Global Explorer | Transactions | CopyToWithALLROLES (Functional Prototype Plan)**.
 - **CopyToWithoutALLROLES**: This transaction demonstrates the use of a Role Code in the CopyToRoleFields business rule to determine which role is updated. The transaction allows the user to select a segment and limits the list of available roles in the drop down based on the roles added to the segment selected. **CopyToRoleFields** in terms of Segments, provides the

capability to update field level information to all of the Segment's roles (with the exception of the CSR) by using the **ALLROLE** attribute. An Individual/single role may be updated specifically, by using the **ROLECODE** attribute. And **From** and **To** and (**FromCollection**) elements provide a method for passing the applicable data to the role specified. Conditions, Statements, and Tests serve as qualifiers in the configuration to determine if the role should be updated or not. **ROLEPERCENT** and **ROLEAMOUNT** can be updated. Activity Field level values can be updated. Undo/Redo or Reversal processing reverts the values back prior to when activity processing was executed to copy to the role field(s). Navigate to **Global Explorer | Transactions | CopyToWithoutALLROLES (Functional Prototype Plan)**.

- **CopyToFromCollection**: This transaction demonstrates the use of the CopyToRoleFields rule using Collection configuration in the transaction. Navigate to **Global Explorer | Transactions | CopyToFromCollection (Functional Prototype Plan)**.
- **SegmentRolesRolesExist**: This transaction demonstrates the use of the RolesExist business rule with Segment roles. Configuration within the rule demonstrates the use of Validations and InterRoleRules. The **RolesExist** business rule allows for the validation of Roles within the segments including **InterRoleRules** and **AgeValidation**. For example, **InterRoleRules** will validate that two roles are served by two separate and distinct clients as in the case where the insured and the beneficiary cannot be the same client. **AgeValidation** will ensure the client's age is appropriate for the designation to the role. For example, the client who occupies the Insured's role, must meet the age range requirement set forth under the plan guidelines. Navigate to **Global Explorer | Transactions | SegmentRolesRolesExist (Functional Prototype Plan)**.
- **SegmentRolesAddRoles**: This transaction demonstrates the use of the AddRoles business rule with Segment roles. The **AddRoles** business rule allows the user to select the client designation, the role designation and the segment to update

using the configuration elements for this business rule. The **AddRoles** business rule allows for the creation of new roles on a particular segment (via the SegmentGUID attribute) when the attached rule is executed on the transaction processed.

Navigate to **Global Explorer | Transactions | SegmentRolesAddRoles (Functional Prototype Plan)**.

- **SegmentRolesCreatePolicy**: This transaction demonstrates the use of the CreatePolicy business rule with Segment roles. The **CreatePolicy** business rule allows for creation of roles at the policy level and segment level. The Roles from the existing source segment as defined by the **SEGMENTNAME** attribute can be copied and created as the new segment on the new policy. If the attribute(**COPYSOURCE**) value is set to No, then the role's section will be generated without the use of copying from the source. From and To elements will provide a vehicle (if copying) for passing the information from the source segment roles to the new segment's roles. Navigate to **Global Explorer | Transactions | SegmentRolesCreatePolicy (Functional Prototype Plan)**.
- **SegmentRolesCOPYALLROLES**: This transaction demonstrates the use of CreatePolicy business rule using the attribute of COPYALLROLES. COPYALLROLES is an indicator (Yes|No) that determines whether or not to copy all of the segment roles from the source policy segment to the new policy's segment. Navigate to **Global Explorer | Transactions | SegmentRolesCOPYALLROLES (Functional Prototype Plan)**.
- **SegmentRolesCreateSegments**: This transaction demonstrates the use of the CreateSegments business rule with Segment roles. The **CreateSegments** business rule allows for the creation of multiple Roles and Role Detail fields on a specified segment. The RoleCode, SegmentName, and StatusCode all apply during the configuration of the business rule. In addition, the From and To sections allow for passing along field level information to the newly created segment. Navigate to **Global Explorer | Transactions |**

SegmentRolesCreateSegments (Functional Prototype Plan).

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Update Multiple Policy Roles with Transaction

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Update Multiple Policy Roles with Transaction

A single transaction can be used to update multiple policy roles. The CopyToRoleFields business rule allows the configurator to limit the roles available for update by role status, to update (a) common field(s) for all roles aligned with a given policy or to specify via RoleGuid or RoleCode those roles subject to update. The role fields being updated are limited to one policy; specifically, the policy associated with the processed activity.

Configuration Requirements

- The **CopyToRoleFields** business rule exists as a transaction level override in the Functional Prototype Plan. Navigate to **Global Explorer | Business Rules | Attached | CopyToRoleFields | Transaction Overrides | CopyToRoleFields (CopyToRoleFields-FunctionalPrototypePlan)**.
- The CopyToRoleFields transaction is used to demonstrate the new functionality. Navigate to **Global Explorer | Transactions | CopytoRoleFields (FunctionalPrototypePlan)**.
 - The DepartmentName field with the display name **Mail sent by** and delivery type of "Priority Mail" is updated to all active Insured Roles on the policy. This is evident by the ROLECODE attribute and the <RoleStatus> configuration. It is only copied to the Insured Role according to <PolicyRoles ALLROLES=No>.
 - The Math Variable DeliveryAgentMV is used to test the ALLROLES=YES attribute and functionality. The math variable value UPS will be copied to all active roles except CSRs.
 - The MathVariable PaymentMethodMap is used to test FromCollection. The values 01 and 02 will be displayed on Insured and Primary Beneficiary TestCollection Field on Role Screen.

If the activity is reversed in OIPA, then all the copied values will be removed. If the roles are not in Active status, then none of the fields will be updated except TestCollection.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > RolesExist Enhancements

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

RolesExist Enhancements

The RolesExist business rule was enhanced to verify that certain specified fields on policy or segment roles are empty at the time an activity is processed in OIPA. The **FieldsDoNotExist** element is the new element that defines the field that cannot hold a value.

Scenario

The RolesExist business rule is attached to a transaction to specify that an Owner Beneficiary role on a policy must not have a Date of Death. The CSR adds an Owner Beneficiary with a Date of Death to the policy. The CSR then tries to process the activity. An error icon appears , which tells the CSR that the Owner Beneficiary role must not have a Date of Death. The activity will not process.

Prototype Example

The **FieldsDoNotExist** prototype was created to show how the RolesExist business rule prevents an activity from processing if specified policy or segment role fields contain a value on the policy. Navigate to **Global Explorer | Transactions | FieldsDoNotExistPrototype2 (Functional Prototype Plan)**. The Attached Rules folder contains the **RolesExist** business rule. Double click the rule to open it in the Configuration Area and display the XML.

```
<RolesExist>
  <Role>
    <RoleCode>30</RoleCode>
    <FieldsDoNotExist>
      <Field GROUP="Client" DISPLAY="Date Of Death">DateOfDeath</Field>
    </FieldsDoNotExist>
  </Role>
  <Segments>
    <Segment SEGMENTNAME="RiderWithRoles">
      <Role>
        <RoleCode>01</RoleCode>
        <FieldsDoNotExist>
          <Field GROUP="Role" DISPLAY="Common Role Field">CommonRoleField</Field>
        </FieldsDoNotExist>
      </Role>
    </Segment>
  </Segments>
</RolesExist>
```

This role cannot have a value in this field.

Roles Exist XML Configuration

Maximum Results 10					
	Activity	Activity Date	Status	Detail	Action
	DisbursementCurrencySelect	1/29/2031	Pending		
	Fields Do Not Exist	1/29/20		<ul style="list-style-type: none">Owner Beneficiary must not have Date Of Death.	

OIPA Error Message for FieldsDoNotExist Prototype

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > RolePercent Enhancements Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

RolePercent Enhancements Prototype

OIPA now allows the role percentages for a role belonging to a policy or segment to exceed 100%. Role fixed fields can be retrieved in ScreenMath using TYPE=EXPRESSION and database column name both in Policy and Segment levels

Prototype Explanation

The following entities were configured in the Functional Prototype Plan, within the Prototype Company, to demonstrate this new functionality.

Business Rules

- **PolicyScreen:** This business rule is configured to allow the role percentages for a certain role to exceed 100%. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Plan Rules | PolicyScreen** and click on the **XML Source** pane. The key configuration for this business rule is explained below.
 - A role has been configured with a <RoleCode> of 41, which corresponds to a role called NoPercentLimit.
 - This same role's <RolePercent> element has a value of "*", indicating that the role percentages are allowed to exceed 100%.
 - The Role screen fixed fields and Role percent can be retrieved in ScreenMath using TYPE=EXPRESSION and database column name like the same syntax as other field mathvariables (ActivityField, Field, ObjectField, PlanField, PolicyField and SegmentField) Example (<MathVariable VARIABLENAME="PrimaryBeneficiaryRolePercent" TYPE="EXPRESSION" DATATYPE="DECIMAL">RolePercent</MathVariable>)

Segments

- **RiderWithRoles:** This segment is configured to allow the role percentages for a certain role to exceed 100%. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Segments | RiderWithRoles** and click on the **XML Source** pane. The key configuration for this segment is explained below.
 - A role has been configured with a <RoleCode> of 41, which corresponds to a role called NoPercentLimit.

- This same role's <RolePercent> element has a value of "*", indicating that the role percentages are allowed to exceed 100%.
- The Role screen fixed fields and Role percent can be retrieved in ScreenMath using TYPE=EXPRESSION and database column name like the same syntax as other field mathvariables (ActivityField, Field, ObjectField, PlanField, PolicyField and SegmentField) Example
(<MathVariable
VARIABLENAME="PrimaryBeneficiaryRolePercent"
TYPE="EXPRESSION"
DATATYPE="DECIMAL">RolePercent</MathVariable>)

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password.
2. Open a policy or create a shell policy belonging to the Functional Prototype Plan.
3. In the menu on the left side of the screen, select **Roles**.
4. Click on the **Find Client** or **New Client** tab.
5. Add a client to the NoPercentLimit role.
6. Set the newly added client's Role Percent to 100%.
7. Add a second client to the NoPercentLimit role.
8. Set this client's Role Percent to any value greater than zero.
9. Click **Save**. The Roles should be saved successfully.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > RoleScreen Actions and Events

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

RoleScreen Actions and Events

Actions and Events can be configured to validate information entered on the Role screen in OIPA. This configuration introduces the ability to validate the information entered when adding a new role against other policy and role information, including the ability to generate error or warning messages when validation is not successful. Validations can also be performed upon deleting and updating roles, with error or warning messages displayed when a role that is not allowed to be updated/deleted is updated/deleted.

Configuration Required

The following configuration must be in place to support redemption fees:

- The **RoleScreen** rule must contain Action/Events configuration for Add, Save, and Delete buttons, which will trigger the validations. The rule must also be configured to display available role in a drop-down field on the Role screen.

Prototype Explanation

The following business rules and transactions were configured in the Dynamic Prototype Plan to demonstrate this new functionality.

Business Rules

- The RoleScreen business rule defines the display and behavior of the Role screen in OIPA. This rule must contain the appropriate Action/Events configuration, as well as configuration to display available roles in a drop-down field. To view the prototype configuration, navigate in the Global Rules Explorer to **Business Rules | Plan Rules | RoleScreen | Plan Overrides | RoleScreen (Dynamic Prototype Plan)** and view the XML Source pane.
 - <**DisplayRoleFields**> defines whether the roles available when adding or finding a client on the Role screen display in a drop-down field or as a series of checkboxes. This element must have a value of Yes, indicating that the roles should display in a drop-down field, in order for the validations to work.
 - In the <**Events**> section, the buttons to display on the Role screen are defined. This section also links the buttons to the corresponding ScreenMath and Actions, which will define the exact validations that are to occur.
 - In the <**ScreenMath**> section, math variables that access policy, client and role information are defined. These math variables drive the validation of the information on the Role screen, and define the validation conditions.
 - In the <**Actions**> section, the validation error/warning messages are defined, which will display on the Role screen when the conditions defined in the <ScreenMath> section are met.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password.
2. Open or create a Custodial IRA policy.
3. Click the **Roles** link on the menu on the left side of the screen. This will open the Role screen.
4. Click on the **Find Client** tab.
5. From the **Roles** drop-down field (in the **Find Client** section), select the **Owner** role type.
6. From the **Type** drop-down field (in the **Client Search Criteria** section), select **Individual**.
7. Enter any additional search criteria and click **Find**.
8. Select a client from the **Client Search Results** grid.
9. Click the **Add** button at the bottom of the screen.
10. An error message should display at the top of the screen that reads "The owner must be a Corporation for a Custodial IRA".

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Role Shading and RoleScreen Access to Plan, Policy, Client and Address Fields Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Role Shading and RoleScreen Access to Plan, Policy, Client and Address Fields Prototype

RoleScreen is an existing business rule, with all the existing functionality and the same is explained in earlier prototypes. The current Role Shading and RoleScreen Access to Plan, Policy, Client and Address Fields Prototype explains the features in OIPA that enable role shading to be performed basis role status and detailed action/events using PLANFIELD, POLICYFIELD, Client and Address details for the role player in the action/events configuration.

Scenario 1

For a specific plan, the Max Issue Age for the plan is 70. In a given instance, the insured DOB is entered as 01/01/1940, a validation is carried out using the client details to find if the Client is within the age of 70 at IssueDate and hence Error is thrown "Max Issue Age for insured is 70 years" and does not allows the Role to get added.

Scenario 2

For an Insured role, the First Name is a required field. In a given instance, the insured Last Name is entered as "Harry" but no First Name is entered. By accessing the First and Last Name, the same is validated and an error message "First Name is required for the last name \$\$\$Client:LastName\$\$\$" is displayed and the role details are not recorded.

Scenario 3

For an Insured role, the gender value Unisex is allowed only in the US state of Montana. In a given instance, the user selects an insured with gender 'Unisex' residing in a state other than Montana. An error message “For an Unisex Insured residence State needs to be Montana” and the Role is not added.

Scenario 4

For an Insured role, there is more than one role that is in deleted/Inactive status after some role changes are performed in the policy. For the Deleted/Inactive roles, the roles fields should not be available for edit and the role should be shaded differently to ensure the deleted/inactive roles are visible differently.

Prototype Configuration

- The existing transaction 'RoleScreen' BR in the Functional Prototype Plan is used for demonstrating the functionality associated with this feature. The following features are configured to demonstrate the same functionality:
 - For Scenario 1, two fields, MaxIssueAge from Plan Field with TYPE="PLANFIELD" and PolicyEffectiveDate (treated as Issue Date) from Policy Field, TYPE="POLICYFIELD" are configured in Role Screen Math. At the same time, Client:DateOfBirth is used as an EXPRESSION for fetching client's date of birth, which acts as a parameter to ANBAgeOf("ClientDOB", "IssueDateMV") to calculate the Age on Issue Date of a client. Alternatively, Plan Date is also used as Issue date when user does not enter Policy Effective date which is an user enterable field, whereas Plan Date is always available, with a system defaulted date. Basis the Issue Age, appropriate error message is displayed, if necessary.
 - For Scenario 2, Client context variable, Client: FirstName is accessed in the validation to ensure the first name is available. If it is not entered, an error message is shown and the value for context variable Client: LastName is used in substitution.
 - For Scenario 3, Client's gender is fetched configuring Client: Sex and Client state details by using the Address: StateCode in the actions and depending on the values for Sex and residence state code, an error message is displayed if Unisex is used with any state other than Montana.
 - For Scenario 4, the role shading feature is enabled in the Prototype Company at the company level by using the CompanyCosmetics BR. This ensures the roles are shaded basis the role status. Please see section on [RoleShading](#) feature for details.

View Prototype in OIPA

1. Log in OIPA using the Prototype Company user ID and password.
2. Click **Policy | New** from the Main menu.
3. Add a shell policy for the Functional Prototype Plan to test the prototype.
4. Navigate to the Role Screen and enter input details as explained in each of the scenario to view the prototype functionality.
5. Make one or more Roles Deleted/Inactive by using appropriate activities to see the Role Shading functionality.
6. Make changes to the various fields and check action/events as explained above to view the functionality.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Quote and Verification Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Quote and Verification Prototype

Activities can be configured to allow a user to see a virtual representation of processing before the activity is actually processed. The Quote functionality allows a user to virtually process an activity and then view all the activity results. Additional activities can be added from the Result screen in the Quote mode as well. The QuoteScreen business rule must be attached to a transaction if quote functionality is needed.

The Verification functionality allows a user to check the success of activity processing before the activity is actually processed. Allocations can be viewed, both original and final, and business errors are listed. The VerificationScreen business rule must be attached to a transaction if this functionality is needed.

In order to demonstrate these two new features, seven transactions were added to the Prototype Company: two Client level transactions and five Functional Prototype Plan (Prototype Child Company) level transactions. Each demonstrates some aspect of the screens' capabilities.

Scenario 1

A withdrawal activity needs to process against a policy. The CSR hits the Verify button on the activity after all activity details are entered. The Verification screen displays and confirms that the withdrawal amount is acceptable for the policy. Final allocations are displayed.

Scenario 2

A premium activity needs to be applied to a policy. The CSR enters the activity details then hits the Quote button to see the results of processing the activity before the activity is actually processed.

Prototype Examples

- Two client level transactions were created to demonstrate that quote and verify activities can be processed at the client level.
 - **QuoteDisbursement:** demonstrates that a client level activity can support quote functionality. Disbursement information is shown in the Disbursement link in the Quote window. Navigate to **Global Explorer | Transactions | QuoteDisbursement (ClientPlan)** to view the transaction XML.
 - **VerifyDisbursement:** demonstrates that a client level activity can support verification functionality. Navigate to **Global Explorer | Transactions | VerifyDisbursement (ClientPlan)** to view the transaction XML.
- The **QuotePremium** transaction has an attached QuoteScreen business rule, which is configured to demonstrate the fulfillment of the following items in quote mode on the Activity Result screen: Entry Fields, Allocations, Math, Spawn, Accounting and Suspense. This activity spawns the QuoteRequirement transaction. Navigate to **Global Explorer | Transactions | QuotePremium (Functional Prototype Plan)** to view the XML.
- The **QuoteWithdrawal** transaction has an attached QuoteScreen business rule, which is configured to demonstrate the fulfillment of the following items in quote mode on the Activity Result screen: Entry Fields, Allocations, Math, Accounting, Valuation, Show on Error attribute, Add New Activity and Activity Summary. Navigate to **Global Explorer | Transactions | QuoteWithdrawal (Functional Prototype Plan)** to view the XML.
 - To demonstrate the Activity Summary tab, the **ActivitySummary** business rule (configured to display Total Cash and Deficiency Amount MathVariables) is attached to the transaction. Navigate to **Global Explorer | Transactions | QuotePremium (Functional Prototype Plan) | Attached Rules | ActivitySummary (QuoteWithdrawal-Functional Prototype Plan)** to view the XML.
 - To demonstrate how to add a new activity from the Quote window (Add New Activity link), two new transactions were added: **SendDeficiencyLetter** and **WithdrawalRequestDenied**. Navigate to

Global Explorer | Transactions | to view the XML for the two transactions.

- The **QuoteRequirement** transaction serves as the activity spawned by the QuotePremium transaction. Navigate to **Global Explorer | Transactions | QuoteRequirement (Functional Prototype Plan)** to view the XML.
- The **VerifyPremium** transaction has an attached VerificationScreen business rule, which is configured to demonstrate all available Field data types, including Message with dynamic inserts, Allocations with SHOWORIGINAL attribute set to “Yes”, and Errors. Navigate to **Global Explorer | Transactions | VerifyPremium (Functional Prototype Plan)** | to view the XML.
- The **VerifyWithdrawal** transaction has an attached VerificationScreen business rule, which is configured to demonstrate Fields with Message data type and dynamic inserts, Allocations set to “No”, and Errors. This transaction also demonstrates that the attached ConfirmationScreen business rule functions in concert with the VerificationScreen business rule. Navigate to **Global Explorer | Transactions | QuotePremium (Functional Prototype Plan)** to view the XML.

View Client Prototypes in OIPA

1. Log into the Prototype company in OIPA.
2. Click **Client** and open a client record. The selected client record must have an address for the prototype activities to successfully process.
3. Click **Add Activity** from the Secondary menu.
4. Select the **QuoteDisbursement** or **VerifyDisbursement** activity from the drop down list.
5. Add some test disbursement information.
6. Click **Quote** or **Verify**. The Quote window or Verification window will open with links to view all pre-processed information for the activity.



To process the other five policy level transactions, open a policy instead of a client and add the desired activity. Click Quote or Verify respectively to view the functionality.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Scheduled Computation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Scheduled Computation

Scheduled computation is used to calculate values on policies that do not have a variable component. The ScheduledValuation business rule was enhanced to include a <Computation> element that when present toggles the rule's action from valuation to computation.

A SQL query within the business rule identifies the policies or segments selected for the various computation calculations such as cash value, surrender charge, and anything else required using standard math syntax. The output values that are calculated are stored in the database in order to be accessed for reporting or picked up by downstream systems.

Scenario

Use a plan level transaction to perform scheduled computation on all policies within the SQL query.



Cycle is involved in scheduled computation. A cycle agent has to be deployed; either in a web container like Weblogic or Websphere, or as a standalone application, and must be running in order for scheduled computation to work correctly. Refer to the Cycle document in the [Documentation Library on OTN](#) for additional information on cycle.

Configuration Requirements

The following components must be configured in order to perform scheduled computation:

- a plan-financial transaction with the <Asynchronous> element must be configured.
- the ScheduledValuation business rule with the <Computation> element must be configured. This should be attached to the transaction.
- the Computation business rule must be overridden at the transaction level.
- a Cycle agent must be deployed and running.

Prototype Samples

There are several business rules used to demonstrate this configuration:

- **ScheduledValuation** business rule demonstrating policy computation: this rule must be overridden at the plan level. The <Query> element contains the SQL statement that identifies the policies involved in the calculation. The <Computation> element identifies that the computation is for policy. Navigate in the Global Explorer to **Business Rules | Attached | Scheduled Valuation| Transaction Overrides | ScheduledValuation (ScheduledComputationPolicy-Functional Prototype Plan .**
- **ScheduledValuation** business rule demonstrating segment computation: this rule must be overridden at the plan level. The <Query> element contains the SQL statement that identifies the segments involved in the calculation. The <Computation> element identifies that the computation is for segment. Navigate in the Global Explorer to **Business Rules | Attached | Scheduled Valuation| Transaction Overrides | ScheduledValuation (ScheduledComputationSegment-Functional Prototype Plan .**
- **PolicyComputation**: this is a system rule that must be overridden at the transaction level. The <Input> element defines the math variables that are used for the calculations. The <Output> element defines the mappings assigned to the calculations. Navigate in the Global Explorer to **Business rules | System Rules | Computation | Plan Overrides | PolicyComputation (Functional Prototype Plan-Prototype Child Company.**
- **SegmentComputation**: this is a system rule that must be overridden at the transaction level. The <Input> element defines the math variables that are used for the calculations. The <Output> element defines the mappings assigned to the calculations. Navigate in the Global Explorer to **Business rules | System Rules | SegmentComputation.**

There are two plan level transactions configured to demonstrate scheduled valuation:

- **ScheduledComputationPolicy**: this transaction must include the `<AsynchronousActivity>` element. Navigate in the Global Explorer to **Transactions | ScheduledComputationPolicy (Functional Prototype Plan)** to view the configuration.
- **ScheduledComputationSegment**: this transaction must include the `<AsynchronousActivity>` element. Navigate in the Global Explorer to **Transactions | ScheduledComputationSegment (Functional Prototype Plan)** to view the configuration.

View Prototypes in OIPA

1. Log in OIPA using the Prototype Company user ID and password.
2. Click **Plan | Plan Activity** from the Main menu.
3. Click **Add Activity** from the Secondary menu.
4. Select the ScheduledComputation activity from the Activity drop down box.
5. Type an effective date in the Effective Date field.
6. Click the lightning bolt icon to process the activity. (Processing would typically be done through cycle.)
7. Give the application about five minutes, then refresh the screen. The processing status will change from Processing (Wait) to Active.
8. View the computation results by referencing the database tables. All information exists on the AsComputationRequest, AsScheduledComputation and AsScheduledComputationField tables. The last table has all of the output data from the computation for each policy or segment.

You are here: Configuration > Valuation > Scheduled Valuation > Scheduled Valuation Configuration

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Scheduled Valuation Configuration

The purpose of Scheduled Valuation is to perform a policy valuation at a specific time for a group of policies and store that valuation for subsequent use. The time intervals for running the valuation are set by the user. The frequency at which policies are subject to valuation is also set by the user. Typical time intervals are quarterly, semi or annually. Multiple currencies are supported in Scheduled Valuation and policy information such as value and deposit information can be stored in the database.

A plan level transaction is configured to perform scheduled valuation. There are two business rules that should be attached to the transaction: **ScheduledValuation** and **CopyToScheduledValuationFields**.

Scenario

Use a plan level transaction to perform scheduled valuation on all policies within the plan.



Cycle is involved in scheduled valuation. A cycle agent has to be deployed; either in a web container like Weblogic or Websphere, or as a standalone application, and must be running in order for scheduled valuation to work correctly. Refer to the Cycle document in the [Documentation Library on OTN](#) for additional information on cycle.

Configuration Requirements

The following components must be configured in order to perform scheduled valuation:

- a plan must be created and the PlanScreen business rule must be configured and segments must be defined.
- funds must be [created](#) and associated with the plan.
- the [InterestRateCalculation](#) business rule must be overridden for each fixed fund.
- the PolicyScreen business rule must be overridden at the plan level.
- a plan-financial transaction with the <Asynchronous> element must be configured. The fields and math should contain data that all scheduled valuation records will receive via the configuration in the CopyToScheduledValuationFields business rule.
- the ScheduledValuation business rule with the PlanGUID for the plan where scheduled valuation should occur. This should be attached to the transaction.
- the CopyToScheduledValuationFields business rule, which should be attached to the transaction.
- the PolicyValues business rule overridden at the plan level. This is where individual values for each policy executing through ScheduledValuation are calculated. These may be accessed by CopyToSecheduledValuationFields.
- Cycle agent must be deployed and running.

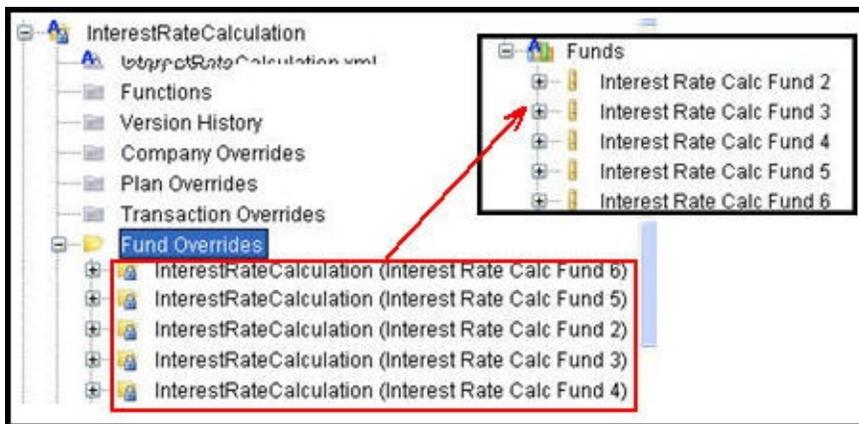
Optional Requirements

If Point-in-Time valuation is needed, then the following additional business rules and tables need to be configured:

- the PointInTimeValuation business rule should be overridden at the plan level
- the WriteValuationElements business rule should be overridden at the plan level
- AsPlan table column named PointInTimeValuation must be set to a value of **Y**(or **T** if the plan is transitioning from Traditional valuation)
- DepositLevelTracking table column named AsFund must be set to a value of **N**

Prototype Samples

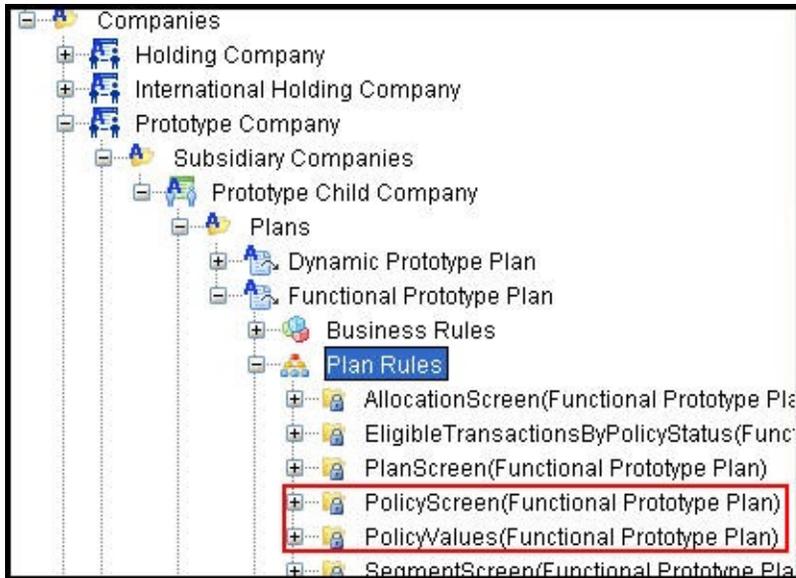
Funds need to be created, along with an InterestRateCalculation business rule override for each fund. The Funds are displayed in the Main Explorer in **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Funds**. The rule overrides are displayed in the Global Rules Explorer in **Business Rules | System | InterestRateCalculation | Fund Overrides**.



Funds from Main Explorer and InterestRateCalculation Overrides from Global Rules Explorer

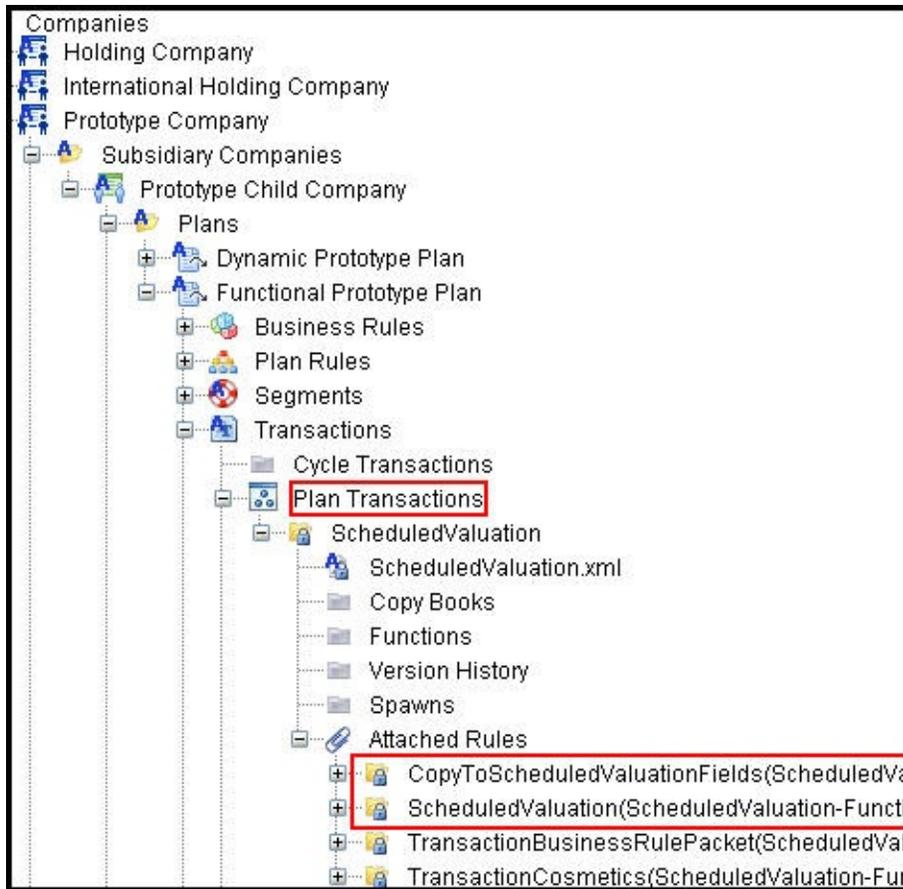
There are several business rules used to demonstrate this configuration:

- PolicyScreen business rule: this rule must be overridden at the plan level. Two new fields are needed: one for InterestBonusQual and one for Effective date. Navigate in the Main Explorer to **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Plan Rules**.
- PolicyValues business rule: this rule should be overridden at the plan level. Navigate in the Main Explorer to **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Plan Rules**.



PolicyValues and PolicyScreen Plan Level Overrides in Main Explorer

- ScheduledValuation business rule: this rule should be overridden at the transaction level and attached to the transaction. Navigate in the Main Explorer to **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Plan Transactions | ScheduledValuation | Attached Rules**.
- CopyToScheduledValuationFields business rule: this rule should be overridden at the transaction level and attached to the transaction. Navigate in the Main Explorer to **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Plan Transactions | ScheduledValuation | Attached Rules**.

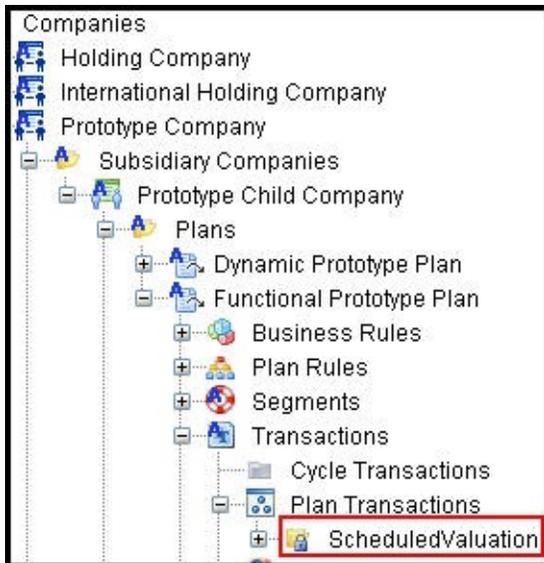


Scheduled Valuation Transaction Attached Rules in Main Explorer

- InterestRateCalculation rule: this rule was discussed in the fund section and should be overridden at the fund level for each fund.

There is one plan level transaction configured to demonstrate scheduled valuation:

- ScheduledValuation: this transaction must include the <AsynchronousActivity> element. Navigate in the Main Explorer to **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transaction | Plan Transactions** to view the configuration.



ScheduledValuation Transaction in Main Explorer

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Shadow Policy

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Shadow Policy

Policies can be removed from the system according to two scenarios. First, if the policy has no activities, segments, roles, or pending or recoverable disbursements, then a Shadow Policy option on the Policy screen Secondary menu allows a user to delete the policy. It will be completely removed from OIPA, but it will still exist in the database. Second, a StatusChange activity can be added to a policy and processed to move the policy to Shadow status. The policy will be completely removed from OIPA, but it will still exist in the database.

Scenarios

A CSR creates a new policy but before any activities, segments, roles or disbursements are added, he receives a call saying that the policy is void. The CSR clicks the Shadow Policy button on the Secondary menu of the Policy screen. A window displays and the CSR clicks **OK** to delete the policy. All validations pass and the policy is deleted. It is still available in the database, but can no longer be viewed in OIPA.

A CSR creates a new policy and adds segments, roles, and activities. The policy is advanced to active status. The CSR is then told to delete the policy. He adds the StatusChange activity to the policy and processes it. The policy is changed to Shadow status and is no longer visible in OIPA. It is still available in the database.

Configuration Requirements

There are three business rules that must be modified to support shadow policy functionality.

1. CompanyCosmetics
2. EligibleTransactionsByPolicyStatus
3. PolicyScreen

A transaction must be created to allow a user to change the status of an activity in OIPA. The StatusChange activity is used to accomplish this.

Business Rules

- **CompanyCosmetics:** This rule is overridden at the company level. A new element ShadowPolicyStatus identifies the status code or codes that represent a shadowed policy. The status codes are defined in the [Code Names](#) editor in the Administration folder of the Admin Explorer.

Navigate to **Global Explorer | Business Rules | System | Company Cosmetics | Company Overrides | CompanyCosmetics (Functional Prototype Plan).**

The screenshot displays the Admin Explorer interface. On the left, the 'AsCode' section shows the 'Code Name: AsCodeStatus' and a table of 'Codes: AsCodeStatus definitions'. The table has columns for 'Code Value', 'Short Desc...', and 'Long Desc...'. The rows are numbered 01 to 14. Row 04 is highlighted in red and contains 'Cancelled' for both short and long descriptions. Row 12 is also highlighted in red and contains 'Shadow' for both. On the right, the XML code for the 'CompanyCosmetics' rule is shown. The code includes a 'ShadowPolicyStatus' element with two 'ShadowStatusCode' children: '12' and '04'. Both '12' and '04' are highlighted in red. A callout box with a red arrow points to these two values, containing the text 'policies in these statuses are shadowed'.

Code Value	Short Desc...	Long Desc...
01	Active	Active
02	Pending	Pending
03	Inactive	Inactive
04	Cancelled	Cancelled
05	Grace	Grace
06	Suspended	Suspended
07	Surrendered	Surrendered
08	Pending	Policy is Pending
09	Pending	Pending/Ready
10	Death	Death
11	Death - Annuity	Death - Annuity
12	Shadow	Shadow
13	NUV Pending	Active - NUVs n
14	Gain/Loss Pend	Active - NUVs n

```
1 <CompanyCosmetics>
2   <CompanyCosmetics>
3     <Tile>background6.gif</Tile>
4   </CompanyCosmetics>
5   <ShadowPolicyStatus>
6     <ShadowStatusCode>12</ShadowStatusCode>
7     <ShadowStatusCode>04</ShadowStatusCode>
8   </ShadowPolicyStatus>
9 </CompanyCosmetics>
```

CompanyCosmetics Rule and AsCodeStatus Definitions

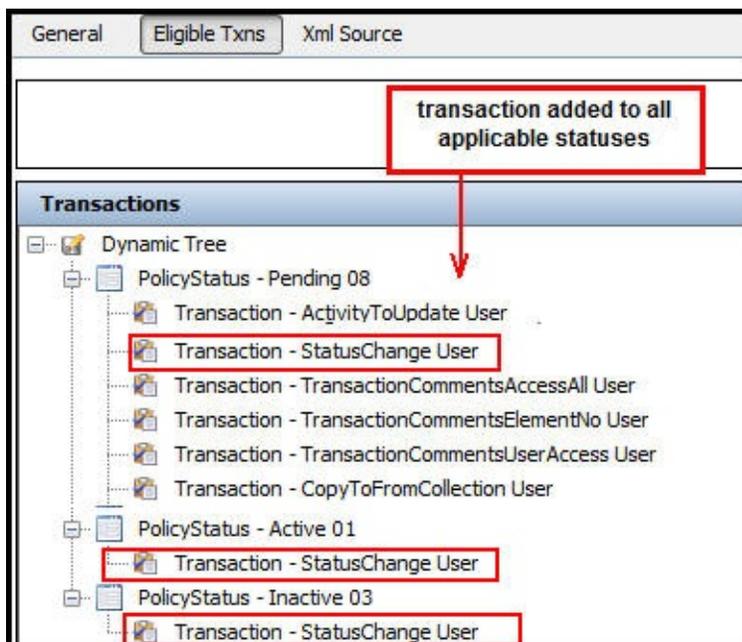
- **PolicyScreen:** This rule is overridden at the plan level. The new section ShadowPolicy defines the functionality of the shadow process. The ShadowStatusCode element identifies the status the policy is moved to after it is deleted. This status must be one of the statuses identified in the CompanyCosmetics rule with the ShadowStatusCode element. The AllowShadowButton element identifies the statuses a policy must be in to display the Shadow Policy option on the Secondary menu. The Validation section identifies the elements that cannot exist on a policy if the Shadow

Policy button is used.

Navigate to **Global Explorer | Business Rules | Plan Rules | PolicyScreen | Plan Overrides | PolicyScreen (FunctionalPrototypePlan)**.

- **EligibleTransactionsByPolicyStatus**: This rule identifies the transactions that are available to a user based on the status of the policy. The StatusChange activity should be added to all policy statuses. This allows the user to add it to a policy regardless of the policy status.

Navigate to **Global Explorer | Business Rules | Plan Rules | EligibleTransactionsByPolicyStatus | Plan Overrides | EligibleTransactionsByPolicyStatus (Functional Prototype Plan)**.



StatusChange Transaction in All Applicable Statuses

Transactions

- **StatusChange:** This transaction provides a list of statuses a user can select from when moving a policy from its current status to a new status. When this activity is processed in OIPA, the policy status changes to the status selected in this activity.

Navigate to **Global Explorer | Transactions | StatusChange**.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > SplitPercentage Assignments Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

SplitPercentage Assignments Prototype

OIPA now supports two new assignment types: SplitPercentageRemoval and SplitPercentageApply. SplitPercentageRemoval removes a specified percentage from each undepleted deposit of each underlying fund associated with a policy, while SplitPercentageApply applies the removed money to a different policy on either the same plan or a different plan.

Prototype Prerequisites

- The policy needs to be in Pending status.
- In the Dynamic Prototype Plan, the Balanced fund has deposit-level tracking enabled.
- The funds and fund codes listed below have been made available to their respective plans.

Dynamic Prototype Plan	Functional Prototype Plan	Fund Code
Dynamic Calc Fund 1	Interest Rate Calc 1	F1
Variable Fund 1	Variable Fund 1	Var1
Variable Fund 2	Variable Fund 2	Var2
Balanced	Interest Rate Calc 2	Bal

- Two new MoneyTypes have been made available in AsCodeMoneyType:
 - 83 – SplitPercentageRemoval
 - 84 – SplitPercentageApply

Prototype Explanation

The following business rules and transactions were configured in the Prototype Company to demonstrate this new functionality.

Transactions

- **PolicySplit** (Dynamic Prototype Plan): When processed, this transaction creates a new policy under the chosen plan complete with the specified policy roles. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Dynamic Prototype Plan | Transactions | PolicySplit**. The key configuration for this transaction is explained below.
 - Five fields are configured to capture necessary information:
 - Owner of the new policy
 - Insured of the new policy
 - Percentage of split
 - Redemption fee (Yes/No)
 - Plan name for the new policy
 - A new policy number is created using the MathVariable NewContractNumber.
 - A new transaction, TransferAssetsOut, is spawned by PolicySplit. The new contract number, as well as the values of the Redemption Fee and Split Percentage fields, are passed as the inputs for TransferAssetsOut on the same day.

Note: PolicySplit does not showcase any of the new functionality implemented by this enhancement. Instead, it serves as a mechanism for setting up different assignment scenarios.

- **TransferAssetsOut** (Dynamic Prototype Plan): This transaction is spawned from PolicySplit. It calculates the percentage of each fund's remaining value that will be removed and, upon successful processing, removes this percentage from the funds. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Dynamic**

Prototype Plan | Transactions | TransferAssetsOut. The key configuration for this transaction is explained below.

- Three fields are passed from PolicySplit, all of which are disabled for editing:
 - SplitPercentage
 - TransferToPolicy (policy number of the new policy created by PolicySplit)
 - RedemptionFee
- The SplitPercentageRemoval assignment type is used to remove the split percentage from each undepleted fund.
- The MathVariable GrossedUpSplitPercentage grosses up the split percentage entered in the PolicySplit transaction to ensure that the proper amount of the fund's remaining balance is removed.
- A new transaction, TransferAssetsIn, is spawned to the new policy created by PolicySplit.
- **TransferAssetsIn (Dynamic Prototype Plan):** This transaction is spawned from TransferAssetsOut. It applies money to the same funds as the removal. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Dynamic Prototype Plan | Transactions | TransferAssetsIn.** The key configuration for this transaction is explained below.
 - This transaction has only one field, TransferAssetsOutsActivityGUID, which is passed in from TransferAssetsOut. This field is disabled for editing.
 - The SplitPercentageApply assignment type is used to apply money to the funds. This assignment uses the Split Percentage Apply MoneyType.
- **TransferAssetsIn (Functional Prototype Plan):** This transaction is spawned from TransferAssetsOut. It applies money to different funds than the removal. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Transactions | TransferAssetsIn.** The key configuration for this transaction is explained below.
 - This transaction has only one field, TransferAssetsOutActivityGUID,

which is passed in from TransferAssetsOuts. This field is disabled for editing.

- The SplitPercentageApply assignment type is used to apply money to the funds. This assignment uses the Split Percentage Apply MoneyType.
- A collection MathVariable called FundList is populated with destination funds using a SQL statement. The SQL statement funds a match for the source funds to the destination funds using a FundField called FundCode.

Business Rules

- **RedemptionAmountFormula** (Dynamic Prototype Plan): This business rule provides the formula for calculating a redemption fee. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Dynamic Prototype Plan | Business Rules | System Rules | RedemptionAmountFormula**. The key configuration for this business rule is explained below.
 - If the plan uses Inception valuation, this rule will use the Original Deposit Date from AsValuation (if present) instead of the valuation record's Effective Date.
 - If the plan uses Point-in-Time valuation with deposit-level tracking enabled, this rule will use the Original Deposit Date (if present) from AsDepositValue instead of the valuation record's Effective Date.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password.
2. Open a policy or create a shell policy belonging to the Dynamic Prototype Plan.
3. Add the PolicySplit activity and enter the required information.
4. Process PolicySplit. Note that the TransferAssetsOut activity is spawned.
5. Process TransferAssetsOut.
6. Navigate to the new policy created by PolicySplit. Note that the TransferAssetsIn activity was spawned.
7. Process TransferAssetsIn.
8. View the allocation information for both the original policy and the policy created by PolicySplit. Note that the designated percentage of each fund was moved from the original policy to the new policy.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > State Approval
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

State Approval Prototype

OIPA supports the use of a filter to restrict policies from being issued in states where the plan isn't approved. Segments can also be filtered so that only segments that are approved in the policy issue state can be added to the policy.

Two steps must be performed to enable state approval filtering for plans and segments. First, the screen business rule must be configured to use state approval. Segment state approval requires configuration to be added to the SegmentScreen rule and plan state approval requires configuration to be added to the PlanScreen rule. Then security must be added to the state approval.

When the state approval filtering is applied to a plan, only those states that have been approved will display in the **Issue State** drop down box on the Policy screen. When state approval filtering is applied to segments, then two categories of segments will display in the Add Segments drop down box on the Segment screen: segments with no state approval filtering assigned and segments with state approval filtering that meet the following filter criteria:

- state approval record exists for the Issue State selected for the policy
- effective date is on or before the plan date. If an expiration date exists for the record, then it must be on or after the plan date.

Scenario

A new policy is created and the CSR wants to issue the policy in the state of New York. The state has an approval record that is on or before the plan date so it is visible in the Issue State drop down box. The CSR selects the state. Roles are added. A RiderWithRoles segment is needed, but the segment is not in the Add Segment drop down box on the Segment screen because the segment isn't approved for New York as of the plan date.

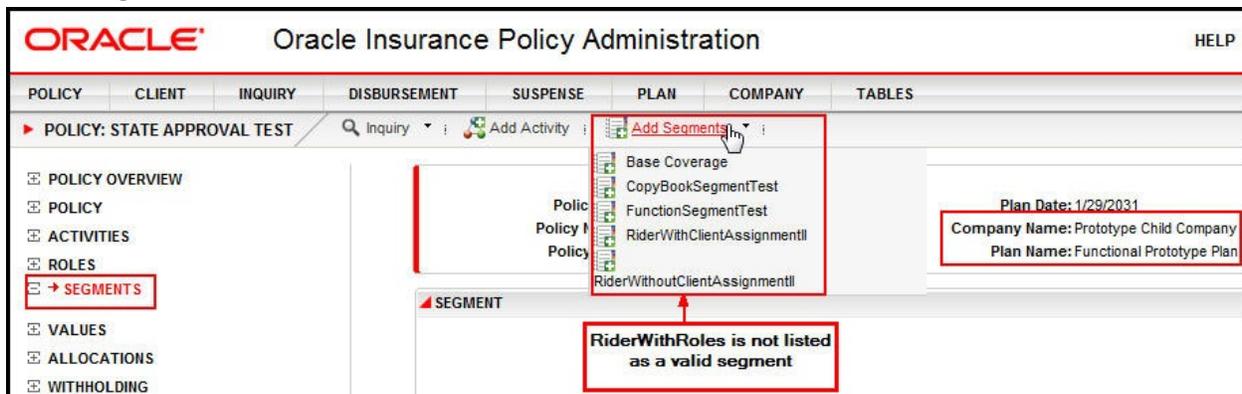
Configuration Requirements

- The Plan screen business rule must have UseStateApproval set to **Yes**. Navigate to **Global Explorer | Business Rules | Plan Rules | Plan Screen | Plan Overrides | PlanScreen (Functional Prototype Plan - Prototype Child Company)**.
- The Segment screen business rule must identify all segments that use state approval filtering. The <SegmentNames> element identifies the section that contains all segments that will use state approval. Only one segment, **RiderWithRoles**, is assigned in this example. Navigate to **Global Explorer | Business Rules | Plan Rules | Segment Screen | Plan Overrides | SegmentScreen (Functional Prototype Plan - Prototype Child Company)**
- Security must be applied to the state approval table so the OIPA users can view the plans and segments that participate in the filtering. This is applied in the Admin Explorer. Navigate to **Admin Explorer | Security | Application Security | Security Groups | Prototype Super | Company Pages | Prototype Company | State Approvals**.
- Plans must be added using the State Approval editor. After the PlanScreen business rule is configured to accept State Approval filtering, the user may add the plan to the State Approval node. Open the plan folder under the editor and add effective dates for all states that approve the plan. An expiration date is optional. Navigate to **Admin Explorer | Administration | State Approvals | Prototype Child Company | Plan State Approvals | State Approval**.
- Segments must be added using the State Approval editor. After the SegmentScreen business rule is configured to apply state approval filtering to various segments, the user may add the segment to the State Approval node. Each segment will have its own folder. Open the segment folder and locate the State Approval editor. Check-out the file and add effective dates. Expiration dates are optional. Navigate to **Admin Explorer | Administration | State Approvals | Prototype Child Company | Plan State Approvals | Segment State Approvals | RiderWithRoles | State Approval**.

View State Approval Filtering in OIPA

In order to see the above configuration in action, follow the steps below. Notice that the Issue State is accepted because New York is one of the approved states. The RiderWithRoles segment will NOT appear in the Add Segment drop down list because it is not approved for the state of New York. All other segments will appear because they do not have state approval filtering applied to them.

1. Navigate to OIPA and login to the Prototype company using a prototype login.
2. Create a new policy on the Functional Prototype Plan and set the Issue State to **New York**. Make sure the policy creation date is on or after the state approval effective date.
3. Add roles.
4. Click the **Segment** link in the Left Navigation menu.
5. Click **Add Segments** on the Secondary menu. Notice the **RiderWithRoles** segment is not available.



OIPA with State Approval Filter Applied to Segment Drop Down Box

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Display Suspense Accounting Details

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

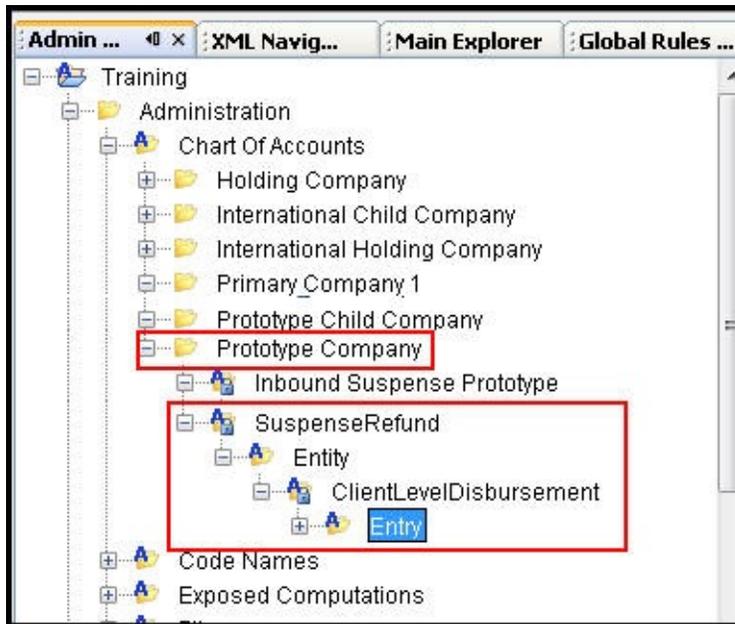
Display Suspense Accounting Details

OIPA needs a way for configuration to indicate when activity accounting should display on the suspense item that is attached to an activity. The [AsChartOfAccountsEntry](#) now holds a LinkSuspenseFlag column (value 1 or 0) for transaction accounting that will link the attached suspense record in the activity to the accounting record created from the transaction. This accounting detail is visible from the suspense screen accounting in OIPA and the Accounting tab on the activity's results.

The prototype plan ClientLevelDisbursement has been setup to use a single suspense element in the transaction as well as a COAEntry for the DisbursementAmount math variable. Processing the client level activity will create an accounting record. This accounting record is visible on the suspense screen for the linked suspense record.

Configuration Requirements

- COA Entity created for Client Level Disbursement. When the client level activity is processed, an accounting record will be generated and viewable through the attached suspense Suspense screen.



COA ClientLevelDisbursement Entity in COA in Admin Explorer

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Suspense Supports
Multiple Currencies

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Suspense Supports Multiple Currencies

When multiple suspense items are attached to a single activity, then OIPA will verify that the currencies of each suspense item match the premium currency selected on the activity. This validation also exists for individual suspense items.

The field section of an existing prototype transaction was altered to allow a payment to accept multiple currencies. The MultiSuspensePrototype transaction contains this modification.

Navigate to **Main Explorer | Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | MultiSuspensePrototype.**

Multiple currencies are now present within the <Currency> element.
<Currency>EUR, USD</Currency>

See also, [Enhanced Suspense Processing](#) prototype information for additional examples of suspense processing.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Suspense Refund

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Suspense Refund Prototype

A transaction was configured to demonstrate the refund of monies to a client. When the transaction is processed as an activity in OIPA, a suspense refund number is automatically generated. This refund number can be used to track the refund in OIPA.

Suspense Refund Number

The suspense refund number is fully configurable and is generated via a dynamic field with "Identifier" <DataType>. Through the use of its <Parts> <Part> sub-elements (with "SEQUENCE", or "VALUE" attributes) the suspense refund number has the capability to contain Prefix, Sequential Value and Suffix components.

Once generated, the suspense refund number may be passed to a variety of locations in OIPA, such as screens, spawned activities, field tables, etc., through the use of available CopyTo rules, Spawn, MaintainSuspense, GenerateSuspense, and other such rules or elements.

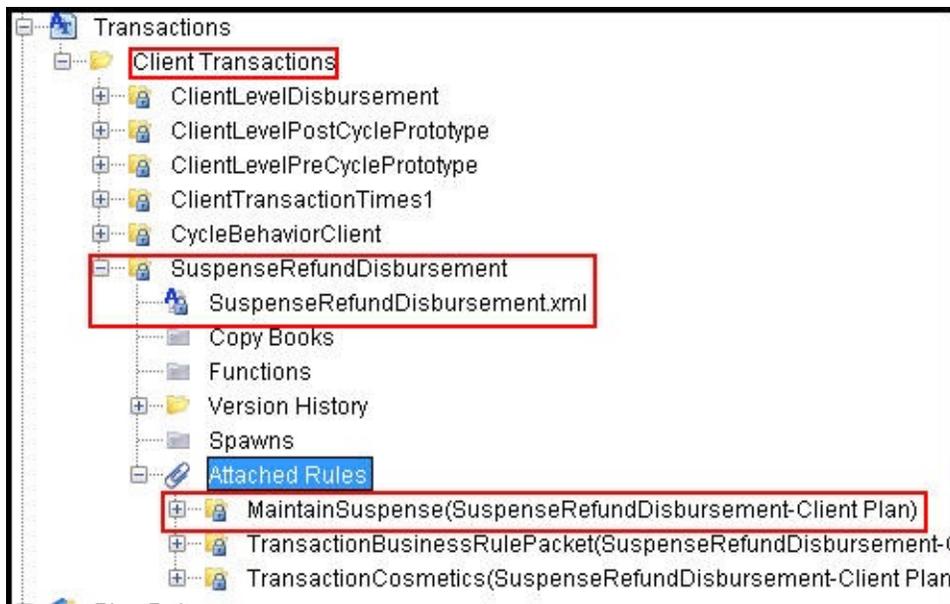
Suspense refund number fields may be used as search criteria in various search screens (Policy Search, Suspense Search, Disbursement Search, Client Search, etc.).

Scenario

A CSR processes a suspense refund activity. After entering all necessary information and clicking **Save**, a suspense refund number is automatically generated by the system and passed to the Suspense screen.

Configuration Requirements

- The **SuspenseRefundDisbursement** transaction is a client level transaction in the Prototype company. It contains the <Suspense> and <Disbursement> elements needed to disburse a suspense refund. It also contains a Suspense Refund Number field and the <Parts> element that defines the format of the refund number. Navigate to **Global Explorer | Transactions | SuspenseRefundDisbursement (Client Plan)** to view the XML for this transaction.
- The **MaintainSuspense** business rule is attached to the SuspenseRefundDisbursement transaction and copies the SuspenseRefundNumber to the Suspense Refund Number field on the Suspense screen.



SuspenseRefundDisbursement and Attached Rules in Main Explorer

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Suspense Refund
Client Level Disbursement

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Suspense Refund Client Level Disbursement

A transaction was configured to generate a client level refund when the activity is processed in OIPA. The new element <DisbursementClient> with the attribute DISBURSEMENTAMOUNT, was added under the <Disbursement> element.

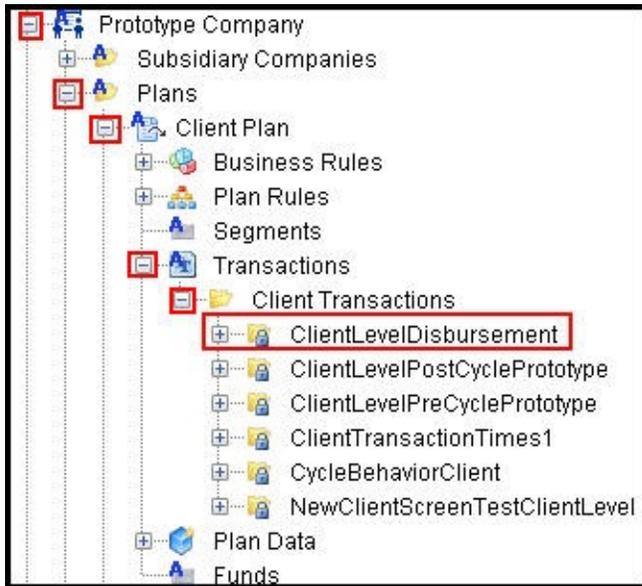
The <DisbursementClient> element identifies the client who will receive the disbursement. The DISBURSEMENTAMOUNT attribute indicates the amount of the disbursement.

XML Example

```
<DisbursementClient DISBURSEMENTAMOUNT="DisbursementAmount">
```

Configuration Requirements

The **ClientLevelDisbursement** transaction works with all field values at the Client Plan level. Navigate to **Main Explorer | Companies | Prototype Company | Plans | Transactions | Client Transactions | ClientLevelDisbursement** to view the XML for this transaction.



ClientLevelDisbursement Transaction in Navigation Tree

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Suspense Field from Activity Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Suspense Field from Activity Prototype

This prototype was created to demonstrate the new SUSPENSEFIELD math variable configuration. The math variable is configured in a Policy-Financial transaction on the Functional Prototype plan.

If the policy number of the suspense item and the policy number where the activity is being added do not match, then an error message will display when you try to add this activity in OIPA.

Prototype Explanation

The following transaction was configured in the Functional Prototype Plan to demonstrate this new functionality.

- The **SuspenseFieldPrototype** transaction was created to demonstrate the comparison of suspense information on a policy and an activity that is being added. The important parts of configuration are explained below. To view configuration, navigate in the Global Explorer to **Transactions | Suspense Field Prototype | Suspense Field Prototype (Functional Prototype Plan)** .
 - The Event section contains an OnSubmit element that defines the action to occur when the activity is added. OIPA will compare the policy number associated with the entered suspense item and the policy number where the activity is being added.
 - The Local Math section identifies the suspense number and the policy number associated with the suspense item. There are also two additional math variables that define the policy number where the activity is being added.

If the policy number of the suspense item and the policy number where the activity is being added do not match, then an error message will display.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password. Make sure the user has privileges to view comments.
2. Create a suspense record. Make sure the policy number on the record DOES NOT match the policy you plan to process this activity against. If the numbers don't match, then you will be able to see the OnSubmit error message, which shows that OIPA tried to match the numbers.
3. Open a policy.
4. Click the **Add Activity** link on the Secondary menu of the Policy screen. This will open the Activity screen.
5. Select the **SuspenseFieldPrototype** activity from the drop down box.
6. Type an amount in the Gross Amount field.
7. Click the Suspense link at the top of the window.
8. Enter the suspense number of the record you created.
9. Click **Ok**.
10. An Error message will be displayed stating that the policy numbers don't match.

You are here: Configuration > Suspense > Configure Suspense > Suspense and Multiple Currencies

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Suspense and Multiple Currencies

When money is paid to an insurance company from a client, it is placed into a suspense account. This account must be able to support the various currencies in which a client could make payment to the insurance company. Suspense functionality allows OIPA to accept payment from clients in various currencies.

Multi-Currency Support

OIPA supports multiple currencies during suspense in OIPA in order to:

- allow the load of Currency Codes to all field types available in Math such as, PolicyField, SegmentField and PlanField.
- carry Currency Code as a logged Math value in AsActivityMath.
- allow reference of a field or MathVariable's currency code.
- allow reference of the original fund currency in order to perform an exchange in Math or calculate Gain/Loss.
- allow currency fields to be spawned.
- calculate accounting exchanges through a link to the Market Maker on the Company.

Scenario

The following scenario demonstrates multi-currency suspense processing.

- The User navigates to the Suspense screen and creates an entry for a policy selecting a specific national currency from a currency drop down box.

Configuration Requirements

There are several components that must be present in order to use suspense in OIPA. They include:

- SuspenseScreen business rule must be overridden at the company level.
- SuspenseSearchScreen business rule must be overridden at the company level.
- transaction with <Suspense> or <Multisuspense> element OR transaction with attached GenerateSuspense business rule.

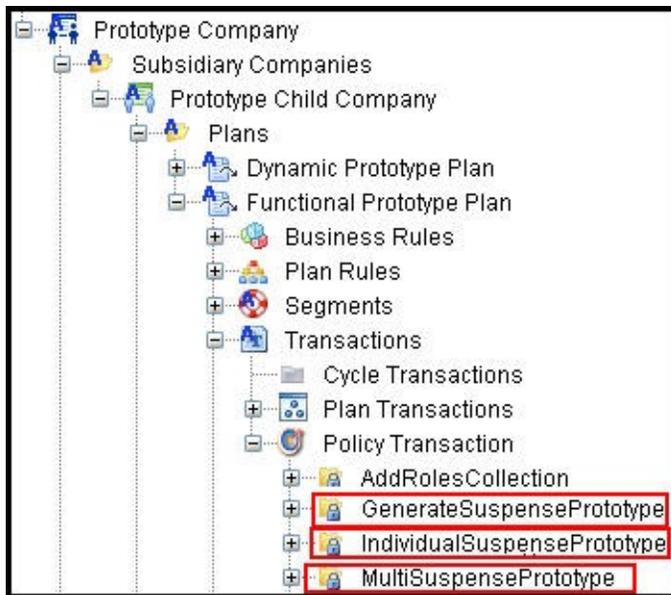
Optional Additional Configuration

- MaintainSuspense business rule can be attached to a transaction to allow the suspense record to be updated when the transaction processes as an activity in OIPA.

Prototype Samples

There are three transactions configured to demonstrate this functionality. Navigate in the Main Explorer to **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions.**

- MultiSuspensePrototype transaction: allows users to enter multiple suspense numbers and amounts on the ActivityDetail screen that corresponds to the records, which have been saved on the AsSuspense table. The currency configured in the transaction must match the currency of the suspense record used.
- IndividualSuspensePrototype transaction: allow users to enter a single suspense number and amount on the ActivityDetail screen that corresponds to a record saved on the AsSuspense table. The user can select the currency associated with the suspense item being used in the transaction. The currency that is selected in the transaction needs to match the currency of the suspense record being used. The MaintainSuspense business rule is attached to demonstrate the ability to update a suspense field during transaction processing.
- GenerateSuspensePrototype transaction: allow users to create a single suspense record saved to the AsSuspense table. The suspense record that is created is demoninated in the plan default currency.



Suspense Transactions in Main Explorer

There are four business rules configured to demonstrate this functionality.

- SuspenseScreen business rule: this rule is overridden at the company level. The enhancements to this rule allow users to select the currency for monies received from a client. Other enhancements to the SuspenseScreen rule include the ability to modify the fixed fields displayed on the screen through configuration. In addition users now have the ability to trigger accounting based on the change of a dynamic suspense field on the SuspenseScreen. Navigate to the Global Rules Explorer and open **Business Rules | Screen | SuspenseScreen | Company Overrides** to see this rule configuration.
- SuspenseSearchScreen business rule: this rule allows users to search for suspense records in a variety of currencies. Navigate to the Global Rules Explorer and open **Business Rules | Screen | SuspenseSearchScreen | Company Overrides** to see this rule configuration.
- GenerateSuspense business rule: this rule allows a suspense record to be generated when an activity is processed in OIPA. Navigate to the Main Explorer and open **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions |**

GenerateSuspensePrototype | Attached Rules to see the configuration.

- MaintainSuspense business rule: this rule is attached to the IndividualSuspensePrototype transaction to allow an activity to update suspense when processed in OIPA. Navigate to the Main Explorer and open **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | IndividualSuspensePrototype | Attached Rules** to see the configuration.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Suspense Search
Screen Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Suspense Search Screen Prototype

This prototype was created to demonstrate the enhanced configuration that is available to manipulate the display of results on the Suspense Search screen.

Prototype Explanation

The SuspenseSearchScreen rule is the only configuration involved in this prototype. An explanation of the enhancements is provided below.

- The **SuspenseSearchScreen** in the Prototype Company shows the optional <Results> section of the configuration, using a common Table definition, to present a results grid that has been manipulated by configuration. To view configuration, navigate in the Global Explorer to **Business Rules | Screen Rules | SuspenseSearchScreen | Company Overrides | SuspenseSearchScreen (Prototype Company)** .
 - Common Table definition will be used within </Results> to present a results grid that demonstrates the ability to configure and then subsequently display both fixed and dynamic fields.
 - Use of the Suspense and SuspenseField Groups is demonstrated in the </Results> </Table> configuration.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password.
2. Click **Suspense | Suspense Records** on the Main menu.
3. Click **Find** at the bottom of the Suspense screen.
4. Type in some search criteria and click **Find**.
5. View the enhanced Results grid at the bottom of the screen.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Title and Label Fields Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Title and Label Fields Prototype

OIPA now supports the ability to create titles and labels for field sections.

Prototype Explanation

The following business rules and transactions were configured in the Functional Prototype Plan, located within the Prototype Company, to demonstrate this new functionality.

Business Rules

- **PolicyScreen:** This business rule contains a Title field that acts as a header for all of the activity's fields and a Label field that acts as a heading for a section of the fields. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Plan Rules | PolicyScreen** and click on the **Fields** pane. The key configuration for this business rule is explained below.
 - A field with a Field Type of "Title," and with a Display Name of "Policy Details"
 - A field with a Field Type of "Label," and with a Display Name of "Policy Screen Label"

Transactions

- **ConfirmationPremium:** This transaction contains a Title field that acts as a header for all of the activity's fields and a Label field that acts as a heading for a section of the fields. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Transactions | ConfirmationPremium** and click on the **Fields** pane. The key configuration for this transaction is explained below.
 - A field with a Field Type of "Title", and with a Display Name of "Payment Details"
 - A field with a Field Type of "Label," and with a Display Name of "Payment Label"

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password.
2. Open a policy or create a shell policy that belongs to the Functional Prototype Plan.
3. On the Policy screen, note the Title field with the display name of "Policy Title" and the Label field with the display name of "Policy Details Label."

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases >
TransactionAllocationScreen Actions and Events Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

TransactionAllocationScreen Actions and Events Prototype

The EventsOnAllocationSupport prototype demonstrates how OIPA allows actions and events in the TransactionAllocationScreen BR.

Scenario

In a specific transaction, the TransactionAllocationScreen needs to be displayed with the EqualPercent checkbox disabled in the From section and available in the To section. Further, using specific trigger actions, the EqualPercent in From section needs to be enabled and errors/warnings displayed based on the % values or amount or no. of units entered in the various funds under From and To sections.

Prototype Configuration

- The transaction 'EventsOnAllocationSupport' under the 'Model Prototype Plan' demonstrates the action/events capabilities through the following actions and events configured in the transaction and the attached TransactionAllocationScreen BR.
 - Calling an event in the TransactionAllocationScreen BR from ONCHANGE event in the activity details: A "CallExternalEvent" Check field is configured. When this check box field is 'CHECKED', the EventBasedOnActivityField event is triggered in the TransactionAllocationScreen BR by using the CALLEXTERNALEVENT.
 - EventBasedOnActivityField CALLEDEVENT in TransactionAllocationScreen BR: The CallTASBREvent is configured in the attached BR that performs the following:
 - Check the AllocationMethod selected in From and To and if they are not the same, throw an error "Allocation Method must be same for both the FROM & TO allocations."
 - Further, the EventBasedOnActivityField event is configured under the MultiField BR used in the same transaction. This is done to demonstrate the precedence when the same CALLEDEVENT occurs in more than one associated business rule of a transaction. Currently MultiFields BR and TransactionAllocationScreen BR allow using CALLEDEVENTs and the precedence is MultiField BR first and then the TransactionAllocationScreen BR if the same exists in both.
 - In this transaction, when the MultiField event CallTASBREvent is triggered, static text "CALLEDEVENT is executed" is assigned to all the rows of the multifold for the TextField field.
- ONLOAD event in TransactionAllocationScreen: An ONLOAD event is configured in the TransactionAllocationScreen BR to disable the EqualPercent checkbox in the From Section and set the EqualPercent checkbox to CHECKED in the To Section.

- ONSUBMIT event in TransactionAllocationScreen: This will get triggered when the transaction is submitted. In terms of precedence - the ONSUBMIT in the activity details screen will be triggered first, next would be any ONSUBMIT events from MultiFields, if any and then the ONSUBMIT event in the TransactionAllocationScreen BR will be triggered.
 - If the target AllocationMethod is Percent, then check if the TotalAllocationAmount is 100%. If not, throw an error "Under TO Allocation Section the Total Percentage must match to 100."

View Prototype in OIPA

1. Log in OIPA using the Prototype Company user ID and password.
2. Click **Policy | New** from the Main menu.
3. Add a shell policy to test the prototype.
4. Click **Add Activity** on the Secondary menu.
5. Select the EventsOnAllocationSupport activity from the Activity drop down box.
6. Make changes to the various fields and process the transaction as explained above to view the functionality.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > TransactionTimes Business Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

TransactionTimes Business Rule

The TransactionTimes business rule has been implemented in order to provide additional controls over when certain transactions can be processed based on the time of day. If this business rule is attached to a transaction, then users can prevent the transaction from being processed as well as update the activity effective date based on certain criteria defined in the business rule.

This attached rule should not be added to the TransactionBusinessRulePacket, but will be configured as a transaction level override. This business rule is a system rule and can be found in the System folder in the Global Rules Explorer.

A complete explanation of the elements, attributes and values used to configure this business rule is included in the XML Configuration Guide. Click **Help** and select the option for the Configuration Guide. This business rule can be found in the System Rules section.

Scenarios

Three scenarios are provided to demonstrate the functionality of this business rule.

- User adds an activity and the default system date will be the next business date according to the AsSystemDate table. This is shown with the DefaultDatePrototype transaction.
- User adds an activity but the time of day determines whether the activity will be added or not. The start and end times are configured directly in the business rule. This is shown with the MarketTimePrototype transaction and the TransactionTimes transaction level override.
- User adds an activity and the activity effective date is set to the next business date when certain criteria are satisfied. This activity uses the company fields to determine the start and end times. This is shown with the UpdateDatePrototype transaction and the TransactionTimes transaction level override.
- User adds 'TransactionTimesAllowAndProhibit' transaction at the policy level. A transaction called "TransactionTimesAllowAndProhibit" is added at the "Functional Prototype Plan" level.

Transaction name: TransactionTimesAllowAndProhibit

Company name: Prototype Individual Child Company

Product Name: Individual Prototype Product

Plan Name :Functional Prototype Plan

Attached BusinessRules: TransactionTimes

Configuration Requirements

In order to use this business rule to perform the scenarios listed above, three pieces of configuration are required:

- configure the TransactionTimes business rule overrides
- [configure security to define which user roles are restricted or enabled](#)
- create the transactions that will control activity processing dates
- TransactionTimes transaction will have business rules attached and is configured with the below mentioned details:

Allow:

- ActionType: Add -When the ActivityEffectiveDate becomes equal to planEffectiveFrom date then the message "Adding transaction is allowed when ActivityEffectiveDate is not equal to PlanEffectiveFrom"
- ActionType: Process - When ActivityEffectiveDate is equal to planEffectiveFrom date then the message "Processing is allowed when ActivityEffectiveDate is not equal to PlanEffectiveFrom".

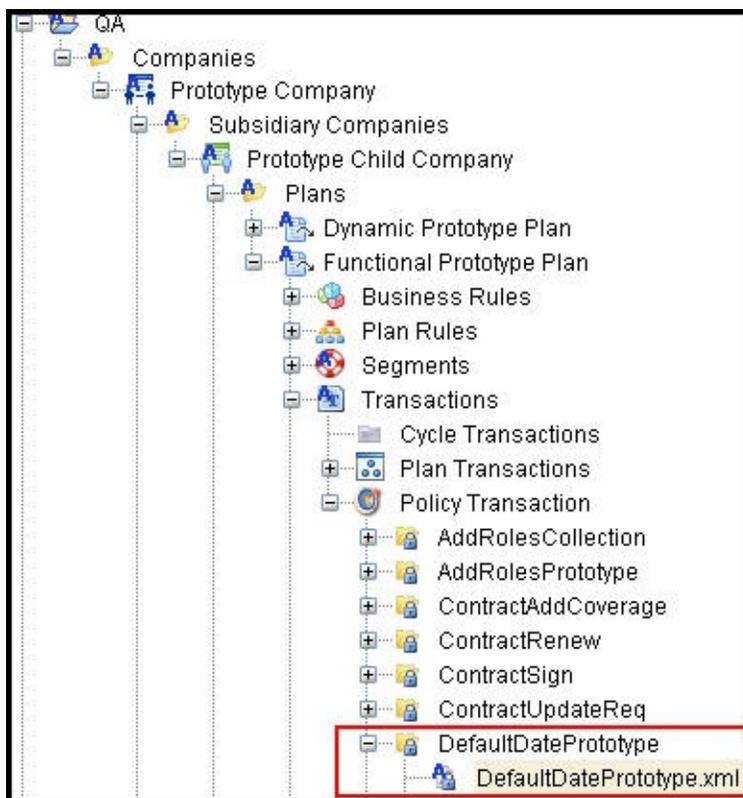
Prohibit:

- ActionType:Reverse - Transaction reversal is prohibited the message reads "Cannot Reverse the transaction".
- ActionType:Update - When the user tries to add the transaction in the time range of 12:00 to 21:00 then, user will be shown this message "Cannot update transactions in the time range of 12:00 to 21:00"

Prototype Examples

In the first scenario, the DefaultDatePrototype transaction is configured with a Value in the Effective Date tag set to NextBusinessDay. When the activity processes in OIPA, the date will be set to the next business date according to the AsSystemDate table. Navigate through the following folders in the Main Explorer to locate the configuration sample:

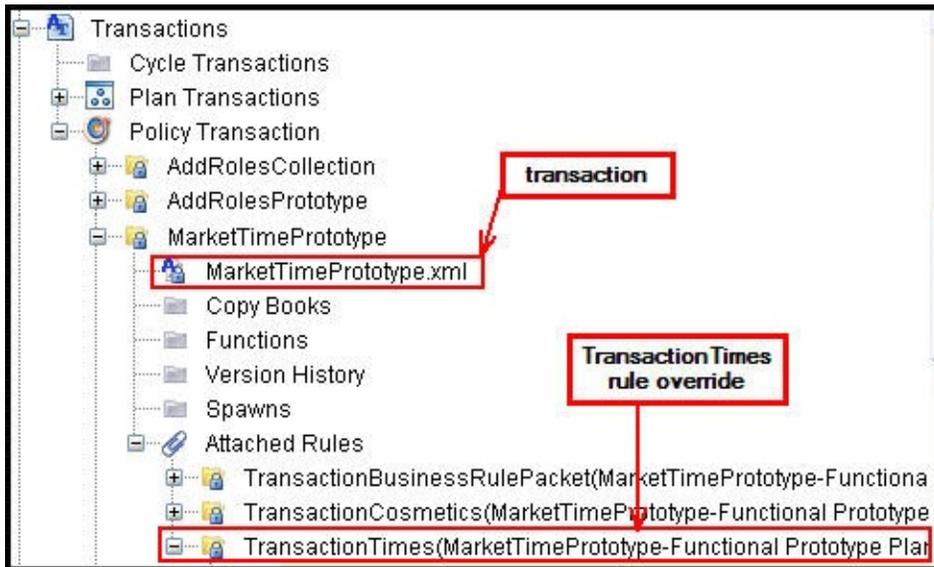
Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | DefaultDatePrototype.



DefaultDatePrototype in Main Explorer

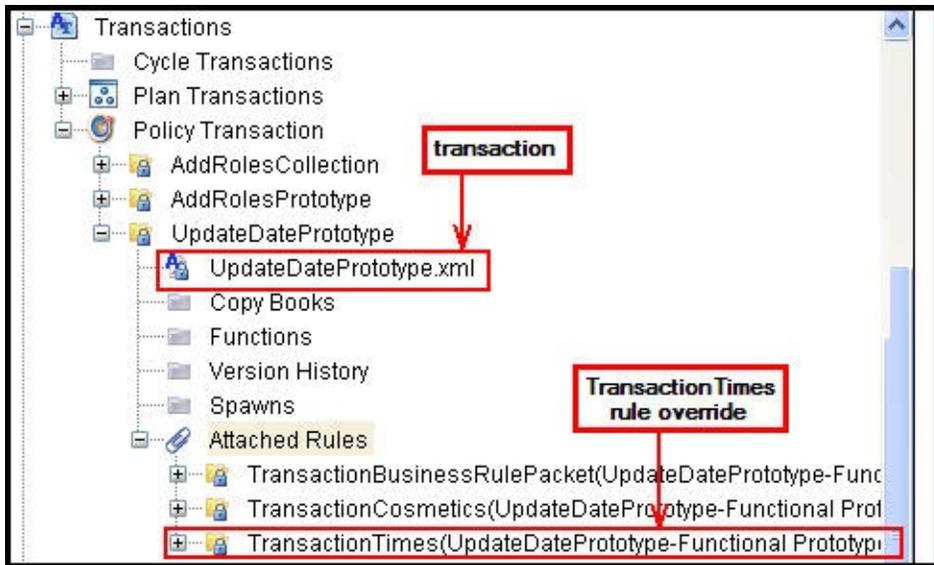
In the second scenario, the MarketTimePrototype transaction is used to set specific times of day when an activity can be updated or added. The TransactionTimes business rule is overridden at the transaction level and attached to the transaction. The rule override can be found in the Main Explorer in the Attached Rules folder under the MarketTimePrototype transaction. The transaction configuration can be found in the Main Explorer under **Companies | Prototype Company | Subsidiary**

**Companies | Prototype Child Company | Plans | Functional
Prototype Plan | Transactions | Policy Transactions |
MarketTimePrototype..**



TransactionTimes Transaction and Override in Main Explorer

In the third scenario, the UpdateDatePrototype transaction can be processed as an activity on a policy to change the effective date. The TransactionTimes business rule is overridden at the transaction level and attached to the transaction. The rule override can be found in the Main Explorer in the Attached Rules folder under the UpdateDatePrototype transaction. The transaction configuration can be found in the Main Explorer under **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | UpdateDatePrototype..**



UpdateDatePrototype Transaction and Rule Override in Main Explorer

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Accounting Enhancements

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Accounting Enhancements

The accounting section now runs with each execution of the activity but it only writes accounting detail once for each Chart of Accounts entry. The criteria to determine when the accounting records are to be run depends on how the Chart of Accounts is configured. COA entries that are based on math variables or non-fund specific allocations (general accounting) will use values in the transaction math and these will write with the first successful execution of the activity (when there are no non-overridden business rule errors). This is standard Accounting processing today.

If the accounting entries are based on fund (separate accounting), then they will wait until all AsValuation records for the activity are moved into an active state (see Valuation status below), there are no non-overridden business rule errors generated from the new PostAssignmentValidateExpressions rule and the element <GenerateAccounting> has the value of **Yes**. All three conditions must be met even though the valuation records may be active before the activity moves into a final active state.

If there are no PostAssignmentValidateExpressions validation errors and all of the valuation records are in an active state, even if the Activity is still in an NUV Pending status, then the Accounting records for each fund can be written. Upon subsequent executions of the activity, the accounting will run again but will check to see if fund accounting records already exist for the activity.

Configuration Requirements

- A **Premium** transaction was configured that uses the PostAssignmentValidateExpressions rule and has a ChartOfAccounts entry by fund. The activity is amount based so valuation will be immediately active upon processing the activity. After effective date and exchange rate validations have been satisfied, the activity will stop processing because of a configured overrideable error in PostAssignmentValidateExpressions. Accounting will generate when this last error is overridden by the user. Navigate to **Global Explorer | Transactions | Premium | Premium (Unit Linked Template)**.

```
<PostAssignmentValidateExpressions>
  <MathVariables>
    <MathVariable VARIABLENAME="Yes" TYPE="VALUE" DATATYPE="TEXT">Yes</MathVariable>
    <MathVariable VARIABLENAME="No" TYPE="VALUE" DATATYPE="TEXT">No</MathVariable>
    <MathVariable VARIABLENAME="WriteAccounting" TYPE="EXPRESSION" DATATYPE="TEXT">No</MathVariable>
    <MathVariable VARIABLENAME="IsValuationActive" TYPE="FIELD" DATATYPE="BOOLEAN">Activity:ActiveValuation</MathVariable>
    <MathIF IF="IsValuationActive = true">
      <MathVariable VARIABLENAME="WriteAccounting" TYPE="EXPRESSION" DATATYPE="TEXT">Yes</MathVariable>
    </MathIF>
  </MathVariables>
  <ValidateExpressions>
    <Expression TYPE="ErrorOnTrue" OVERRIDABLE="Yes" MESSAGE="Accounting is ready to process">WriteAccounting = Yes</Expression>
  </ValidateExpressions>
  <GenerateAccounting>WriteAccounting</GenerateAccounting>
</PostAssignmentValidateExpressions>
```

PostAssignmentValidateExpression Rule that Controls Processing and Accounting for Premium Transaction

- An **InitialFundReallocation** transaction was configured that uses the PostAssignmentValidateExpressions rule and has a ChartOfAccounts entry by fund. The activity is percent based and fund prices are not immediately available so valuation will not be active on initial processing. The activity status will be NUV Pending and accounting records will not be generated. Navigate to **Global Explorer | Transactions | InitialFundReallocation | InitialFundReallocation (Unit Linked Template)**. Once prices are known, the activity will process up to the configured overrideable error in PostAssignmentValidateExpressions. When the user overrides this error, the activity will complete processing and its status will change to Gain/Loss Pending. Accounting records will be generated.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Use Last Known Prices

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

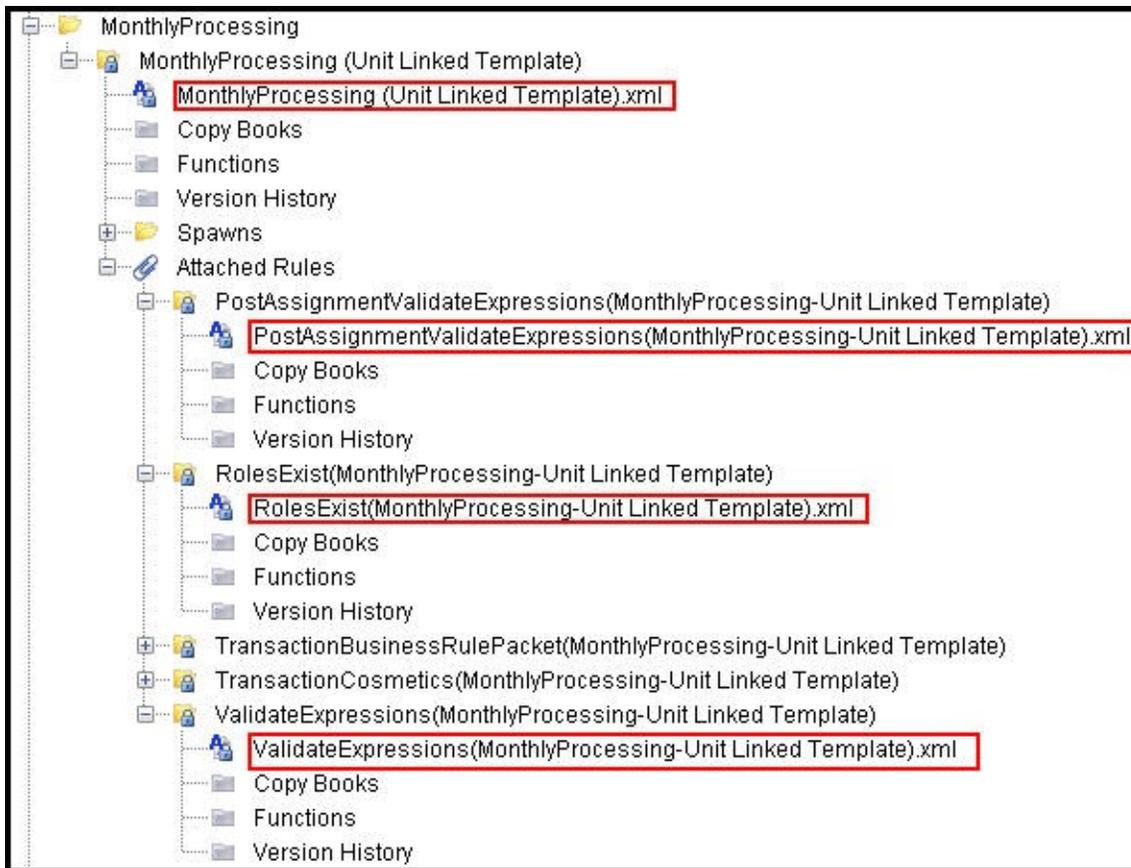
Use Last Known Prices

OIPA needs to be able to use last known fund prices when calculating valuation units and amounts to write during assignment processing. This is accomplished through the new MathStatement function called **FindLastExchangeDatePricing**. The function will find the last known exchange dates from the Market Maker's exchange rate table entries that are equal to or less than the activity's effective date. The allocated funds will use the same last date methodology for price lookup disregarding all offset values.

If the same function is called more than once or both functions are executed in the same activity, then the Assignment logic will reference the output values of the last one executed.

Configuration Requirements

- The **MonthlyProcessing** transaction in the Unit Linked template was modified to demonstrate this functionality. The new [Math Statement](#) function was used to find the last exchange date pricing. Navigate to **Global Explorer | Transactions | MonthlyProcessing (Unit Linked Template)**.
- The **PostAssignmentValidateExpressions** business rule is attached to the transaction. This rule is associated with a transaction as a transaction override. It does not exist in the TransactionBusinessRulePacket. This allows the rule to process each time the activity processes while in an NUV pending state. The rule contains a math section that allows for standard math calculations, and a validation section that supports standard validation expressions.
- The **RolesExists** business rule is attached to this transaction. It tells OIPA that the Insured role must exist on the policy and two fields must be populated: Date of Birth and Sex.
- The **ValidateExpressions** business rule is attached to this transaction. It makes sure the Insured Age is within the acceptable range.



Global Explorer Folder Structure For MonthlyProcessing Transaction

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Math Can Access Valuation Status

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Math Can Access Valuation Status

As part of the new rolling valuation process for Unit Linked functionality, it is possible for the valuation of an activity to be considered in an active state while the activity itself is still in an NUV Pending status. The activity's valuation record becomes active when enough details are known for the funds in the activity allocation and the effects of the activity may be included in subsequent policy valuation such as the policy's Values screen or other activities. . When the valuation becomes active the ActivityGUID will automatically be written to the AsActiveValuationActivity table. Once written, the activity will be included with all systematic valuation calculations for the policy.

From within the Math section of the [PostAssignmentValidateExpressions rule](#), the configuration can check the state of the Valuation records held in memory as the activity processes but before it is committed to the database. This state will tell whether the records to be written to AsValuation have enough information to complete the Accounting process as well as be included in valuation calculations by other screens or activities.

The determination criteria for active Valuation records is:

- If funds are being sold, all the guaranteed prices and valuation amounts must be known.
- If funds are being purchased, all the guaranteed valuation amounts must be known.

Configuration Requirements

When the activity becomes active, an entry will automatically be made on the AsActiveValuationActivity table. In order to see the valuation status for the activity as the activity is processing, two pieces of configuration are required:

- create the transaction that will process assignment values
- configure the PostAssignmentValidateExpressions business rule that includes a field type math variable holding "Activity:ActiveValuation"

The following transactions in the Unit Linked Template check the valuation state in their PostAssignmentvalidateExpressions rules. Navigate to **Main Explorer | Companies | International Holding Company | Subsidiary Companies | International Child Company | Plans | Unit Linked Template | Transactions | Policy Transactions.**

- AutomaticInvestment
- AutoRebalance
- Premium
- Switch
- Withdrawal

XML Example

```
<PostAssignmentValidateExpressions>
  <MathVariables>
    <MathVariable VARIABLENAME="IsValuationActive" TYPE="FIELD"
DATATYPE="BOOLEAN">Activity:ActiveValuation</MathVariable>
  </MathVariables>
</PostAssignmentValidateExpressions>
```

XML Example of PostAssignmentValidateExpressions Business Rule

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > New Switch Assignment Type

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

New Switch Assignment Type

A new assignment type called Switch has been added to OIPA in order to support fund transfer calculations with Unit Linked functionality. Switch functions in much the same way as the existing Transfer assignment except that the underlying calculations are percentage based rather than derived by the calculated amounts. While Transfer will first convert the entered transfer percentages into currency amounts before calculating the unit removals, Switch will apply the transfer percentage directly to the balance of units invested in the fund.

Even though the system calculations are percentage driven, the Switch assignment type supports all allocation methods (Amount, Percent of Fund, and Units etc.).

The switch assignment also supports calculating and removing fees either as a fixed amount or a percentage removed from the surrendered funds on a prorated basis.

Note: All fees are assumed to be in the base currency for the plan.

This new assignment will support an IF attribute to evaluate a condition expression and if true, process the assignment. If false, then it will not process.

Configuration Requirements

- Create the transaction that will process Switch assignment. The Switch transaction was configured in the Unit Linked Template. Navigate to **Global Explorer | Transactions | Switch | Switch (Unit Linked Template)**.
- Configure the MathStatement in the transaction to use either the FindNextExchangeDatePricing or FindLastExchangeDatePricing functions.

Note: This assignment types does not support the use of the MoneyType element as the fee money type is an attribute of the Assignment itself.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Rolling Valuation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Rolling Valuation

As part of the new Unit Linked processing, a new attached business rule called **PostAssignmentValidateExpressions** has been added to support math calculations and business edits that need to occur after transaction assignment processing has completed.

The processing sequence of an activity is now as follows:

1. Suspense
2. Valuation
3. Math
4. Attached Business Rules (existing configurable processing breakpoint)
5. Assignment
6. Post Assignment Business Rules (new configurable processing breakpoint)
7. Disbursement
8. Accounting
9. Spawns

The PostAssignmentValidateExpressions rule will be optionally associated to a transaction by a transaction override on the AsBusinessRules table but not exist in the TransactionBusinessRulePacket itself. This distinction allows the rule to process each time the activity processes while in an (NUV) pending state.

The PostAssignmentValidateExpressions rule is made up of three sections with distinct uses.

1. **Math:** This is where standard math logic can be configured to calculate necessary values. It will have access to any field or logged math variable from the transaction in addition to assignment records calculated to write to AsValuation. Where regular transaction math

does not process again after the activity moves to NUV Pending status, the math of the PostAssignmentValidateExpressions will process with every execution until the activity is Active or Gain Loss Pending. **Note:** Variables in the PostAssignmentValidateExpressions math cannot be logged and are not visible in the processed activity math pane. They are visible in the AsActivity XMLData.

2. **Expressions:** Configured expressions operate the same as the ValidateExpressions rule but use values from the PostAssignmentValidateExpressions math and fields or logged math variables from the transaction. An additional feature is the optional automatic cancellation attribute that will force the system to shadow the activity if the conditional expression resolves to true. Activities that are systematically rejected with this rule still appear in the policy activity list but have a status of "Cancelled". An application of this feature would be if an owner requests a withdrawal of 5% of their policy but days later when the policy value is known, it turns out this calculation is below the plan minimum surrender amount. Rather than notifying users of the business error via an action required report, the system can automatically reject the activity.
3. **GenerateAccounting:** This element holds a simple Yes or No value (literal value or math variable) that tells the system if accounting should process for the activity. When set to Yes, accounting will process and write records for the applicable chart of accounts entries provided a PostAssignmentValidateExpressions error is not thrown. Accounting will run each time the activity processes but will only write to the database once. The value of this element acts as a configurable gate that prevents accounting from writing records prematurely. **Note:** If the transaction writes accounting from the transaction math, then those variables should be logged to ensure they will be available for the accounting detail.

Spawns

- In order to prevent unwanted spawning of duplicate activities, the PostAssignmentValidateExpressions math can create indicators based on existing spawned activities that can be read by the SpawnIF conditions. Similarly, the <From> field values in the spawn section should be able to draw from the PostAssignmentValidateExpressions math in addition to the normal transaction math and fields.
- If an activity is spawned again upon subsequent processing of the parent activity, the spawned activity may cause prior spawned activities to reprocess only if its effective date and/or processing order are earlier than those of its siblings.
-

- **Status Changes**

- If the new PostAssignmentValidateExpressions rule stops the activity from processing, then the activity will go to its prior status. Even if all the prices for the funds are found on the first execution of the activity (from a pending status), and if an error occurs, then the activity will remain in a pending status. If the error is overridden, then the activity will process again (including math and attached rules) and if the same error is given again, then the system will retain that it was overridden and will not stop the activity from executing.

Configuration Requirements

- Create the transaction that will process assignment values.
- Configure the PostAssignmentValidateExpressions business rule override.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Valuation Loop

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Valuation Loop

The new `PostAssignmentValidateExpressions` introduces a new loop type called Valuation. This new type allows math to loop through the values of the Valuation records held in memory as the activity processes but before it is written to the database tables (`AsValuation` and `AsUnitLinkedCalcValue`). The loop is able to reference all valuation records related to the activity; records both in memory and those already committed to the database during prior processing.

The [looping method](#) operates much like the Segment loop in that it only loops through a predefined object (in this case, valuation records in memory). The loop may be filtered at a high level based on FundGUID so that the loop only contains records for a specific fund or it can include the entire valuation dataset. The data available to the loop includes all the columns defined in `AsValuation` as well as `AsUnitLinkedCalcValue` and `AsFund` as a seamless data source.

It is not required to specify an activity or policy for the loop as it will only access the records in memory for the current activity.

Common applications of the loop are:

- Check to see if the activity results in a fund being over allocated (insufficient funds for the transaction).
- Check to see if the total amount of the transaction is within the business defined tolerances.
- Calculate accurate allocation values for spawned activities based on the total transaction amount.

Configuration Requirements

The Withdrawal transaction in the Unit Linked Template uses the valuation loop to check the total removal amount of a fund specified in the activity entry fields. Navigate to **Global Explorer | Transactions | Withdrawal | Withdrawal (Unit Linked Template)**.

- create the transaction that will process assignment values.
- configure the PostAssignmentValidateExpressions business rule that includes a valuation type math loop.

The AggregateFunction (Method=Count) syntax is supported to provide a count of valuation records in memory as defined by the loop math variable (either all records or those containing a specified Fund GUID).

The LoopIndex math variable allows specific calculations on an individual record. The Exit-Loop math variable type is also supported so that a loop may terminate once a predefined condition is met.

To access specific values of a given valuation record, a math variable type of ValuationField is used. The value of this math variable would be a column name from the fund or valuation tables and would use the source array equal to the current loop and an appropriate data type for the column.

Note: This new looping method will only be available to the PostAssignmentValidateExpressions rule and as it only applies to unit linked funds. Point in Time valuation will not be supported.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > Update Requirement Activity Date

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Update Requirement Activity Date

The DeliveryRequirements business rule was enhanced to support a NEWACTIVITYDATE attribute that will specify the Effective Date an activity should use once all requirements are satisfied. The new attribute accepts configured date values from a field or math variable and then sets the activity's Effective Date to the New Activity Date when all the requirements are satisfied. If this attribute is not used then the DeliveryRequirements rule will set the activity's Effective Date to the latest of the non-cancelled individual requirement's CloseDate, when PROCESSIONCOMPLETE is set to **Yes** in the configuration.

Note: The attribute will only accept date values that are greater than or equal to the activity's current effective date.

Scenario for Prototype

A transaction is configured to generate a requirement through the use of the attached GeneratePendingRequirements business rule. The rule will add a requirement to the policy. When the transaction processes, a requirement icon will appear, alerting the user that a requirement needs to be met. Once the requirement is satisfied, the activity will process. The date displayed on the Activity screen is the date defined in the DeliveryRequirements business rule.

Configuration Requirements

There are three business rules that need to be configured:

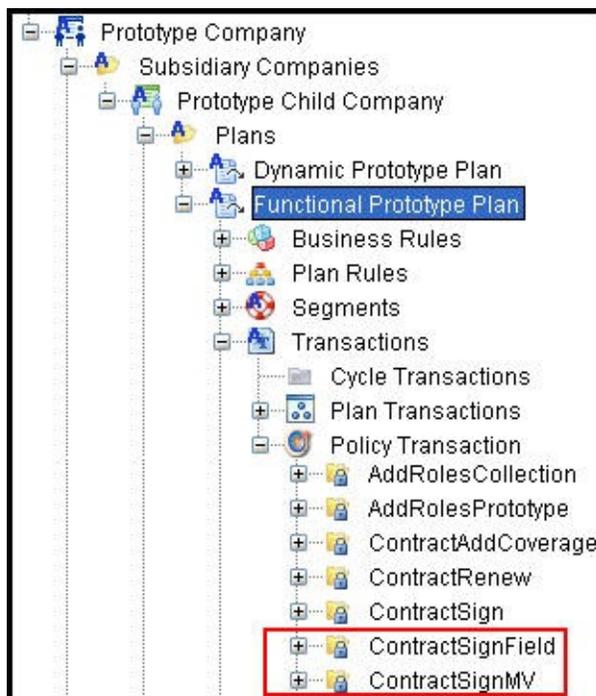
- **ActivityRequirementScreen**: the global rule should exist, but does not need to be configured. This rule may be overridden at the company level and may also be overridden at the transaction level. This rule does not need to contain configuration, as the fixed fields are defined by base Java code.
- **DeliveryRequirements**: this rule should be overridden at the transaction level and listed in the `TransactionBusinessRulePacket`.
- **GeneratePendingRequirements**: this rule should be attached to a transaction and listed in `TransactionBusinessRulePacket` before `DeliveryRequirements`.

Two transactions are configured to demonstrate both the field and math variable options to the `NewActivityDate` attribute.

Prototype Samples

Two new transactions were configured:

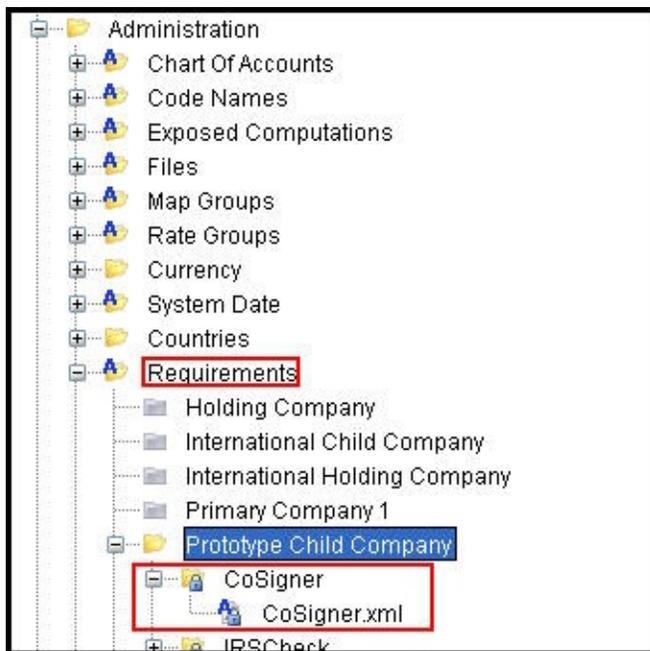
- **ContractSignField**: tests a transaction that passes a field name to the "NewActivityDate" attribute. Navigate to Main Explorer and open **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions**.
- **ContractSignMV**: tests a transaction that passes a math variable name to the "NewActivityDate" attribute. Navigate to Main Explorer and open **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions**.



Transactions that Demonstrate New Requirement Attribute

- There is one requirement definition that is used to demonstrate requirement processing. This definition is referenced from the transaction and represents a requirement that must be filled. The criteria are listed below. Navigate to Admin Explorer and open **Administration | Requirements | Prototype Child Company** to see the requirement definitions.

- CoSign: this requirement needs all three criteria to be met:
 - Date: The activity's Effective Date must be less than March 1, 2010.
 - Checkbox: The CoSign requirement will generate if the ContractSign activity's CoSign Checkbox is unchecked.
 - PlanGUID: PlanGuid must match the PlanGUID of the Functional Prototype Plan. This criterion should always be met when processing an activity in the Functional Prototype Plan.



Requirement Definition in Admin Explorer

- Two business rules were used to demonstrate requirements processing. Navigate in the Global Rules Explorer to **Business Rules | Attached**. Open the folder for each rule and then open the Transaction Overrides folder.
 - GeneratePendingRequirements: This rule will be attached to the transaction generating requirements. In Contract Sign it will define the conditions under which the CoSign requirement is created. This rule will actually generate the requirement.
 - DeliveryRequirements: This rule will be attached to Contract Sign. It will define whether the Effective Date is updated when

the requirements are satisfied.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases >
ValidateExpressions Warning Enhancements Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

ValidateExpressions Warning Enhancements Prototype

OIPA no longer displays an override checkbox for expressions defined as warnings that are returned during activity processing.

Prototype Explanation

The following business rules and transactions were configured in the Functional Prototype Plan, located within the Prototype Company, to demonstrate this new functionality.

Transactions

- **SetValidationMessageDisplay**: This transaction allows users to test the warning functionality of OIPA's ValidateExpressions business rule, which has been enhanced so as not to display an override checkbox for warnings. With this transaction, the user can clearly differentiate between an overridable error and a warning. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Transactions | SetValidationMessageDisplay** and click on the **Fields** pane. The key configuration for this transaction is explained below.
 - An activity combo field is configured to give the user the option to "Test Error" or "Test Warning."
 - "Test Warning" is configured to display the Activity Error pop-up, which will display the warning message, and will not display the override checkbox.
 - "Test Error" is configured to display the Activity Error pop-up, which will display the error message with the override checkbox.

Business Rules

- **ValidateExpressions**: This business rule is attached to the SetValidationMessageDisplay transaction, and allows the user to configure pop-up error messages. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Transactions | SetValidationMessageDisplay | Attached Rules | ValidateExpressions** and click on the **XML Source** pane. The key configuration for this business rule is explained below.
 - One <Expression> element specifies that the expression for Test Error

should be overridable.

- A second <Expression> element specifies that the expression for Test Warning should be a warning, and therefore not overridable.
- **VerificationScreen:** This business rule is attached to the SetValidationMessageDisplay transaction, and allows the user to preview errors and/or warnings that will occur on the activity. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Transactions | SetValidationMessageDisplay | Attached Rules | VerificationScreen** and click on the **XML Source** pane. The key configuration for this business rule is explained below.
 - The <Errors> element is given a value of "Yes," so that the Errors section of the Verification screen is displayed.
 - The <Warnings> element is given a value of "Yes," so that the Warnings section of the Verification screen is displayed.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password.
2. Open a policy or create a shell policy that belongs to the Functional Prototype Plan.
3. Add the SetValidationMessageDisplay activity.
4. Select **Test Error** or **Test Warning**.
5. Click the **Verify** button.
6. If Test Error was selected, note that the error message displays with an override checkbox. If "Test Warning" was selected, note that the warning message displays without an override checkbox.

You are here: Main Explorer > Prototypes > Prototype Samples for Previous Releases > ValuesBlock
Prototype

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

ValuesBlock Prototype

OIPA now supports the ability to configure the Values section of the Activity Detail screen independently of the TransactionAllocationScreen attached business rule. With this new functionality, the ValuesBlock continues to be available for configuration along with the TransactionAllocationScreen business rule. Three transactions are configured in the prototype plans to demonstrate the feature.

Configuration Required

Net Asset Values need to exist for the funds that are selected for the allocations on the current system date to complete processing of the VerifyPremium activity explained below. Otherwise, the activity will go into NUV Pending status.

Prototype Explanation

The following transactions were configured in the Prototype Company to demonstrate this new functionality.

Transactions

- **VerifyPremium** (Functional Prototype Plan): This transaction allows the user to select funds to be used for allocations. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Transactions | VerifyPremium** and click on the **XML Source** pane. The key configuration for this transaction is explained below.
 - The <FundAllocations> element has the SHOWVALUATION attribute set to "Yes." This indicates that the Values section of the Activity Detail screen's Allocation tab should be displayed.
 - The <ValuesBlock> element creates a separate "Values" section on the Allocation tab of the Activity Detail screen. This section displays the values on the policy.
- **ValuesBlockNoAllocation** (Functional Prototype Plan): This transaction allows the user to view the values of all the current invested funds specific to the policy. To view the prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Functional Prototype Plan | Transactions | ValuesBlockNoAllocation** and click on the **XML Source** pane. The key configuration for this transaction is explained below.
 - No <FundAllocation> element is present in the transaction configuration.
 - No TransactionAllocationScreen business rule is attached to the transaction.
 - The <ValuesBlock> element is present in the transaction configuration, which causes a separate Values tab to display on the Activity Detail screen.
- **ValuesBlock** (Model Prototype Plan): This transaction is used for investing money in the form of additional contributions by the user. To view the

prototype configuration, navigate in the Main Explorer to **Companies | Prototype Company | Plans | Model Prototype Plan | Transactions | ValuesBlock** and click on the **XML Source** pane. The key configuration for this transaction is explained below.

- The TransactionAllocationScreen business rule is attached to the transaction, which specifies the configuration for the Activity Detail screen's Allocation tab.
- The <ValuesBlock> element creates a separate "Values" section on the Allocation tab of the Activity Detail screen. It is configured to display the Fund Name, Cash Value and Number of Units within this section.

View Prototype in OIPA

1. Log in to OIPA using the Prototype Company user ID and password.
2. Open a policy or create a shell policy that belongs to the Functional Prototype Plan.
3. Click **Add Activity**.
4. Select the **VerifyPremium/ValuesBlockNoAllocation/ValuesBlock** transaction.
5. Note that the Values section displays on the appropriate tab.

You are here: Configuration > Transactions > View PDF Documents

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

View PDF Documents

OIPA allows policy-specific PDF documents to be generated and viewed with Crystal Reports. A Document tab was added to the Activity Results screen in OIPA. Documents that are generated by activities can be viewed from this tab.

Scenario

A CSR processes an activity that is configured to generate a policy-specific PDF document. The CSR opens the Activity Results screen for the processed activity that generated the document and clicks the Document tab. The generated document is displayed.

Configuration Requirements

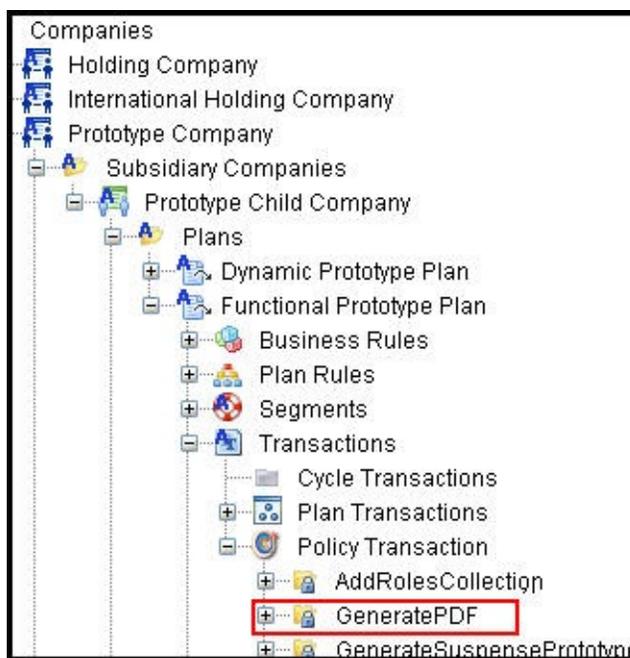
The following configuration requirements are necessary to implement this functionality.

- transaction configured to generate a PDF file.
- ExternalProcess rule attached to the transaction. This rule is configured to trigger the GenerateDocument rule.
- GenerateDocument business rule overridden at the transaction level. This rule is not listed in the TransactionBusinessRulePacket for GeneratePDF. GenerateDocument calls a template that is located on the server to dynamically generate a PDF file based on input gathered from a SQL query in the GenerateDocument rule.
- file template added to the server at the path specified in the PAS.properties file. This file name should match the name specified in the GenerateDocument configuration.

Prototype Samples

One transaction was configured to demonstrate this new functionality. Navigate in the Main Explorer to **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | GeneratePDF**.

- **GeneratePDF**: this transaction is configured to simply generate a PDF. No calculations or other functions are performed by the transaction.



GeneratePDF Transaction in Main Explorer

Two business rules are needed to demonstrate this new functionality.

- **ExternalProcess**: This rule calls the DocumentGenerator code extension, which will in turn run the GenerateDocument rule. This rule should be overridden at the transaction level and included in the TransactionBusinessRulePacket.



ExternalProcess and GenerateDocument Business Rules in Main Explorer

- **GenerateDocument:** This rule calls a template that is located on the server to dynamically generate a PDF file based on input gathered from a SQL query in the rule. This rule should be overridden at the transaction level but **NOT** listed in the TransactionBusinessRulePacket for the GeneratePDF transaction.

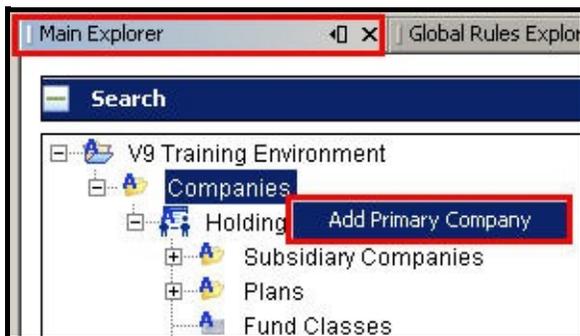
You are here: Main Explorer > Companies > Add New Companies

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Add New Companies

Primary companies and secondary companies can be created from the **Main Explorer** tab. All company information is written to three tables within the database: AsClient, AsCompany and AsRole.

Once a new company is added, [security must be assigned](#) to it or it will not be visible in OIPA.



Company Right-Click Option



Once a new company is created, it cannot be deleted through the Rules Palette.

Steps to Add a New Primary Company

1. From **Main Explorer**, right-click on the **Companies** folder and select **Add Primary Company**.
2. Enter the company name.
3. Select the currency code from the drop down list. If the currency you are looking for is not listed, navigate to **Admin Explorer** and open the [Currency Editor](#). You can add a new currency from this editor.
4. Select a [Market Maker](#) from the drop down list.
5. Select the effective date for that company.
6. Select **Finish** after you have entered the information. Your new company should display directly under the environment folder.

Steps to Add a New Subsidiary Company

1. From **Main Explorer**, right-click on the subsidiary company folder and select **Add Subsidiary Company**.
2. Enter the company name.
3. Select the currency code from the drop down list. If the currency you are looking for is not listed, navigate to **Admin Explorer** and open the [Currency Editor](#). You can add a new currency from this editor.
4. Select the effective date for that company. This is not a required field.
5. Select **Finish** after you have entered the information. Your new subsidiary company should display under the primary company folder.

After the company is created and security is added, [plans](#) and [transactions](#) can be added.

You are here: Main Explorer > Companies > Edit Company Properties

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Edit Company Properties

After a company has been created, you can edit company properties such as the company name, the currency code associated with the company, the Market Maker and the effective date. The steps necessary to edit company properties are shown below.

Steps to Edit Company Properties

1. From the Main Explorer window, right-click on the company name and select **Edit Company**. The Edit Company window will appear.
2. Make the necessary changes to the company properties by clicking on the drop down box next to each property.
3. Select **Finish**. This will save your changes to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Main Explorer > Companies > Company Data Fields

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Company Data Fields

IMPORTANT: The CompanyScreen business rule must be created before working with Company data.

Each company has a Company Data folder, which stores constant values for a company that are not based on any criteria. This information is then available in rules through the use of a MathVariable TYPE=FIELD with *Company:Name of Company field* lookup, and if needed, is available through SQL calls.

Example: <MathVariable VARIABLENAME="MyCompanyFieldMV"
TYPE="FIELD"
DATATYPE="TEXT">Company:MyCompanyField<MathVariab.

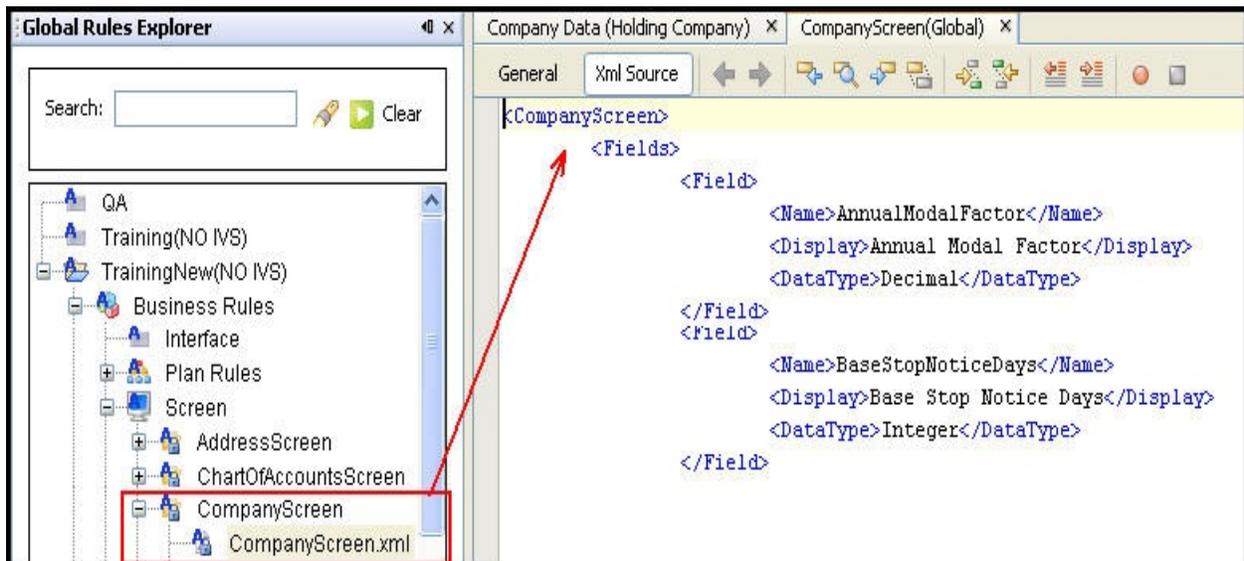
The fields that hold the constant values are defined in the CompanyScreen business rule. They can only be configured at the global level. When company data fields need to be changed or configured , select the [Global Rules Explorer](#) and locate the CompanyScreen rule.

Fields that hold currency values are defined with <DataType>Money</DataType> in the CompanyScreen business rule and will display a currency field for entry in the Rules Palette **CompanyData** node. Rounding and truncation guidelines for values are held in [AsCurrency](#).

The steps to update field information are listed below. The values stored in these fields, [Company Data Values](#), can be changed through the Company Data node in the Main Explorer.

Steps to Update Company Data Fields

1. Open the Global Rules Explorer.
2. Open the **Business Rules | Screen | CompanyScreen** nodes.
3. Right-click on the CompanyScreen XML file and select **Check-out**.
4. Use the XML Source Pane to update the field information.
5. Right-click on the XML file and select **Check-in**. This will save your changes to the database.



Company Screen Business Rule Open in Configuration Area

You are here: Main Explorer > Companies > Company Data Values

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Company Data Values

Each company has a Company Data folder, which stores constant values for a company that are not based on any criteria. The values included in the Company Data folder are determined solely by the CompanyGUID and do not require any additional criteria. If a second criterion is needed, such as state code, then [map groups](#) should be used to store the value.



The CompanyScreen business rule must be created before working with Company Data.

When the Company Data file is opened in the Configuration Area, the Company Fields pane is available. The fields in this pane are extracted from the CompanyScreen business rule. When a value is entered into one of the fields, the value is stored in AsCompanyField. Company fields data represents the information that is generic to the company, such as limits, factors, fees, price offsets, etc.

This information is then available in rules through the use of a math variable that is TYPE=FIELD with *Company:Name of Company field* lookup, and if needed, is available through SQL calls.

Example: <MathVariable VARIABLENAME="MyCompanyFieldMV" TYPE="FIELD" DATATYPE="TEXT">Company:MyCompanyField<MathVariable>

Fields defined as <DataType>Money</DataType> in the CompanyScreen business rule display a currency field for entry in the Rules Palette Company Data node. Truncation and rounding methods are defined in [AsCurrency](#).



Company Data should not be changed once in production unless an error was made in the entry of the information or unless

changing the information will not affect prior policy activity calculations. For example, a minimum value could be reduced, but a factor for a premium deduction charge should not be changed. Any values that could change in the future and might affect calculations should be stored using the [map_group](#) tables.

Steps to Update Company Data Values

1. Expand the Company folder that needs to be updated.
2. Expand the **Company Data** folder.
3. Select and right-click on the company data XML file.
4. Select **Check-out**. The company fields will display in the Configuration Area.
5. Enter the new value in the appropriate field.
6. Select and right-click on the company data XML file.
7. Select **Check-in** to update the database with the changes.

You are here: [Main Explorer](#) > [Products](#) > [Product Overview](#)

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Product Overview

Products are the highest level of coverage packaged to include all possible plan offerings, benefit packages, features and ranges. Product and Child Product information is stored in AsProduct.

Plans inherit data and rules from a Product or Child Product and only contain the benefit options for a specific group of employees. While Products contain the entire pool of options, plans contain a specific set of benefits for a specific set of employees.

Security Considerations

Security is not necessary for Products since they are not visible in the Oracle Insurance Group Policy Administration (OIGPA) system. Instead, users will see the more narrow plan options.

Configuration Considerations

Products and Plans have a close relationship, so it is important to understand their connections before beginning configuration. The following are important considerations for product and plan configuration:

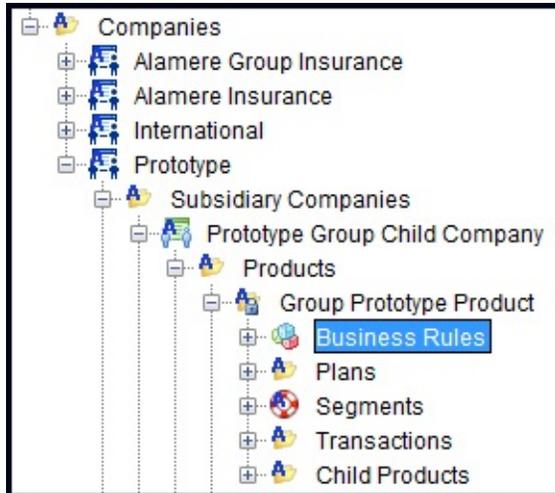
- A Product should define only those fields expected to apply to all Child Products. All field values defined for a Product are inherited by all of its Child Products. If additional fields are needed, it is recommended that multiple Child Products be created.
- When Child Products are created, only override/replacement values will be physically owned by and stored in the database for the affected Child Product.
- Child products do not represent a separate override level from products—child product records will be stored in AsProduct exactly the same as product records. The fundamental difference between a product and child product is that each child product has a relationship to a product.
- Any new fields added to a Child Product (i.e. not defined on and inherited from parent Product) require an override of the PlanScreen business rule at the Child Product level for that Child Product.
- A plan is linked to a Child Product or Product and will inherit all configured information. A plan level override allows the plan to change inherited Product information.
- The PlanScreen business rule defines what is displayed to the OIGPA user. When the PlanScreen is loaded through the UI the system gathers all of the fields defined at the Product, Child Product, and Plan level and displays the highest overridden instance of each.

All information entered on the Plan screen in OIPA is stored in AsPlanField.

See the [Product-Plan Data Inheritance](#) page for detailed information on the inheritance of data through the Product/Child Product/Plan structure.

Product Navigation Tree

Product details can be viewed through the Main Explorer tab. All product information is stored according to Primary Company. Open the company folder to the lowest level (including all subsidiary companies) to view the Product node. An explanation of all folders included in the Product node is provided below.



Group Product Navigation Tree

- **Business Rules:** Any rules that are attached to the product, interface, screen, system, user defined, calculate and CopyBooks.
- **Plans:** specific options from the product that are bundled together for eligible employees. New plans can be created from this node using the right-click menu. Plans created at the agreement level in OIPA will also appear under this node.
- **Segments:** All segments attached to the product. New segments can be created from this node using the right click menu.
- **Transactions:** All transactions attached to the product. New transactions can be created from this node using the right-click menu.
- **Child Products:** Variations of the primary product that maintain a relationship to the primary product.

You are here: Group Information > Creating and Editing Products

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Creating and Editing Products

A product is defined as the line of business or highest level of coverage packaged to include all possible plan offerings, benefit packages, features and ranges. Dental, Life, Health or Disability are examples of products. This page contains information on creating and editing products and child products in the Rules Palette.

Products

Creating Products

To create a new product:

1. Open the **Main Explorer** window.
2. Expand the **Companies** node and expand the relevant company node and subsidiary company node for the subsidiary company to which the product should belong.
3. Right-click on the **Products** node and select **Add Product**. The **New Product** wizard will open. The **Primary Company Name** and **Subsidiary Company Name** fields will already be populated with the names of the primary company and subsidiary company chosen in step 2.
4. Enter a name for the product in the **Product Name** field.
5. Enter a description of the product in the **Product Description** field.
6. Enter a date when the product should take effect in the **Effective Date** field.
7. Enter a date when the product should no longer be valid in the **Expiration Date** field.
8. Click **Finish**.

After a new product is created with the New Product wizard, a record will be created and stored in the AsProduct database table.

Editing Products

To edit an existing product:

1. Open the **Main Explorer** window.
2. Expand the **Companies** node, expand the relevant company node and subsidiary company node for the subsidiary company to which the product that requires editing belongs.
3. Expand the **Products** node.
4. Right-click on the product that requires editing and select **Edit Product**. The **Edit Product** wizard will open.

5. The details below will be available for editing. Edit the relevant information.

- Product Name
- Product Description
- Effective Date
- Expiration Date

6. Select **Finish**.

Child Products

Creating Child Products

Additional child products can be created under each product and child product. To create a new child product:

1. Open the **Main Explorer** window.
2. Navigate through the relevant company, subsidiary company and product/child product nodes and expand the node for the product/child product to which the child product should belong.
3. Right-click on the **Child Products** node and select **Add Child Product**. The **New Child Product** wizard will open. The **Primary Company Name**, **Subsidiary Company Name** and **Product Name** fields will already be populated with the names of the primary company, subsidiary company and product chosen in step 2.
4. Enter a name for the child product in the **Child Product Name** field.
5. Enter a description of the product in the **Child Product Description** field.
6. Enter a date when the child product should take effect in the **Effective Date** field.
7. Enter a date when the child product should no longer be valid in the **Expiration Date** field.
8. Click **Finish**.

After a new child product is created with the New Child Product wizard, a record will be created and stored in the AsProduct database table. Child products do not represent a separate override level from products—child product records will be stored in AsProduct exactly the same as product records. The fundamental difference between a product and child product is that each child product has a relationship to a product. Parent and child product relationship details are stored in the AsProductRelationship database table.

Editing Child Products

To edit an existing product:

1. Open the **Main Explorer** window.
2. Navigate through the relevant company, subsidiary company and product/child product nodes and expand the node for the product/child product to which the child product belongs.
3. Expand the **Child Products** node.
4. Right-click on the child product that requires editing and select **Edit Product**. The **Edit Product** wizard will open.
5. The details below will be available for editing. Edit the relevant information.
 - Child Product Name
 - Child Product Description
 - Effective Date
 - Expiration Date
6. Select **Finish**.

You are here: Group Information > Product Plan Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Product Plan Overview

Products, plans and agreements are all interrelated in a Group insurance setting. The **product** is the highest level of coverage containing all possible plan offerings. The **plans** are more specific selections that the Group Customer selects from the Product to offer to eligible members. The **agreement** contains the contract information detailing the plan selections.

Product

A product is defined as the line of business or highest level of coverage packaged to include all possible plan offerings, benefit packages, features and ranges. Dental, Life, Health or Disability are examples of products.

A product can be broken down into distinct child products. For example, a Dental Product can be broken down into child products such as Dental DMO or Dental PPO. At the Plan Coverage level, a Dental DMO or Dental PPO could additionally offer a Hi, Low, Preventative or Comprehensive coverage.

The data associated with a product contains all options, rules, rates and events necessary to define every variation of the product that may be offered to a customer. Products, like plans, can be organized in a hierarchical fashion to create product “templates” that are subsets of the parent product.

The templates inherit rules and data from their parent product and make it easy to pre-define the subsets of options and data that will be offered to a group insurance customer. A child product or template will, unless otherwise overridden, inherit the data values and rules from the parent template or product.

See the [Creating and Editing Products](#) page for more information on working with products.

Plan

A plan is a specific selection of product offerings selected by the Group Customer to offer to groups of employees. A plan can also be described as the collection of specific benefits and rates packaged when a Group Customer selects from the Carrier's product ranges. A plan is an instance of the product.

The Group Customer decides which members the plan(s) is offered to - all active employees, or a subset of employees, known as a [class](#). Plans are offered during the enrollment period to either all active employees or a class of employees.

Plans are created as Group Customer specific instances of a product and inherit the rules selected from the product. Plan inheritance allows the field configuration defined at the Product and Plan level overrides of the PlanScreen business rule to be displayed in OIPA on the screen at the same time with no impact to the appearance of the fields.

A Class Sub-Plan is a subset of plan benefits available for election and enrollment by a participant. This is also known as Plan Coverage. The Sub-Plan, or coverage, can be attached and removed via an association to the class.

Database Relationship Between Products and Plans

A plan is an instance of the product. The **ProductGUID** relates **AsPlan** to **AsProduct**.

The plan is a subset of what is contained within AsProduct.

Product-Plan Data Inheritance



The following information only applies when the Product structure is in use.

For the following entities, a Plan will inherit configuration defined in a Product-level override:

- Plan Fields
- PlanScreen business rule
- Segments

For example if “Product A” and “Plan A” are directly related to each other, and each has it’s own override of the PlanScreen business rule, the PlanScreen displayed in the application will be the combination of the two overrides because of the direct relationship. This data inheritance works exactly the same for Plan Fields and Segments used throughout the system.

The data inheritance also functions in a similar way if the relationship between the Product and Plan is indirect. For example, if “Product A” has a direct relationship to “Child Product B” and “Child Product B” has a direct relationship to “Plan A,” but the PlanScreen Business Rule only has overrides for “Product A” and “Plan A,” the PlanScreen that is displayed in the application will be the combination of the two overrides because of the indirect relationship between “Product A” and “Plan A.” Again, this data inheritance works the same for Plan Fields and Segments.

Fields configured in multiple related overrides of the PlanScreen will display in the system UI in the following order:

1. Fields defined at the Product level
2. Fields defined at the Child Product level
3. Fields defined at the Plan level

When a field with the same name is present in multiple related overrides,

the field will display on the screen at its earliest definition. The field will not be able to be moved, only hidden.

Agreements

Under the Group Customer context, an Agreement links a Product to a Group Customer creating a Plan. Under the Agreement Definition, the PRODUCT and PLAN attributes define whether one or more Products and one or more Plans can be linked to an Agreement. Under each agreement, ability is provided to have more than one plan which can then be offered to the group members basis eligibility and other business constraints applicable.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Group Information > Product-Plan Data Inheritance

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Product-Plan Data Inheritance



The following information only applies when the Product structure is in use.

For the following entities, a Plan will inherit configuration defined in a Product- and/or Child Product-level override:

- PlanScreen business rule
- Segments

For example if “Product A” and “Plan A” are directly related to each other, and each has its own override of the PlanScreen business rule, the PlanScreen displayed in the application will be the combination of the two overrides because of the direct relationship. This data inheritance works exactly the same for Plan Fields and Segments used throughout the system.

The data inheritance also functions in a similar way if the relationship between the Product and Plan is indirect. For example, if “Product A” has a direct relationship to “Child Product B” and “Child Product B” has a direct relationship to “Plan A,” but the PlanScreen Business Rule only has overrides for “Product A” and “Plan A,” the PlanScreen that is displayed in the application will be the combination of the two overrides because of the indirect relationship between “Product A” and “Plan A.” Again, this data inheritance works the same for Plan Fields and segments.

PlanScreen Data Inheritance

Fields are the only aspect of the PlanScreen business rule that are able to utilize data inheritance. There are several important points to consider when configuring a PlanScreen business rule that will be inherited within the Product-Plan structure:

- When multiple related overrides at the Product, Child Product, and/or Plan level of the PlanScreen exist, all of the fields configured in the related rules will be displayed in the system UI.
- Fields configured in multiple related overrides of the PlanScreen will display in the system UI in the following order:
 1. Fields defined at the Product level
 2. Fields defined at the Child Product level
 3. Fields defined at the Plan level
- When a field with the same name is present in multiple related overrides, the field will display on the Plan screen in OIPA at highest override level where it's defined. The field will not be able to be moved or deleted by lower-level configuration, only hidden.
- When only one override at the Product, Child Product and/or Plan exist, then only the fields configured in that rule will be displayed in the system UI.
- An optional "Final" configuration element can be added to any configured field to prevent a lower level override from overriding the Field configuration. See the [Field Properties](#) page for more information on configuring fields.

Segment Data Inheritance

The following pieces of segment configuration are able to utilize data inheritance:

- Fields
- Events
- ScreenMath
- Actions
- Roles
- RoleViews

See the sections below for detailed information on configuring these aspects of segments when data inheritance is being utilized.

Fields

When multiple related segment overrides at the Product, Child Product, and/or Plan level exist with the same name and type, all of the fields configured in the related rules will be displayed in the system UI.

- If a field with the same name is present in multiple related overrides, the field configuration at the lowest override level will be displayed in the system UI. The field will not be able to be deleted by lower-level configuration, only hidden.
- An optional "Final" configuration element can be added to any configured field to prevent a lower level override from overriding the Field configuration. See the [Field Properties](#) page for more information on configuring fields.
- When only one override at the Product, Child Product and/or Plan exist, then only the fields configured in that rule will be displayed in the system UI.

Buttons

Segment button configuration will not support data inheritance. Each override of the segment will need to define a <Buttons> configuration

section. When multiple related overrides of a single segment at the Product, Child Product and/or Plan level have been created, only the buttons configured in the lowest level override will be displayed in the system UI.

Actions

Actions sections defined in multiple related overrides of a single segment at the Product, Child Product, and/or Plan level that have unique ActionSet IDs will be combined to form a single Actions section.

If multiple Action sections with the same Action ID are present in related overrides, only the lowest level override's Action configuration with the ID conflict will be utilized by the system.

Events

The <Events> configuration section can be defined in multiple related overrides of a single segment at the Product, Child Product, and/or Plan level, all of which will be combined to form a single Events section.

- For ONLOAD Events, only the Event configuration in the lowest level override that is of the type ONLOAD will be utilized by the segment, because only one ONLOAD section can exist per screen.
- For ONCHANGE Events, all of the ONCHANGE Events that operate on a unique field will be utilized by the PolicyScreen.
 - If two ONCHANGE Events with the same field specified are present in multiple related overrides, only the lowest level override will be utilized.
- For ONCHANGE Events, all of the ONCHANGE Events that operate on a unique field will be utilized by the PolicyScreen.
 - If two ONCHANGE Events with the same field specified are present in multiple related overrides, only the lowest level override will be utilized.
- For ONSUBMIT Events, only the Event configuration in the lowest level override that is of the type ONSUBMIT will be utilized by the segment, because only one ONSUBMIT section can exist per screen.

ScreenMath

The ScreenMath sections defined in multiple related segment overrides at the Product, Child Product, and/or Plan level that have unique Math IDs will be combined to form a single ScreenMath section.

If multiple Math sections with the same Math ID are present in related overrides, only the lowest level override's Math configuration with the ID conflict will be utilized by the system.

Roles

Roles configuration can be defined in multiple related overrides of a single segment at the Product, Child Product, and/or Plan level, and will be combined to form a single roles section.

- Each role that is configured with a unique RoleCode will be appended to the list of roles displayed on the Segment Role screen.
- Roles can be added to the list, but not deleted by a lower-level override.
- When multiple roles with the same RoleCode are configured in related overrides of a single segment, only the role configuration in the lowest level override will be used by the system.
- Segment roles configured in multiple related overrides of a single segment will display in the system UI in the following order:
 1. Roles defined at the Product level
 2. Roles defined at the Child Product level
 3. Roles defined at the Plan level
- An optional **Final** configuration setting can be added to any configured field to prevent a lower level override from overriding the role configuration. See the [Create a New Segment](#) page for more information on creating and configuring segment roles.
- A role may be hidden using the **Hidden** configuration setting, but not deleted via a lower-level override's configuration. See the [Create a New Segment](#) page for more information on creating and configuring segment roles.
- The **Final** setting will take precedence over the **Hidden** setting. If a role has

Final set to **Yes**, a lower level override will not be able to hide the Role.

RoleViews

Segment RoleViews configuration that is defined in multiple related overrides of a single segment at the Product, Child Product, and/or Plan level will be combined by the system to form the list of available RoleViews.

- Multiple RoleViews configured in related overrides will be combined to form a single RoleViews configuration section as long as each RoleView has a unique View Name, and as long as the RoleViews are used by different Roles.
- When multiple related overrides contain RoleView configuration with the same View Name, the RoleView in the lowest level override will be used by the system.
- Only one RoleView can be configured for use by all roles. As a result, the lowest level override of segment that contains RoleView configuration used for all roles will be used by the system.
- Each uniquely-named RoleView will display in the **View** combo box on the SegmentRoleScreen in the following order:
 1. RoleViews configured at the Product level
 2. RoleViews configured at the Child Product level
 3. RoleViews configured at the Plan level

You are here: Main Explorer > Plans > Plans Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Plans Overview

Each individual plan in the Rules Palette has six main categories of data. You can create new plans or edit existing plans from the Main Explorer tab. You can also modify plan data [fields](#) and [values](#).

There are three database tables that store plan information.

1. **AsPlan**: stores the information entered when a new plan is created.
2. **AsPlanFields**: stores values for the plan. These values can be edited through the Plan Data folder in the Main Explorer.
3. **AsPlanGroup**: stores GUIDs of the parent and child plans.

Six Main Categories of Data

1. **Business Rules:** Any rules that are attached to the plan, interface, screen, or system, rules that are user defined, and rules that apply to Calculate and CopyBooks.
2. **Plan Rules:** Any rules that are overridden at the plan level and are required for a plan to run in the application.
3. **Segments:** All the segments attached to the plan.
4. **Transactions:** All the transactions attached to the plan.
5. **Plan Data:** Constant values for a plan that can be used for configuration calculations.
6. **Funds:** Only for plans that use funds. This will store fund information.



Plan Folders

When a new plan is created, the following plan rules are automatically attached to the plan.

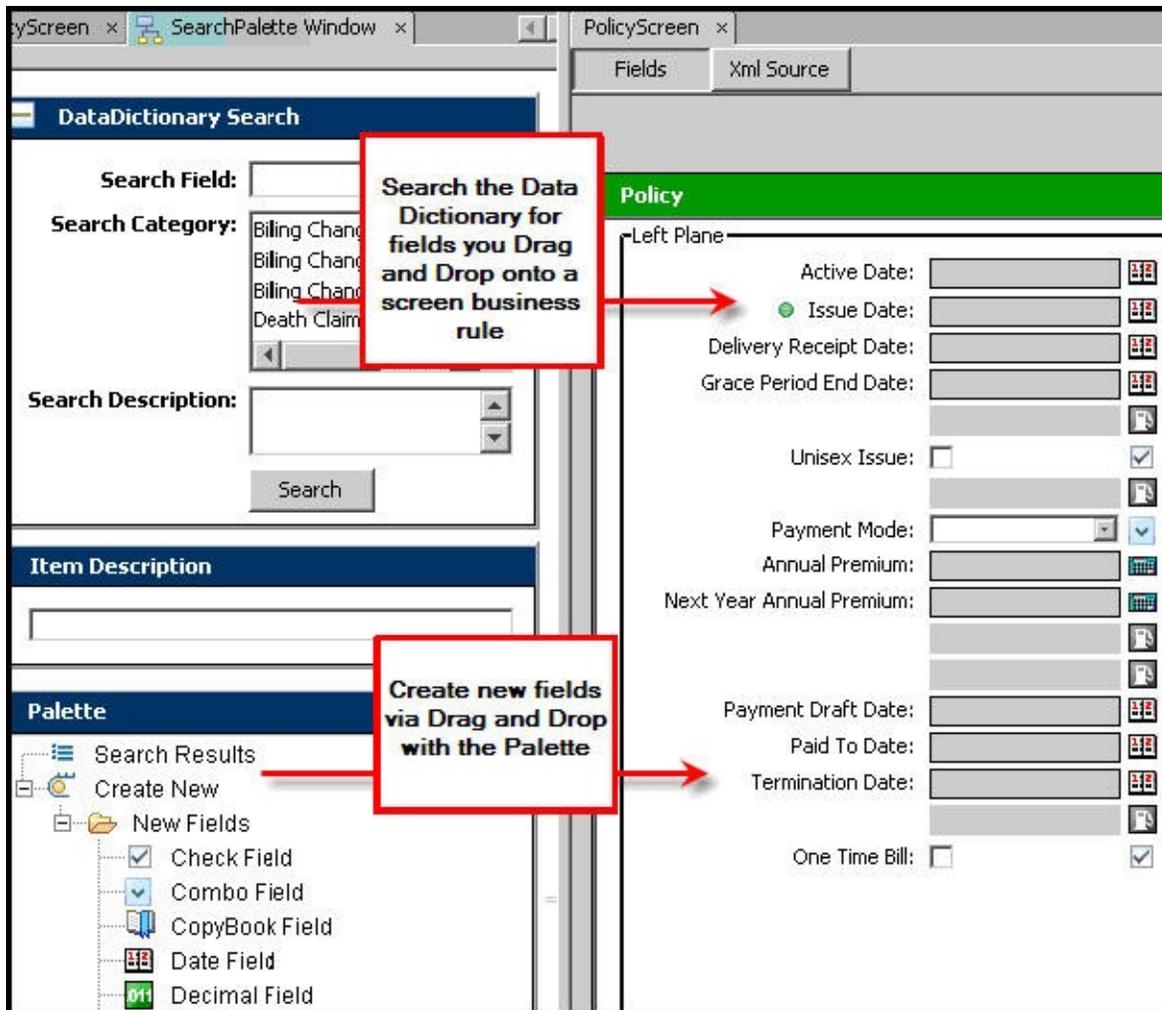
1. [Allocation Screen](#): Defines the allocations that are assigned for the plan and policy. This is the default method for configuring allocations.
2. [EligibleTransactionByPolicyStatus](#): Controls the transactions that are available to the OIPA user depending on the [status](#) of the policy and whether in that status the transaction will be initiated by the user or by the system.
3. [Policy Screen](#): Defines the foundation of the contract, such as the issue date, premium mode, and free look end date.
4. [Plan Screen](#): Defines the plan structure, such as maximum face amount, maximum age or grace to lapse days.
5. [PolicyValues](#): Used to display valuation on a policy and allows the

user to configure other fields to run at the time valuation is executed in a transaction

6. [Segment Screen](#): Defines the display of the columns in the segment summary, which displays above the segment detail information. This is also where the ability to add new segments can be disabled, unless the segment is added by a transaction.
7. [Values Screen](#): Defines the formatting of non-fixed fields on the Values screen.

When configuring any of these screen business rules, a comprehensive visual editor for field entry is available in the Rules Palette. The visual editor provides drag and drop functionality and allows the user to see the screen while editing. However, if XML is preferred, configuration can still be done in XML via the [XML Source pane](#). Configuration for the additional screen tags must still be done through the XML Source pane.

The [DataDictionary](#) or the Palette, which are located in the [SearchPalette window](#), can be used to drag and drop items into the Configuration Area.



SearchPalette Window

You are here: Main Explorer > Plans > Add New Plan

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Add a New Plan

For individual Products, new plans are added to companies through the Main Explorer tab. Once a new plan is added, [security](#) must be assigned to it or it will not be visible in OIPA. When a plan is created, the plan information is written to the AsPlan table. Plan information can be [edited](#) by right-clicking on the plan name and selecting the **Edit Plan** option. This will open the Plan Maintenance window.

For group Products, new plans are added in the application itself.

A plan cannot be deleted once it has been created.

Plan Rules

Some business rules are automatically created and attached to a new plan. In OIPA, the Plan Screen, Segment Screen, Policy Screen, Policy Values Screen, Values Screen, EligibleTransactionsByPolicyStatus and PolicyAllocation Screen business rules are automatically created and can be found in the [Plan Rules](#) folder under the new plan. Any other business rules needed by the plan must be added by [creating rule overrides](#) from the Global Rules Explorer folder.

Steps to Create a New Plan

1. Right-click on the plan folder where the new plan should be added.
2. Select **New Plan**. The New Plan wizard will appear.
3. Enter the plan information.
 - a. **Company Name**: This field is disabled and is populated with the Company name associated with the plan.
 - b. **Product**: The Product field will only be visible if Products are turned on in the Web Application Utility during installation. When Products are on, this field is pre-populated with the Product name and disabled as the new plan must be created under a Product.
 - c. **New Plan Name**: This is a required field. Enter the name of the plan. The plan name may not be the same name within a company, but the plan name may be the same across primary and subsidiary companies.
 - d. **Currency Code**: Select the default currency for monies applied to this plan. If the currency code is not listed, go to **Admin Explorer** and open the **Currency** folder. Add the new currency code to the Currency Editor. Refer to the [Currency section](#) for detailed steps on adding a new currency code.
 - e. **Market Maker**: Select the **Market Maker** you plan to use for any currency conversions. If the Market Maker is not listed, go to **Admin Explorer** and expand the **Market Maker** folder. Open the Market Maker Editor and add the new Market Maker. Refer to the [Market Maker section](#) for more details.
 - f. **Plan Effective Date**: This field is pre-populated with the current system date. The date can be changed, but must be earlier chronologically than the Plan Expiration Date.
 - g. **Plan Expiration Date**: Select the date when this plan will expire. If the date entered in this field is earlier than the Plan Effective Date, an error will occur.
 - h. **Point In Time Valuation**: Only two options can be selected for a new plan that does not yet have policies associated with it. Select **Yes** for Point in Time valuation or **No** for Traditional

valuation. The **Transitioned** option is not available as there are no existing policies to transition. Once policies are added to this plan, the valuation type cannot be changed.

- i. **Mixed Valuation**: This field will be disabled. It is only enabled for existing plans if Point In Time Valuation is set to Transitioned.
4. Select **Finish** when all of the plan information has been entered. The Finish button will only become enabled when all of the required plan information is entered.

The screenshot shows a 'New Plan' wizard window. On the left, a 'Steps' pane indicates the current step is '1. New Plan'. The main area contains the following fields:

- Company: Test Company
- New Plan Name: New Plan
- Currency Code: USD
- Market Maker: USA
- Plan Effective Date: Thu 10/06/2011
- Plan Expiration Date: (empty)
- Point In Time Valuation: Yes
- Mixed Valuation: (empty)

A red box highlights the 'Point In Time Valuation' field, and a red arrow points from a text box labeled 'Determine type of valuation' to this field.

New Plan Wizard with Valuation Highlighted and Product Functionality Off

Agreement Plans

Plans can also be created at the agreement level by using the Agreement screen in OIPA. If the user has the proper security settings enabled, an **Add** button will display above the Plan Agreement Details section of the screen, which will allow for the creation of agreement-level plans. Once saved, these plans will display in the Rules Palette.

You are here: Main Explorer > Plans > Edit Plan Properties

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Edit Plan Properties

After a plan has been created the plan properties can be edited. The plan name and company name are the only fields that cannot be edited. The steps necessary to edit the plan properties are shown below.

Steps to Edit Plan Properties

1. From the Main Explorer window, right-click on the plan name and select **Edit Plan**. The Plan Maintenance window will appear.
2. Make the necessary changes to the plan properties by clicking on the drop down box next to each property.
 - **Company**: this field cannot be edited.
 - **Plan Name**: this field cannot be edited.
 - **Currency Code**: Specifies the default currency for monies applied to this plan. If the currency code is not listed, go to **Admin Explorer** and open the **Currency** folder, and add the new currency code to the Currency Editor. Refer to the [Currency section](#) for detailed steps on adding a new currency code.
 - **Market Maker**: Specifies the [market maker](#) to be used for any currency conversion for this plan. If the Market Maker is not listed, go to **Admin Explorer** and expand the **Market Maker** folder. Open the Market Maker Editor and add the new Market Maker. Refer to the [Market Maker section](#) for more details.
 - **Effective Date**: Specifies the date the of the plan's inception. This is a required field.
 - **Expiration Date**: Specifies the plan's expiration date. If this date is earlier than the Effective Date, then an error will occur.
 - **Point In Time Valuation**: This field indicates whether the plan uses Point in Time valuation or Traditional valuation. It updates the PointInTimeValuation column in the AsPlan database table. The options are as follows:
 - If Point In Time Valuation is blank, a choice must be entered.
 - **Yes** can only be selected if no policies exist yet on the plan. If policies are already associated with the plan, then **Transitioned** or **No** must be selected.
 - **No** continues to use traditional valuation.
 - **Transitioned** indicates a conversion from Traditional to Point in Time valuation.



If Point in Time Valuation is changed to Transitioned, the plan must be transitioned to Point in Time valuation. Contact Oracle support for help converting to Point in Time valuation.

- **Mixed Valuation:** This field is enabled only if Point In Time Valuation is set to Transitioned. If enabled, the choices are:
 - **Yes:** Use mixed valuation methods. This setting indicates that both of the valuation methods will be used throughout the life of the policy. The transition date will determine how valuation should be written.
 - **No:** Use only Point in Time valuation. Any activity that is processed on the policy will write Point in Time valuation and no transition date will be required. This includes activities that are reprocessed, which previously calculated valuation using Period of Time.



A plan cannot switch back and forth between valuation types. Once a plan is identified with Point in Time (Transitioned or Yes) it cannot be changed back to Traditional valuation.

3. Select **Finish**.

Plan Maintenance

Steps	Plan Maintenance
1. Plan Maintenance	
	Company: <input type="text" value="Prototype Child Company"/>
	Plan Name: <input type="text" value="Functional Prototype Plan"/>
	Currency Code: <input type="text" value="USD"/>
	Market Maker: <input type="text" value="USA"/>
	Effective Date: <input type="text" value="Sat 01/01/2000"/>
	Expiration Date: <input type="text"/>
	Point In Time Valuation: <input type="text" value="Yes"/>
	Mixed Valuation: <input type="text"/>

Window to Edit Plan Properties

You are here: Main Explorer > Plans > Plan Data Values

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Plan Data Values

Each plan has a Plan Data folder, which stores constant values for a plan that are not based on any criteria. The values included in the Plan Data folder are determined solely by the PlanGUID and do not require any additional criteria. If a second criterion is needed, such as state code, then [map groups](#) should be used to store the value.

Plan Data has three panes: **Plan Fields**, **Withholding** and **Plan Allocations**. The fields that display on each pane can also be edited.

Plan Data has three panes: **Plan Fields**, **Withholding** and **Plan Allocations**. The fields that display on each pane can also be edited.

- **Plan Fields:** Field data is extracted from the [PlanScreen business rule](#) and the values are stored in AsPlanField. Plan fields data represents the information that is generic to the plan, such as limits, factors, fees, etc. This information is then available to rule and transaction configuration via math or calls to the database via SQL statements. Refer to [Plan Data Fields](#) for more information on updating field information.
- **Withholding:** The Withholding field data is extracted from the WithholdingScreen business rule and the values are stored in AsWithholdingField. The Withholding information represents the default Plan level withholding that will be applied to disbursements if policy or activity withholding is not defined.
- **Plan Allocation:** Allocation data is extracted from the AllocationScreen business rule and the allocation values are saved in AsAllocation. The allocation information represents the default plan level allocations that will occur when plan allocations are defined via configuration. There are two options in Allocation Default Method and Model Supported Method. System Codes have either OIPA or NBU or both. (This determines what all the functionality we want to use for this plan.

- **Default method**: Field data is extracted from the AllocationScreen business rule and the allocation values are saved in AsAllocation. The allocation information represents the default plan level allocations that will occur when plan allocations are chosen through configuration. Proper setup of a plan level allocation is required in the AllocationScreen business rule. Then from the Plan Allocation pane, select the Expand All button to see the Allocation. Select on the allocation name under the Allocations node. Enter an Effective date and select the **Add Group** button. Add the funds under the newly created group using the **Add Fund** button. Fund(s) and percentage(s) can be defined there.
- **Model Supported method**: The contents of the Plan Allocation tab for models does not come from the AllocationScreen rule, but rather is hard coded. An action bar and Editor pane allow a user to enter allocation values.
 - The Action Bar has six action buttons: Expand All, Collapse All, Add Fund, Remove Fund, Add Model and Remove Model.
 - The Main Editor pane contains two sub-panes: Allocation Nodes (tree structure) and Fund / Model selection. The Add Fund or Add Model buttons can be used to add funds or models to allocations. One, but not both should be used. The Fund/Model sub pane is used to define percentages. A user can click on the Add Allocation button, and the allocation's name and percentage information will appear in the newly generated Fund or Model sub-node under the Allocation node.

Fields defined as <DataType>Money</DataType> in the PlanScreen, WithholdingScreen and AllocationScreen business rules display a currency field for entry in the Rules Palette PlanData node. Truncation and rounding methods are defined in [AsCurrency](#).



Plan Data should not be changed once in production unless an error was made in the entry of the information or unless changing

the information will not affect prior policy activity calculations. For example, a minimum value could be reduced, but not a factor for a premium deduction charge. Any values that could change in the future and might affect calculations should be stored using the [map group](#) tables.

The screenshot displays a software interface with three tabs at the top: "Plan Fields", "Withholding", and "Plan Allocation". The "Plan Fields" tab is selected. Below the tabs, the "Plan" field contains the text "Guaranteed Level Premium Term". A red box highlights the "Plan Fields" tab and the "Plan" field, with a red arrow pointing to the "Plan Fields" tab and another red arrow pointing to the "Plan" field. A red box also highlights the "Plan Fields" tab and the "Plan" field, with a red arrow pointing to the "Plan Fields" tab and another red arrow pointing to the "Plan" field. Below the "Plan" field, there is a section titled "Plan Fields" with a blue header. This section contains several fields with values: "Annual Modal Factor: 1.0", "Base Rate Description: Premium", "Base Stop Notice Days: 90", "Billing Lead Days: 28", "Conversion Stop Notice Days: 90", and "Conversion Age Reference: 75".

Annual Modal Factor:	1.0
Base Rate Description:	Premium
Base Stop Notice Days:	90
Billing Lead Days:	28
Conversion Stop Notice Days:	90
Conversion Age Reference:	75

Plan Data Tabs

Security

OIPA users can view plan data if security privileges are granted to the user. Read-only access is the only type of access granted. Plan data cannot be changed in OIPA.

- To grant a user security access to view Plan Fields in OIPA, open **Admin Explorer | Security | Application Security | Security Groups** and double-click the security group associated with the user. Then open **Company Security | Company Pages** and check out the PlanFields page. Check the box for Plan Fields then check in the file.
- To grant a user security access to view Plan Withholding in OIPA, open **Admin Explorer | Security | Application Security | Security Groups** and double-click the security group associated with the user. Then open **Company Security | Company Pages** and check out the PlanWithholding page. Check the box for Plan Withholding then check in the file.
- To grant a user security access to view Plan Allocations in OIPA, open **Admin Explorer | Security | Application Security | Security Groups** and double-click the security group associated with the user. Then open **Company Security | Company Pages** and check out the PlanAllocations page. Check the box for Plan Allocations then check in the file.

Steps to Update Plan Data Values

1. Expand the Plan folder.
2. Expand the Plan Data folder.
3. Select and right-click on Plan Data.
4. Select **Check-out**.
5. In the [Configuration Area](#), select the pane (Plan Fields, Withholding or Plan Allocation) that stores the value to be updated.
6. Enter the new value in the appropriate field. Refer to the descriptions of the three panes provided above if additional explanation is needed.
7. Select and right-click on the plan data XML file.
8. Select **Check-in** to update the database with the changes.

You are here: Main Explorer > Plans > Plan Data Fields

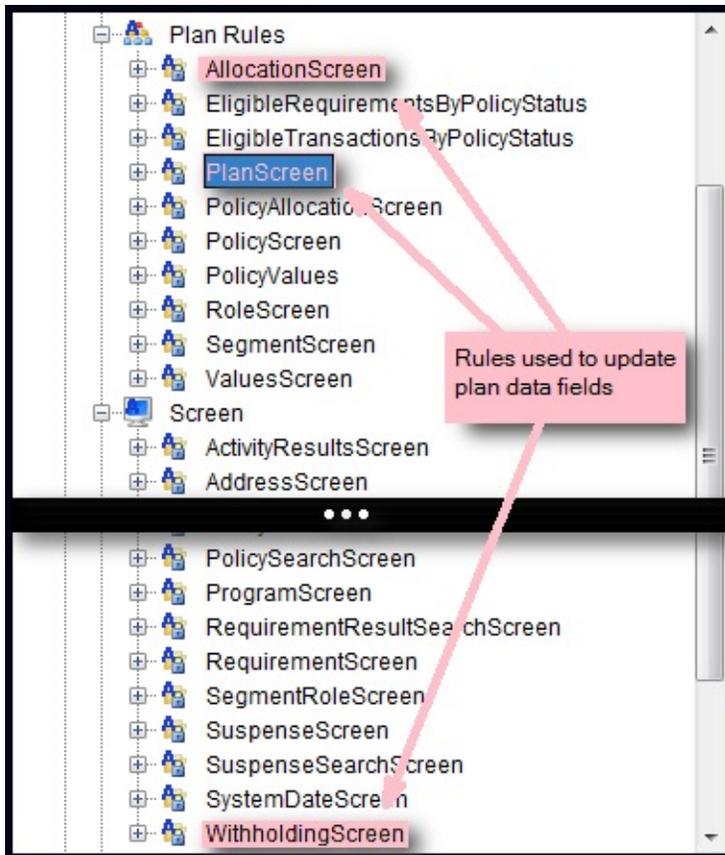
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Plan Data Fields

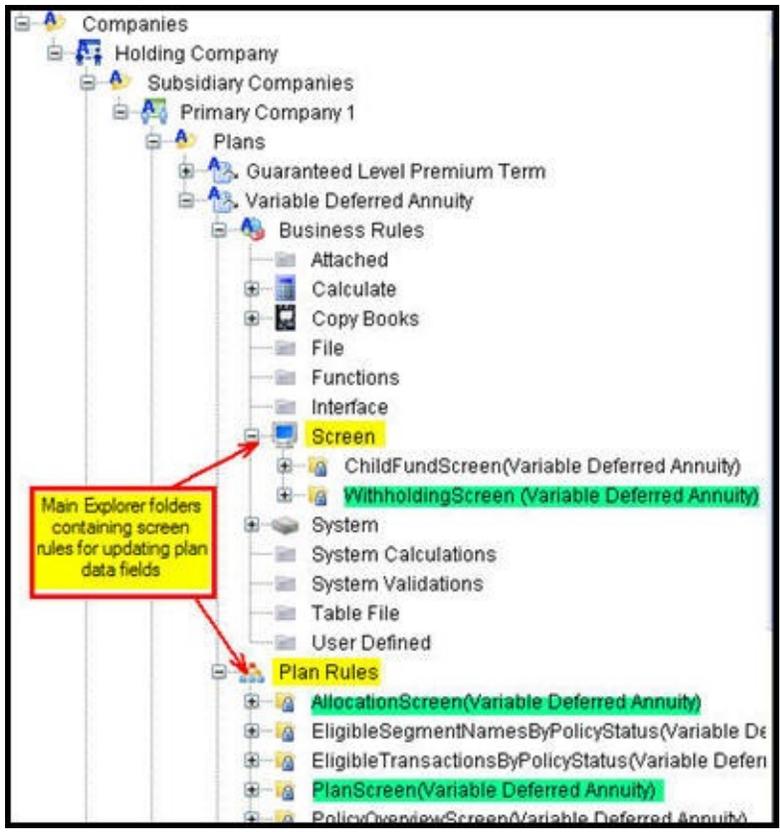
Plan Fields and Plan Withholding are defined via a two step process in the Rules Palette. Fields are first defined in the PlanScreen and WithholdingScreen rules respectively. Field values are then entered via the Palette Plan Data editor.

PlanAllocations are set in the AllocationScreen if allocations are [configured using the default method](#) or the PolicyAllocationScreen is allocations are [configured using models](#).

The Withholding screen business rule is located initially only in the Global Rules Explorer in the **Business Rules | Screen** folder. This is because it is not a plan rule so it is not automatically created when a new plan is created. A plan level override of the Withholding screen can be created by going into the Global Rules Explorer, right-clicking on Withholding Screen and selecting **New Withholding Screen Override**. Once a plan level override is created, locate the Withholding screen in the Main Explorer in the **Company | Plans | Plan | Business Rules | Screen** folder.



Global Rules Explorer Folder Structure



Main Explorer Folder Structure

Steps to Update Plan Data Fields

1. There are two rules identified with a rules category of PlanRules that are used to create plan definition. These are [PolicyAllocationScreen](#) (used to [define allocation models](#)) and [PlanScreen](#). If the [default method of allocation configuration](#) is being used, then the AllocationScreen rule should be configured.
Expand the Plan Rules folder and select the desired role

2. Expand the **Plan Rules** folder and select the appropriate rule. If using the AllocationScreen rule, expand the **Screen Rules** folder to locate the rule.



If the Withholding screen is the only rule that needs to be updated, skip to Step 6.

3. Right-click on the rule to update and select **Check-out**.
4. Use the visual configuration tools to update either the PlanScreen or AllocationScreen.
5. Right-click on the plan rule XML file and select **Check-in**. This will save the changes to the database.
6. Expand the **Screen** folder and select **WithholdingScreen**. If using Main Explorer, open the appropriate plan folder first and then locate the **Business Rules | Screen** folder.
7. Click the **XML Source** tab and make the updates.
8. Right-click on the plan rule XML file and select **Check-in**. This will save the changes to the database.

The Plan Data editor will show the fields configured in these business rules in the corresponding tabs.

Plan Fields Withholding Plan Allocation

Plan: Guaranteed Level Premium Term

Plan Fields Tabs to enter plan data values

Annual Modal Factor:	1.0
Base Rate Description:	Premium
Base Stop Notice Days:	90
Billing Lead Days:	28
Conversion Stop Notice Days:	90
Conversion Age Reference:	75

Plan Data Tabs

Inherited Plan Data



See the [Product-Plan Data Inheritance](#) page for detailed information on the inheritance of data within the Product/Child Product/Plan structure.

Field configuration that is inherited from the Product or Child Product to which the plan belongs will display on the Plan Fields tab grouped by override level. If the user hovers the mouse over the field name, a tooltip will display providing the name of the Product or Child Product from which the field is inherited.

Each inherited field will display in grey to signify that the field is inherited, and is therefore disabled for editing. However, the field configuration may still be able to be overridden—a visual indicator * **Final** will display next to any field that is unable to be overridden at the current override level.

To disable a plan field from being overridden at a lower level, check the **Final** checkbox in the [Field Properties](#) window.

You are here: Main Explorer > Inquiry > Inquiry Screen Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Inquiry Screen Overview

The Inquiry screen enables users to access information via a rules defined format. Inquiry screens can pull various data from different tables in the OIPA database that are relevant to the type of inquiry a business requires. Using inputs, outputs and various layout features, an Inquiry screen can be configured to display pertinent information.

The rule can be created at the Main Menu, Client or Policy level. The level determines where the Inquiry screen will be accessed in OIPA. It also determines which GUID is passed to the Inquiry screen rule.

- PrimaryCompanyGUID for Main Menu and Client
- PolicyGUID for Policy

The fields defined within the Inquiry screen XML determine the type of information the Inquiry will return. View the **XML Configuration Guide** option from the Help menu. Inquiry screen configuration is located in **Business Rules | Screen Rules | Inquiry Screen**.

Security for the Inquiry Screen

Only OIPA users with the proper security permissions will be able to view Inquiry screens in OIPA.

The Main menu level Inquiry screen contains a security permission in the [Company Pages](#) folder. The permission is called [Inquiry](#). The page must be checked out and the box checked.

The Client level Inquiry screen contains a security permission in the [Company Pages](#) folder. The permission is inside the [Client](#) page. The page must be checked out and the Inquiry box checked.

The Policy level Inquiry screens are granted permissions in the [Plan Pages](#) folder. There are several permissions that control Inquiry privileges. Click each link below to view the security information.

- [Policy](#)
- [PolicyOverview](#)
- [PolicyActivity](#)
- [PolicyAllocation](#)
- [PolicyRole](#)
- [PolicySegment](#)
- [PolicyValue](#)
- [Programs](#)
- [AllocationHistory](#)
- [RoleHistory](#)
- [SegmentRole](#)
- [SegmentRoleHistory](#)
- [PolicyWithholding](#)

You are here: Main Explorer > Inquiry > Inquiry Screen Configuration

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Inquiry Screen Configuration

Inquiries can be configured to display in OIPA in three places: the Main Menu, the Client screen or Policy related screens. The type of inquiry is identified when a new Inquiry screen is created.

Inquiry screens configured to display on the Main Menu will appear in OIPA under the Inquiry heading on the Main Menu.



Main Menu Inquiry Screen in OIPA

Inquiry screens configured to display at the Client level will appear in OIPA on the Client screen. The Inquiry link will appear on the Secondary Menu.



Client Level Inquiry in OIPA

Policy level inquiries have undergone significant changes. Their contextual properties have been revised to include six new override levels, which replace prior versions' Primary Company level. The new override levels are Subsidiary Company, Product, Plan, Subsidiary Company and State, Product and State, and Plan and State. Policy level inquiries do not have Global versions. Inquiry screens configured to display at the Policy level will continue to appear in OIPA on the Policy

screen. The Inquiry link will appear on the Secondary Menu.



Policy Level Inquiry Screen in OIPA

Steps to Create an Inquiry Screen

1. Open the **Main Explorer** tab.
2. Open the appropriate company folder.
3. Right-click on **Inquiry Screens** and select **Add Inquiry**.
4. Enter a name for the Inquiry. It should reflect the purpose of the Inquiry so that OIPA users can identify its purpose.
5. Select an inquiry type from the drop down box. This will determine the screen where the Inquiry screen will be available to OIPA users. If Main or Client level inquiry is selected, then skip to step 8. If Policy level inquiry is selected, then continue with the following steps.
6. Click **Next**.
7. Select the override levels. When a new override of an existing inquiry is created, the Override Wizard will check if selected override level (Company, Product, Plan, etc...) already exists. If it does, a validation message will appear, indicating that the override already exists.
8. Click **Finish**. The new Inquiry screen will be listed in the appropriate level folder under the Inquiry screen node.

Once an Inquiry screen has been created, check it out and configure it. View the **XML Configuration Guide** topic in the Help menu. Inquiry screen configuration is located in **Business Rules | Screen Rules | Inquiry Screen**.

Make sure to also set the proper [security settings](#) to allow OIPA users to view the Inquiry.

Valuation Date on Inquiry Screen

To make the Valuation Date available on Inquiry screens, the PolicyValues rule needs to be configured with the following math variable:

```
<MathVariable VARIABLENAME="ValuationDate" TYPE="F:  
    DATATYPE="DATE">Valuation:Policy:EffectiveDate
```

Functions and CopyBooks

CopyBooks and Functions configured within Main Menu and Client level inquiries are resolved at the Primary Company context. CopyBooks and functions configured within policy level inquiries are resolved at the Policy context.

Currency Representation on Inquiry Screen

A currency type can be displayed on dynamic fields on the Inquiry screen in both the Input and Output sections.

In the Input section, this is accomplished by using the “Money” <DataType> element value, currency codes (THD, USD, etc...) for the <Currency> element, and a single currency code for the <DefaultCurrency> element.

In the Output section, this is accomplished by using the “ToCurrency” function math variables that are passed to dynamic fields with a “Money” <DataType>.

Inquiry screen has a new attribute configured as "Total" for the output column specification wherein if the Column Data is numeric, a total of the results would display. The total that referred is the entire results set, that is added to the Column to fetch the value results for the numeric column data at the bottom of the grid via configured SQL statement.

You are here: Main Explorer > Inquiry > Inquiry Screen Overrides

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Inquiry Screen Overrides

Inquiry screens can be configured to display on the Main menu, the Client screen and Policy screens. Main menu and Client screen inquiries can be overridden at the Primary company level. The Primary company's GUID is stored in the AsInquiryScreen table's CompanyGUID column. Both of these types of inquiries do not have global versions.

Policy inquiry overrides have much more flexibility. Like Main menu and Client inquiries, there are no global versions. There are; however, six override levels. These levels are listed below from least specific to most specific.

- Subsidiary Company
- Product
- Plan
- Subsidiary Company and State
- Product and State
- Plan and State

Subsidiary Company Overrides

While on policy-type screens, users will be able to access all inquiries overridden by the policy's subsidiary company. If a more specific override exists with the same name, which also belongs to the same subsidiary company, then that override will display instead. No inquiries overridden by other subsidiary companies will be accessible.

Product Overrides: While on policy-type screens, users will be able to access all inquiries overridden by the policy's Product. If a more specific override exists with the same name, which also belongs to the same Product, then that override will display instead. No inquiries overridden by other Products will be accessible.

Plan Overrides: While on policy-type screens, users will be able to access all inquiries overridden by the policy's plan. If a more specific override exists with the same name, which also belongs to the same

plan, then that override will display instead. No inquiries overridden by other plans will be accessible.

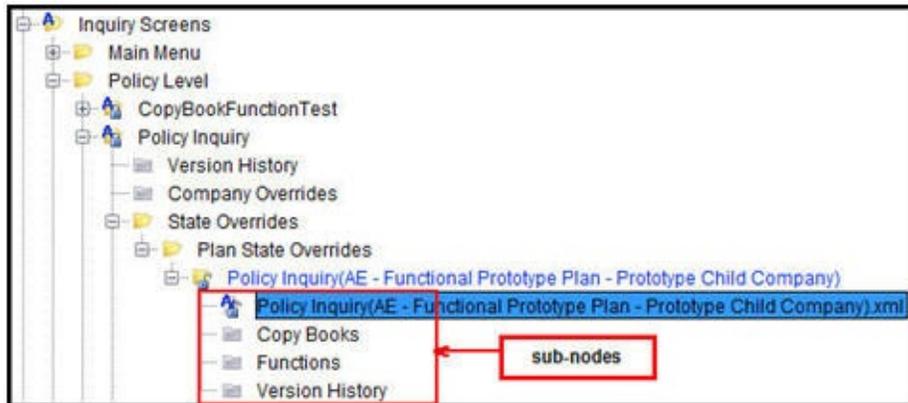
Subsidiary Company and State Overrides: these are overrides at the subsidiary company level that are available in one specific state only. If the state designated in the policy's "Issue State" fixed field is the same as the inquiry's overriding state, then this version of the inquiry will be available. All other "non-same-name" subsidiary company overridden inquiries will also be available.

Product and State Overrides: these are overrides at the Product level that are available in one specific state only. If the state designated in the policy's "Issue State" fixed field is the same as inquiry's overriding state, then this version of the inquiry will be available. All other "non-same-name" Product overridden inquiries will also be available.

Plan and State Overrides: these are overrides at the plan level that are available in one specific state only. So if the state designated in policy's "Issue State" fixed field is the same as the inquiry's overriding state, then this version of the inquiry will be available. All other "non-same-name" plan overridden inquiries will also be available.

Policy Level Navigation Hierarchy

The following new override sub-nodes are available under the Policy level node: Company, Product, Plan and State. Each of the override sub-nodes will contain their own set of sub-nodes (similar to those for business rules): XML file, CopyBooks, Functions and Version History (for those environments that have IVS).



Sub-Nodes for Policy Level Inquiry in Main Explorer

You are here: Global Explorer and Business Rules > Global Rules Explorer Tab

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Global Rules Explorer Tab

The **Global Rules Explorer** tab is one of the two tabs that can be used to log into a Rules Palette environment. When logged in, the Global Rules Explorer organizes information by category and not by company, as the Main Explorer does. Users can see all [overrides of rules](#), [CopyBooks](#) and functions contained for the various rules. Overrides for rules include company, plan, transaction, fund and state level overrides.

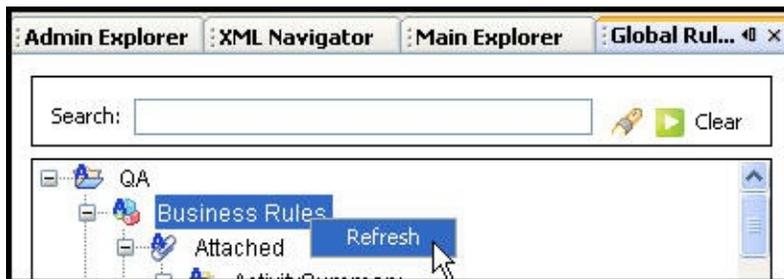
Business Rules and Plan Rules

The Global Rules Explorer tab provides the option to create new rules and rule overrides. This is the only tab that gives users the option to perform these tasks.

Transactions and segments cannot be created in this tab. The Main Explorer tab must be used to create them.

To view a rule, just expand  the rules folder and double-click on its XML file. The configuration will display in the [Configuration Area](#).

The Business Rules, Transactions and Segment folders all have a right-click **Refresh** option. To refresh, select one of the three main rule folders, then click the Refresh button. Any recently added or revised rules that other configurators have been working on will display.

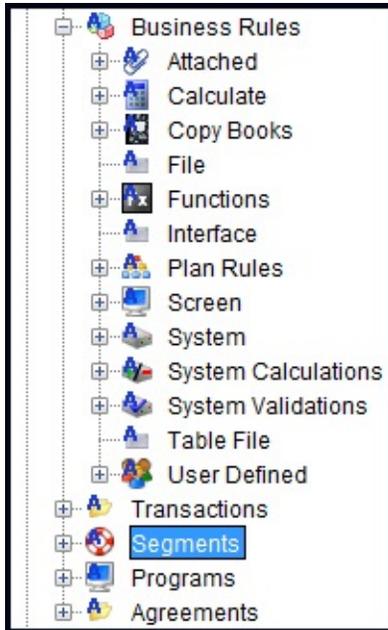


Business Rules Refresh Option

Global Rules Folders

- **Attached** folder stores all the rules that may be attached to a transaction. The rules in the attached folder provide the ability to configure additional processing options.
- **Calculate** folder stores all the rules that perform calculations during segment processing.
- **Copy Books** folder stores all CopyBook rules (can be global or plan-specific).
- **File** folder stores all files configured to select, transform and upload incoming data sent from an external application
- **Functions** folder stores rules that apply specific calculations and guidelines to various plans.
- **Interface** folder stores all rules that are used to interface XML data to different systems.
- **Plan Rules** folder stores all rules that are required in order to configure any new plan.
- **Screen** folder stores rules that are associated with a screen in the application.
- **System** folder stores rules that must be present in order for the system to process correctly.
- **System Calculations** folder stores rules with pre-defined names that are used to perform set calculations. The syntax required is defined by the rule name. These rules should only be used for backward compatibility.
- **System Validations** stores rules with pre-defined names that are used to perform set validations. The syntax required is defined by the rule name. These rules should only be used for backward compatibility.
- **Table File** folder stores all rules that direct the system to a table that stores required data.
- **User Defined** folder stores optional rules.
- **Transaction** folder stores all configured transaction rules.

- **Segments** folder stores all configured segment rules.
- **Programs** folder stores all configured programs
- **Agreements** folder stores all configured agreements.



Global Rules Explorer Folders



If all the rule folders are empty, then there is a possibility that the XML schemas are null in the Business Rules table. In this case, upload schemas using the **Upload Schema** function on the toolbar. Refer to the [Upload Schemas](#) section for more information.

You are here: Global Explorer and Business Rules > Attached Rules > Attached Rules Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Overview of Attached Rules

Attached rules are rules that are attached to transactions or requirements for supplemental processing. On the Main Explorer they are listed under the transaction's or requirement's Attached Rules node, and, for transactions, are also added to the TransactionBusinessRulePacket.



The XML Configuration Guide contains a detailed explanation of the elements, attributes and values used to configure these business rules. Select **Help** from the Main Menu and open the XML Configuration Guide. Attached Rules are located in the **Business Rules | Attached Rules** folder.

Attached Rules

Activity Summary: The ActivitySummary business rule drives the Activity Summary screen. This rule allows for the configuration of the summary screen and summary groups including activity fields and math variables, allocation changes and activity results.

This rule should be attached to a transaction but not placed in the TransactionBusinessRulePacket.

Add Agreement Roles: This business rule can be attached to a plan or client level activity to add existing clients in the database to new agreement roles on an existing agreement. The rule consists of a test condition, which when true will add the agreement role and copy back the agreement role detail information established during the Add Agreement Role activity. This rule can only add roles to the agreement the activity is processed against.

AddImpairments: This business rule is attached to a transaction in order to add Impairments based on the configured Impairment criteria.

AddRequirements: This business rule is attached to a transaction or requirement in order to add requirements based on the configured requirement criteria.

Add Roles: This business rule adds existing clients in the database to new roles on an existing policy or segment. This rule can only be attached to a policy-level transaction and can only be used to add roles to the policy on which the activity is being processed. [View Add Roles prototype](#).

[ConfirmationScreen](#) : This business rule should be attached to a transaction if a Confirmation screen is required. It should NOT be placed in the TransactionBusinessRulePacket. The rule adds a Confirmation screen to the activity it is attached to and displays the screen immediately after all validations have been processed, the activity has been saved, and the user clicks the OK button on the Activity Details screen, or on the

Verification screen.

The screen provides custom information to the user. Title, message and field sections are available for configuration. The title is displayed as the heading for the Confirmation screen. If a title is not configured, then the policy number and transaction name will display. Messages can be static or dynamic. Dynamic messages will include substitution fields only from the main body of the transaction. Only static messages will be translated. Fields must be configured within the main body of the transaction to be available for display on the Confirmation screen. Multi-currency support is available for fields.

A confirmation number can also be configured for tracking purposes. The originating activity's confirmation number should be configured using an [Identifier](#) field and an [Identifier math variable](#) should be used to generate confirmation numbers for spawned activities.

Refer to the Confirmation screen prototype for a detailed explanation of how to configure this screen.

CopyToAddressFields: This business rule allows one or more math variables to be copied from an activity to one or more address fields, upon processing the activity with this rule attached.

CopyToAgreementFields: This business rule is used to copy one or more activity values to one or more agreement fields. A MathVariable or a field name can be used to place a single value into an agreement field and a collection can be used to place multiple values onto multiple agreements. The values that are copied to agreement fields will be written to the corresponding columns of the AsAgreement or AsAgreementField database tables.

In addition to field values, CopyToAgreementFields will automatically update the OptionText of combo box and radio button fields.

This rule must be listed in the TransactionBusinessRulePacket.

CopyToAgreementRoleFields: This business rule is used to copy one or more activity values to one or more agreement role fields. A MathVariable or a field name can be used to place a single value into an agreement role field and a collection can be used to place multiple values onto multiple agreement roles. The values that are copied to agreement role fields will be written to the corresponding columns of the AsAgreementRole or AsAgreementRoleFields database tables.

In addition to field values, CopyToAgreementRoleFields will automatically update the OptionText of combo box and radio button fields.

This rule must be listed in the TransactionBusinessRulePacket.

CopyToClassFields: This business rule is used to copy values from one or more activity fields to one or more class fields. The values that are copied to class fields will be written to the corresponding columns of the AsClass or AsClassFields database tables.

When this business rule is attached to a transaction, it will also need to be added to the corresponding TransactionBusinessRulePacket business rule.

CopyToClassGroupFields: This business rule is used to copy values from one or more activity fields to one or more class group fields. The values that are copied to class group fields will be written to the corresponding columns of the AsClassGroup or AsClassGroupField database table.

When this business rule is attached to a transaction, it will also need to be added to the corresponding TransactionBusinessRulePacket business rule.

CopyToClientAltIdFields: This business rule is used to copy values from one or more activity fields to one or more client alternate ID fields. The values that are copied to client alternate ID fields will be written to the corresponding columns of the AsClientAltId database table.

When this business rule is attached to a transaction, it will also need to be added to the corresponding TransactionBusinessRulePacket business rule.

CopyToClientFields: This business rule is attached to a transaction or requirement to allow one or more math variables to be copied from an activity to one or more client fields upon processing the activity to which the rule is attached.

CopyToClientRelationshipFields: This business rule is used to copy values from one or more activity fields to one or more client relationship fields. The values that are copied to client relationship fields will be written to the corresponding columns of the AsClientRelationship or AsClientRelationshipField database tables.

When this business rule is attached to a transaction, it will also need to be added to the corresponding TransactionBusinessRulePacket business rule.

CopyToGroupCustomerFields: This business rule is used to copy values from one or more activity fields to one or more group customer fields. The values that are copied to group customer fields will be written to the corresponding columns of the AsGroupCustomer database table.

When this business rule is attached to a transaction, it will also need to be added to the corresponding TransactionBusinessRulePacket business rule.

CopyToIntakeFileFields: This business rule is used to copy values from one or more activity fields to one or more Data Intake file fields. The values that are copied to Intake File fields will be written to the corresponding columns of the AsIntakeFile or AsIntakeFileField database tables.

When this business rule is attached to a transaction, it will also need to be added to the corresponding TransactionBusinessRulePacket business rule.

CopyToIntakeProfileFields: This business rule is used to copy values from one or more activity fields to one or more Data Intake profile fields. The values that are copied to Intake Profile fields will be written to the corresponding columns of the AsIntakeProfile or AsIntakeProfileField database tables.

When this business rule is attached to a transaction, it will also need to be added to the corresponding TransactionBusinessRulePacket business rule.

CopyToIntakeRecordFields: This business rule is used to copy values from one or more activity fields to one or more Data Intake record fields. The values that are copied to Intake Record fields will be written to the corresponding columns of the AsIntakeRecord or AsIntakeRecordField database tables.

When this business rule is attached to a transaction, it will also need to be added to the corresponding TransactionBusinessRulePacket business rule.

CopyToPendingActivityFields

This business rule can be used to update activity field and withholding field values in policy activities that have not yet become active on the system and have not processed the transaction math. A single activity or withholding field can be updated or multiple fields may be updated.

Only policy activities that are in a pending status are eligible to have field values changed. Pending statuses are defined as activities that have not yet run the Transaction Math section or any attached rules, typically statuses '02' and '09' (Pending and Pending with errors respectively). Plan level activities will not support the undo/redo processing of the CopyToPendingActivityFields rule.

If the updating activity attempts to update the fields of a target activity that is not in a pending status or if the updating activity and target activity are one and the same, a business error will be generated and activity processing will halt.

The <Test> element within <Tests> can be used to set conditions for the update of the activity or withholding fields of a policy. When multiple test conditions are configured to determine if a pending activity or activities' fields should be updated, all conditions must be met in order for the update to be made.

CopyToPolicyFields: This business rule is attached to a transaction or requirement to allow one or more math variables to be copied from an activity to one or more policy fields. If the fields are displayed on the Policy screen, the values will be viewable.

CopyToProgramFields: The CopyToProgramFields business rule is used to update program fields. The update capability of the rule is restricted so that fixed program fields and program status may not be updated. The rule may only be attached to a program transaction. Updates are limited to dynamic disabled program fields.

As a best practice, the ProgramGUID should be referenced in the configuration so that the GUID can be used as an identifier. An example of this is shown below:

```
<MathVariable VARIABLENAME="ProgramGUID" TYPE="EXPRESSION" D,
```

CopyToRequirementFields: This business rule is attached to a transaction or requirement to allow the results of the activity's math calculations to write out to a defined group of RequirementGUIDs.

CopyToRoleFields: This business rule is attached to a transaction and allows one or more math variables or activity fields to be copied to one or more specified RoleFields, upon processing the activity to which the CopyToRoleFields business rule is attached. The rule allows updates to multiple roles. This rule must be listed in TransactionBusinessRulePacket.

CopyToScheduledValuationFields: This rule will create fields in the AsScheduledValuationField. It allows a ScheduledValuation activity to store valuation data from various sources in one convenient location.

This rule, though attached to a transaction, does not appear in the TransactionBusinessRulePacket business rule.

CopyToSegmentFields: This business rule is used to copy one or more activity values to a segment field. A MathVariable or a field name can be used to place a single value into a segment field and a collection can be used to place multiple values on multiple segments. This rule must be listed in the TransactionBusinessRulePacket

CopyToWithholdingFields: This business rule loads the withholding fields from the database and updates them based on the values specified in the business rule. The <From> element identifies the math variable or field where a value is being obtained. The <To> element identifies the field that is being updated. Only one set of withholdings can be updated by a rule.

The <Test> element within <Tests> can be used to set conditions for the update of the Withholding fields of a policy or client. When multiple test conditions are configured to determine if a pending activity or activities' Withholding fields should be updated, all conditions must be met in order for the update to be made.

CreateAdditionalRates: This business rule adds a new rate to AsRate using criteria specified in the plan- or company-level transaction to which it is attached. A Rate Table (a set of Rate Groups with the same name) is specified in the <RateDescription> element of this rule, and the Rate Table must have a consistent set of criteria across all of its data. There may be anywhere from zero to 10 criteria, and only Rate Tables that contain one Rate Group are eligible for the addition of new rates. If the Rate Table does not exist or has multiple Rate Groups, a business error will be generated. Any number of Rate Sets may be added to a Rate Table by using the repeatable <CreateRate> element. The exact rates to be added are specified by a math variable, the name of which should be the value of CreateAdditionalRates' <RateCollection> element.

CreateClientAltId: This business rule is used to copy back values to the alternate name for the purpose of creating a new alternate name

instance. The rule can be attached to a transaction to create an instance of an alternate name for a client. The rule uses the current ClientGUID the user is adding the alternate name to and also uses the From and To format similar to CopyTos to create the values for the alternate name.

CreateClientAddress: This business rule is attached to a transaction to identify the client or array of clients that will receive the address information captured by the transaction. This should only be used with [non-reversible transaction](#) types.

CreateClients: This business rule allows the insertion of new client records from an activity, as well as the creation of relationships, class memberships, addresses and activities related to the new clients being created. Activities created for new clients can be inserted into activity sequencing.

CreateRates: The CreateRates business rule creates a rate group data table record and the rate data table records to go with it. Activity processing is expected to add rates by a forward processing activity. The CreateRates rule applies only in special situations and is not used for valuation rates. This rule is only available to plan level transactions in Primary or Subsidiary companies.

The CreateRates rule must be attached to a transaction and be listed in the TransactionBusinessRulePacket of the transaction.

CreatePolicy: The business rule is introduced to provide the ability to configure the activity-based creation of a new policy based on events occurring on an existing policy. One or more policies may be generated from a single source policy, although only one new policy per activity will be supported. All policies will be created in a pre-issue status pending user review. CreatePolicy business rule will be attached to a non-reversible activity and must be listed in the TransactionBusinessRulePacket.

For Benefit Split, this rule should allow for Benefit fund GUID/amount collection. A target benefit split type code should be specified. The rule should allow a "Copy Source" operation that will copy a specified

segment's activity benefit split based on type code. This feature only applies if the COPYSOURCE attribute of the Segment parent element is set to "Yes". If no source benefit split record exists, then no error is given and no benefit split record will be created for the created policy segment.

CreateSegment: This business rule creates one or more new segments on a policy when attached to a transaction and specific expressed conditions are satisfied. This rule must be listed in the TransactionBusinessRulePacket. [View Create Segment prototype](#).

For Benefit Split, the rule should allow for Benefit fund GUID/amount collection. A target benefit split type code should be specified. Benefit fund GUIDs are specified as this writes directly to AsBenefitSplit and there is no relation logic to look them up from the Parent fund.

CycleProcessBehavior: The CycleProcessBehavior business rule is used to determine how errors are handled during cycle processing. The business rule allows users to stipulate conditions for retrying a transaction that has failed during cycle processing, as well as the number of times the system should automatically retry it. The <Halt> element in the rule's configuration specifies a condition for ending or continuing the cycle, while the <RetryIterations> element specifies the number of retry attempts.

CycleProcessBehavior is available for plan, policy and client transactions. When this rule is present, the system will automatically retry the activity that has returned an error. If the system performs the specified number of retry attempts without success, the transaction will remain pending.

[DeliveryRequirements](#): This business rule defines the fulfillment criteria of the activity's requirements via its date fields. The rule must be attached to a transaction and needs to be in the TransactionBusinessRulePacket of the transaction.

DisbursementUpdate: This business rule is attached to a plan level transaction and will update the status of any pending disbursements identified by the transaction to an active status. The rule allows for a

change in disbursement status from pending to active only. Accounting can optionally be triggered by this rule. Disbursement Accounting is defined by the [Chart of Accounts Entry screen](#). Accounting is invoked during the process of moving a pending disbursement to active. Reversal processing is handled by the policy or client level activity reversal. Reversing the plan level activity will not undo the changes to the disbursement or its accounting.

DoBenefitSplitChange: The DoBenefitSplitChange will allow an activity to redefine/restructure the benefit split records on any given segment to align with a specified parent fund allocation. This rule writes the updates from one segment's benefit split records to AsBenefitSplit. Segment allocation records do not need to be built as a result of this rule. An AssignmentMethod element will indicate the type of action the rule is to perform. These are not assignment types, just general directions of money movement.

DoSegmentRecalculations: This business rule can be attached to a transaction that specifies the names and execution orders of segments attached to a policy. When the rule is processed, the segments for the current policy will be loaded and calculated in the order specified.

For Benefit Split, the Activity GUID must be carried to the benefit split records created from the segment calculate general rules.

ExternalProcess: The ExternalProcess business rule provides the capability of calling custom Java code in order to facilitate greater functionality. A common use may be to generate documents in a manner specific to the document generator being employed. This is an attached rule and must be listed in TransactionBusinessRulePacket. This business rule should be used in conjunction with the GenerateDocument rule.

[FundListForAllocation](#): This business rule is introduced to define controls for the fund dropdown on the Activity add/edit screen. It should not be included in the TransactionBusinessRulePacket.

Generate Document: This business rule describes the document that needs creation upon the execution of a transaction. Multiple output file formats are supported and can be created using a single rule. This rule is not included in the TransactionBusinessRulePacket. This business rule should be used in conjunction with the ExternalProcess rule. [View Generate Document prototype.](#)

[GeneratePendingRequirements:](#) GeneratePendingRequirements defines the comparisons of requirement criteria to math variable values. If all conditions identified in the rule are true, then the requirement is generated. This rule must be used in conjunction with DeliveryRequirements and TransactionBusinessRulePacket.

[GenerateSuspense:](#) This business rule is used to create a suspense record when a transaction with this attached rule is processed.

MaintainSuspense: This business rule allows suspense field values to be changed and accounting generated through a collection of multiple suspense tickets. This rule is attached to a transaction and must be listed in the TransactionBusinessRulePacket.

MatchRequirementResult: This business rule is attached to a requirement to match a specified requirement result with the requirement to which this rule is attached.

PostAssignmentValidateExpressions: This rule processes after the math and assignment processing has completed and can be configured to halt the continued processing of an activity in Pending or NUV Pending status. If configured to do so, error messages can be displayed that can be overridden or canceled by the user. Warning messages, which will not halt processing, can also be configured. If post assignment math is run multiple times in an activity with this attached rule, then the value from the last time run is the one that is logged.

A math variable (from the MathVariables section) or Field substitution may be made in the validation message. The only supported data types for substitutions are 'TEXT', 'INTEGER', 'CURRENCY', 'DATE' or

'DECIMAL'. Therefore any other data types will need to be converted to TEXT, INTEGER or DECIMAL before being passed into the validation message. Special characters such as '\$' or '%' need to be added in the validation message and will not be included in the substitution.



Refer to the [ValidateExpressions](#) description below for additional information on validation message substitutions.

PriceCorrection: This rule will accept fields, multi-value fields or math variables as input. The PriceCorrection business rule must be specified in the TransactionBusinessRulePacket and can only be attached to transactions that are configured within Primary Companies' plans. When attached to a Primary Company level transaction, the PriceCorrection business rule performs the following two tasks:

- Updates records in the AsNetAssetValue database table to reflect the corrected Base, Bare, and/or Offer Unit Value(s). As long as funds are configured within the price correcting activity's Field(s), MathVariable(s) or referenced MultiField business rule, and as long as at least one of the Unit Values for any identified fund is different from the price used in the original activity, then the Unit Value(s) for that fund will be updated accordingly. If the Unit Values for a fund are not different from the original set, then the fund's price will not be updated and the rule will prevent the activity from searching for policies affected by the price correction.
- Searches for policies affected by the price correction. These policies have processed activities with the following criteria:
 - Price Correction Indicator set to "Yes"
 - have been selected as those requiring price correction (i.e. contained within <FundGUID> element of the PriceCorrection business rule)
 - have used the unit value that is being corrected as of the same date as the price correction date (i.e. date contained within <EffectiveDate> element of the PriceCorrection business rule).

Once all affected policies are identified, the rule inserts (spawns) a high Processing Order "dummy" activity into the affected policies. The effective date of the "dummy" activity is the effective date of the earliest

activity (if more than one) that had fund movements, for each of the affected policies. Once processed, the “dummy” activity reverses all subsequent activities, so that they can be re-processed with the corrected set of unit values.



Each plan that has policies within a Primary Company and has unit values capabilities must have a “dummy” transaction configured. Each one of these Policy level “dummy” transactions has to have the same name. If the “dummy” transaction is not configured on an affected policy’s plan, then a system error will be generated when the PriceCorrection activity is processed.

The Price Correction activity that this rule is attached to will be able to update all fund levels, including Benefit Split when the first task listed above is performed. However, if Benefit Split Allocations were calculated using means other than activities (e.g. via CalculateGeneral business rule, invoked from a Segment screen), then the PriceCorrection business rule will not be able to identify the affected policies. Therefore, it is a recommended best practice to use activities for initial and all subsequent Benefit Split Allocation calculations.



Point-in-Time functionality will affect how the PriceCorrection business rule searches for the affected policies.

ProcessActivities: This business rule is attached to a requirement to set conditions for the automatic processing of activities.

ProcessRequirements: This business rule is attached to a transaction in order trigger the processing of requirements. This rule supplements the ability of requirements to schedule execution. Processing a transaction with this rule attached will trigger the processing of requirements that are not necessarily scheduled to be executed at the given moment.

QuoteScreen: This business rule is attached to a transaction to enable display of a Quote button on the Activity Detail screen. The Quote button

allows users to view activity results prior to actually processing an activity. This button is only available for Client and Policy level non-document transactions – including any lower sub-categories of these transaction types (e.g. Policy-Financial-Reversible-Nonreversing, Policy-Financial-Nonreversible-Nonreversing, Client-Financial-Nonreversible-Nonreversing).

Transaction configuration can now define the buttons available to the Activity Detail screen. When the Quote screen is attached to a transaction, the Quote button by default is available. If all buttons available to the screen are not needed, the <Button> element (a sub element of <Buttons>) can be used in transaction XML to define the available buttons. If the button configuration is absent from the transaction configuration, then all buttons will be available by default. For more information on configuring buttons on the Activity Details screen, see the **Buttons Element** page in the XML Configuration Guide—navigate to **Transaction Rules | Transaction Elements | Buttons Element**.

The QuoteScreen rule and the VerificationScreen rule are mutually exclusive. If both the Quote Screen and Verification Screen are attached to a transaction, only the Verify button is available on Activity Detail.

Activities can be added from the Quote screen. An Add New Activity link will display in the Quote window if the following QuoteScreen configuration exists:

- QuoteScreen business rule contains a <Transactions> element with at least one transaction that can be added from the Add New Activity link.
- EligibleTransactionsByPolicyStatus business rule (policy level activities only) contains at least one transaction for the current Policy status, that is also configured within the <Transaction> element of the QuoteScreen business rule.
- User has appropriate OIPA Security role.

ReassignAllocations: ReassignAllocations business rule is used to override existing activity allocations. This is useful when the end user is

not allowed to select allocations or with automated activities where the user is not involved in an on-line experience with the activity. Allocation records are created when this business rule is processed. The Allocation records contain the funds the money is moving into or out of.

To support Benefit Split functionality, this rule can be modified to accept a collection of FundGUID/Percent or Amount value pairs in lieu of the standard SQL in the <To> element. The COLLECTION attribute holds a math variable collection of either FundGUID/Percent or FundGUID/Amount value pairs. ALLOCATIONMETHOD is an attribute that holds a literal value or variable value (i.e. code value) indicating the allocation method to use. Amount, indicated by the code value "02", is the only value allowed.

ReassignBenefitSplit: This rule deactivates the current active benefit split record and activates the future record. The rule writes to AsBenefitSplit. The <From> element specifies the source BenefitSplit type code of the allocation to activate. Generally "51" is the Type Code that is used here. The <To> element specifies the target BenefitSplit type code for the allocation. Generally "05" is the type code used here. The "From/To" elements should be in repeatable groups that allow a single activity to update multiple Segment Benefit Splits. Each group should identify the related GUID for the policy's active benefit split records to be modified.

RolesExist: This business rule is used primarily to validate whether or not the specified role codes exist on the policy, using role code configuration. Apart from that, this business rule is used to verify that certain specified policy or segment role fields are empty (FieldsDoNotExist element) or filled with a value (FieldsExist element) at the time the transaction is processed. [View Roles Exist prototype](#).

ScheduledValuation: This business rule allows the transaction to which it is attached to execute valuation or scheduled computation for all policies returned by a query contained within the rule without adding policy financial activities to those policies. It is best suited for plan level

transactions. The rule also includes switches to include the PolicyValues business rule during the valuation process, write deposit information and what left-over processes from prior attempts may be deleted.

The <Computation> element indicates that scheduled computation should be performed, instead of scheduled valuation. The value of Policy or Segment within the element tells OIPA the type of computation to perform. The RULENAME attribute of this element indicates the rule that should be used to drive the scheduled computation process.

View the [Scheduled Valuation prototype](#) or the [Scheduled Computation prototype](#).

ShadowPendingActivities: This business rule is used to delete an activity from the Activity screen when the specified condition is satisfied. More than one activity can be deleted, but the activity must be in pending status. The activity will appear shadowed once processed. OIPA will give an error if the user tries to delete an activity that is not in pending status.



Only activities of the same type can be deleted. For example, the rule attached to a policy activity can delete only pending policy activities.

SpawnActivities: This business rule is attached to a transaction or requirement in order to spawn an activity from the transaction or requirement.

StatusChange: This business rule processes any status changes on a policy bringing the policy up to date. Processing the transaction or requirement to which this business rule is attached will change the status of the policy to the one specified in this rule. For example, changing the policy status from lapse to loan repayment.

[TransactionBusinessRulePacket](#): This business rule controls the order in which the rules attached to the transaction are processed.

TransactionCosmetics: This business rule controls the icon, button and reverse icon for the associated transaction. It also sets amount values from the transaction to display on the Activity screen. A tool tip element can be used to provide custom details to the user when the cursor hovers over icons in OIPA.

UpdateRoleStatus: The UpdateRoleStatus business rule allows a transaction to update the status of multiple roles. The update is not necessarily to the same status. Each status on a role is updated to its own value. The system supports three status code values from AsCodeRoleStatus:

- 01: Active
- 98: Inactive
- 99: Deleted

The OIPA implementation may develop more role status codes as needed. The rule is attached to a transaction and must be listed in TransactionBusinessRulePacket.

ValidateExpressions: The Validation section in a screen rule or a transaction supports the ability to configure hard edits and pop-up error or warning messages. The purpose of this business rule is to identify errors, and to fix and allow overrides. This business rule is overridden at the transaction level.

ValidateExpressions can also enable the ability to override errors from an activity's Verification screen by giving the OVERRIDABLE attribute of the <Expressions> element a value of "Yes". If errors are configured to be overridable, expressions defined as warnings will not have the option to be overridden.

A math variable (from the MathVariables section) or Field substitution may be made in the validation message. The only supported data types are 'TEXT', 'INTEGER', 'CURRENCY', 'DATE' or 'DECIMAL'. Therefore any other data types will need to be converted to TEXT, INTEGER or DECIMAL before being passed into the validation message. Special

characters such as '\$' or '%' need to be added in the validation message and will not be included in the substitution.

Formatting of substitutions will not be handled except with data type 'CURRENCY'. If a 'CURRENCY' math variable is used, then all that will display is the decimal portion without the currency code. The currency code should be added in the configuration. This can be accomplished by adding a math variable with the currency code and using it in the validation such as \$\$\$CurrencyCode\$\$\$.

Some validation messages need to reference the actual value of a math variable. In those situations, the ValidateExpressions or PostAssignmentValidateExpressions business rule may be attached to a transaction, with the math variable referenced in the Expression surrounded by \$\$\$.

```
<Expression TYPE="ErrorOnTrue" OVERRIDABLE="Yes" MESSAGE="Owner's Age of
$$$OwnerAgeMV$$$ must be less than or equal to $$$MaxAgeNY$$$">OwnerAgeMV &gt;
MaxAgeNY And IssueStateCodeMV = '32'</Expression>
```

In the expression above, if the OwnerAgeMV was set to 82 and MaxAgeNY is 80, then the validation message would appear when the lightning bolt icon is clicked on the Activity screen. The message would read, "Owner's Age of 82 must be less than or equal to 80."

Current V9 configuration supports assigning a math variable using a prefix plus a value from major system area such as Policy, Plan, Activity or Client in segment or transaction and activity Action/Events processing. Instead of assigning a prefixed value to a math variable (Plan: MinimumFaceAmount for example), the prefix and value may be used directly in the validation message.

A prefixed value can also be referenced in a validation message. If a face amount minimum limitation defined on a certain plan as \$100,000 needs to be exposed through a message, then the configuration would read:

```
<Expression TYPE="ErrorOnTrue" OVERRIDABLE="Yes" MESSAGE="Minimum Face Amount
for this plan is $$$Plan:MinimumFaceAmount$$$>FaceAmountMV &lt; MinimumFaceAmountMV>
</Expression>
```

VerificationScreen: This business rule is attached to a transaction to enable display of a Verify button on the Activity Detail screen. The Verify button allows users to display a separate Verification screen with custom information. The fields displayed on the Verification screen are configurable to include any field or any math variable that would be available in activity results had the activity been processed. The screen will also optionally display activity allocations, requirements and error messages. The SHOWORIGINAL attribute in the rule controls whether original allocations display along with the final allocations.

Transaction configuration can now define the buttons available to the Activity Detail screen. When the Verification Screen is attached to a transaction, the Verify button by default is available. If all buttons available to the screen are not needed, then the <Button> element (a sub element of <Buttons>) can be used in transaction XML to define the available buttons. If the button configuration is absent from the transaction configuration, then all buttons will be available by default. For more information on configuring buttons on the Activity Details screen, see the **Buttons Element** page in the XML Configuration Guide—navigate to **Transaction Rules | Transaction Elements | Buttons Element**.

The VerificationScreen rule and the QuoteScreen rule are mutually exclusive. If both the Quote Screen and Verification Screen are attached to a transaction, only the Verify button is available on Activity Detail.

OIPA users are able to override errors from the Verification screen. The ability to do so requires that the OVERRIDABLE attribute of the <Expressions> element in the ValidateExpressions rule be given a value of Yes.

The Verify button is only available for Client Financial and Policy Financial transactions – including any lower sub-categories of these transaction types (e.g. Policy-Financial-Reversible-Nonreversing, Policy-Financial-Nonreversible-Nonreversing, Client-Financial-Nonreversible-Nonreversing).

WriteDefaultAllocations: Allocations define how money is applied to a policy. The WriteDefaultAllocation business rule instructs the system to write/copy the default allocations, which are pre-defined funds, to the AsAllocation table for both policy and plan levels. It is customary to establish a level default allocation. The allocation specified in this business rule is used if no other allocation instructions are given or if transaction processing dictates that all funds are deposited into a certain fund or set of funds.

This business rule allows an activity to insert/update default allocation records. It is also used to control and recognize only positive allocations and ignore the negative allocations for all the source funds. The effective date column can be set in AsAllocation table via this business rule when the new allocation records are written/copied. This business rule can be used with any transaction (like RebalanceStart, InitialPremium, SystematicWithdrawalStart, etc.) that performs fund allocations.

You are here: Configuration > Suspense > Generate Suspense Business Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Generate Suspense Business Rule

The GenerateSuspense business rule is used to create a suspense record when a transaction with this attached rule is processed. The rule populates a value in the fixed fields of the AsSuspense table (SuspenseScreen) as well as in the dynamic fields of the AsSuspenseFields and AsSuspenseMultiValueFields table. Examples of these fixed fields are: Policy Number, Effective Date, Amount, Type Code and Status. Some of these fields can be populated based on the configuration defined in this business rule. Suspense records and the accounting to the suspense records can be conditionally created.

Attach the GenerateSuspense Business Rule

The GenerateSuspense business rule must be attached to transactions. When the transaction is processed as an activity in OIPA, it will automatically generate a suspense record.

There are two steps involved in this process. First, configure the GenerateSuspense business rule. Then attach the business rule to the transaction.

Steps to Configure GenerateSuspense Business Rule

1. Navigate to the Global Rules Explorer tab and open **Business Rules | Attached**.
2. Right-click on the GenerateSuspense node and select **New GenerateSuspense Override**.
3. Select the Company, Plan and transaction where the override will apply.
4. Click **Finish**.

Steps to Attach GenerateSuspense Business Rule

1. Navigate to the Main Explorer tab and open the plan where the transaction resides.
2. Open the transaction folder and find the transaction.
3. Right-click on the transaction's XML file and select **Edit Attached Files**.
4. Move the GenerateSuspense business rule over to the [attached rules](#) column.
5. Click **Finish**.
6. Open the XML for the rule override and configure as needed.
7. Check-in the rule to save changes to the database.

Note: If multiple suspense records are needed, the multisuspense element can be used. Refer to the **XML Configuration Guide** section **Transaction Elements | Suspense** for additional information.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Funds > ReassignAllocations Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Reassign Allocations Rule

The ReassignAllocations business rule is used to override existing activity allocations. This is useful when the end user is not allowed to select allocations or with automated activities where the user is not involved in an on-line experience with the activity. Allocation records are created when this business rule is processed. The Allocation records contain the funds the money is moving into or out of.

This should be used in lieu of configuring Allocation elements. Even if allocations tags are configured and selected in an activity, the attached ReassignAllocation business rule's configuration takes precedence and applies money according to the business rule. This situation should be avoided.

The Assignment Type when using ReassignAllocations should be a type that expects allocations to direct the money movement, such as Apply or FullWithdrawal. An assignment type like ApplyByFund is expecting a fund specification and is not appropriate with ReassignAllocations.

This business rule must be attached (overridden) at the transaction level. There is no visual editor in the Rules Palette, so it will need to be configured in XML.

XML Syntax

```
<ReassignAllocations>  
  <From>SELECT SQL statement </From>  
  <To> SELECT SQL statement</To>  
</ReassignAllocations>
```

Identify where money comes from and where it goes.

- The <From> tag is used to specify an allocation that moves money out of funds. This tag is not necessary when money is always moving into funds. For example, a Premium transaction is always moving into funds and not out of funds.
- The <To> tag is used to specify an allocation that moves money into funds. This tag is not necessary when money is always moving out of funds, such as a withdrawal or surrender.

For both the <From> and the <To> tag, a SELECT SQL is written to load values in creating allocations for the current activity. Three columns must be returned in each SQL: the FundGUID, the AllocationMethodCode, and then, depending on the AllocationMethodCode, either AllocationPercent, AllocationAmount, or AllocationUnits. For each record returned in the SQL statement, an AsAllocation record will be generated for the activity. Any allocations generated by the <From> tag will be treated as money out allocations and will have their allocation (AllocationPercent, AllocationAmount, or AllocationUnits) set as a negative value. Any allocations generated by the <To> tag will be treated as money in and will be positive.

Use this rule to perform either money in or money out. Neither tag is required, but one of the two tags must be used. If money needs to be transferred, then configure both tags.

This rule's purpose is to set the allocations that will be used in the activity's Assignment processing. Most Assignment types rely on the activity having a group of AsAllocation records that already exist in order to know how to generate AsValuation records. For example, an Apply

Assignment type will create an AsValuation record for each AsAllocation record for the activity. See the [Assignment](#) section for more details on how valuation records are created from all Assignment types.

Executing this rule makes any allocation records for the activity inactive. It supersedes any allocations that the activity may already have.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Global Explorer and Business Rules > Attached Rules > TransactionBusinessRulePacket

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transaction Business Rule Packet

Every new transaction automatically overrides and attaches the **TransactionCosmetics** and **TransactionBusinessRulePacket** business rules. You **do not** need to manually override and attach these business rules. These two rules will have been copied from the existing Global Rules. They are mandatory for the transaction to process and cannot be deleted from the transaction; however, the information may be edited.

TransactionCosmetics business rule controls the icon, button and reverse icon for the associated transaction. You can also set amount values from the transaction to display on the Activity screen. A tool tip element can be used to provide custom details to the user when the cursor hovers over icons in OIPA.

TransactionBusinessRulePacket business rule controls the order in which the rules attached to the transaction are processed. All rules attached to a transaction must be listed in the **TransactionBusinessRulePacket** in order to be processed. [Exceptions to this](#) are rules that do not require processing for the rule to be invoked. The Rules Palette automatically updates the **TransactionBusinessRulePacket** with the rules you attach. Please see the [Attached Rules](#) section for steps on attaching rules. You may need to re-arrange the order in which the rules are listed because it directly correlates with the processing order.

In the **TransactionBusinessRulePacket** business rule, attached rules must be listed with the exact name of the rule as the value of the `<Rule>` tag. The `<Rule>` tag is a sub-element of the `<TransactionBusinessRulePacket>` tag.

```
<TransactionBusinessRulePacket>
  <Rule>CopyToPolicyFields</Rule>
  <Rule>CopyToSegmentFields</Rule>
  <Rule>DeleteActivity</Rule>
</TransactionBusinessRulePacket>
```

Example XML for TransactionCosmetics

Steps to Edit the Attached Business Rules

1. Edit the attached TransactionCosmetics XML file to change the graphics and display Amount Values. In the TransactionCosmetics business rule graphic files must be listed with their exact file name as the value of the <Icon>, <Button> and <Reverse> sub-element tags. These tags must be in-between a start and end <TransactionCosmetics> tag.

```
<TransactionCosmetics>
  <Icon>AsIconDeathPending.GIF</Icon>
  <Button>AsButtonDeathPending.GIF</Button>
  <Reverse>AsReverseDeathPending.GIF</Reverse>
</TransactionCosmetics>
```

Example XML for TransactionCosmetics

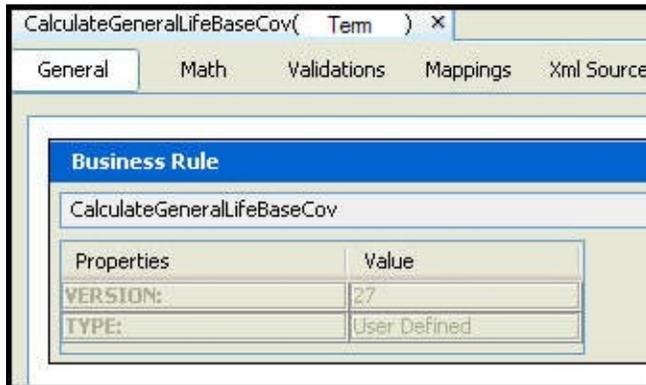
2. Use the [General](#), [Fields](#), [Math](#), [Debug](#), [Assignment](#), [Spawn](#), [Events](#), [Allocations](#) and/or [XML Source](#) sections for information on how to configure.
3. Override and attach needed business rules to the transaction. See the [Attached Rules](#) section for steps on attaching rules. Review the TransactionBusinessRulePacket business rule's XML file to make sure the attached rules are listed in the proper [processing order](#).

You are here: Global Explorer and Business Rules > Calculate Rules > General Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Calculate General Pane

The General Pane displays the name of the rule, the version and the type, which is User Defined.



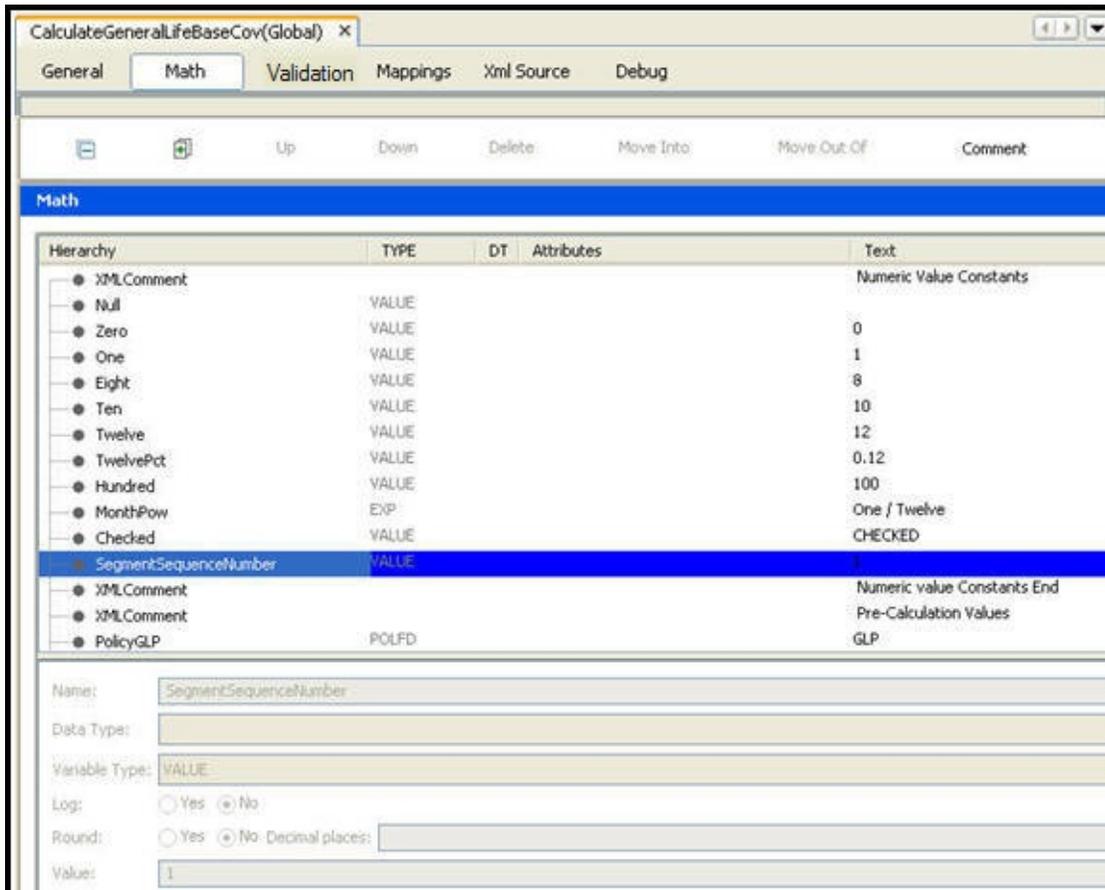
Calculate General Pane

You are here: Global Explorer and Business Rules > Calculate Rules > Math Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Calculate Math Pane

The Math pane in Calculate Rules is used to perform calculations and can use information from Segment screens as well as any stored values. The Math pane allows for the visual configuration of math. More detailed information on configuration using the Math pane can be found in the [Math Pane](#) section.



Calculate Math Pane

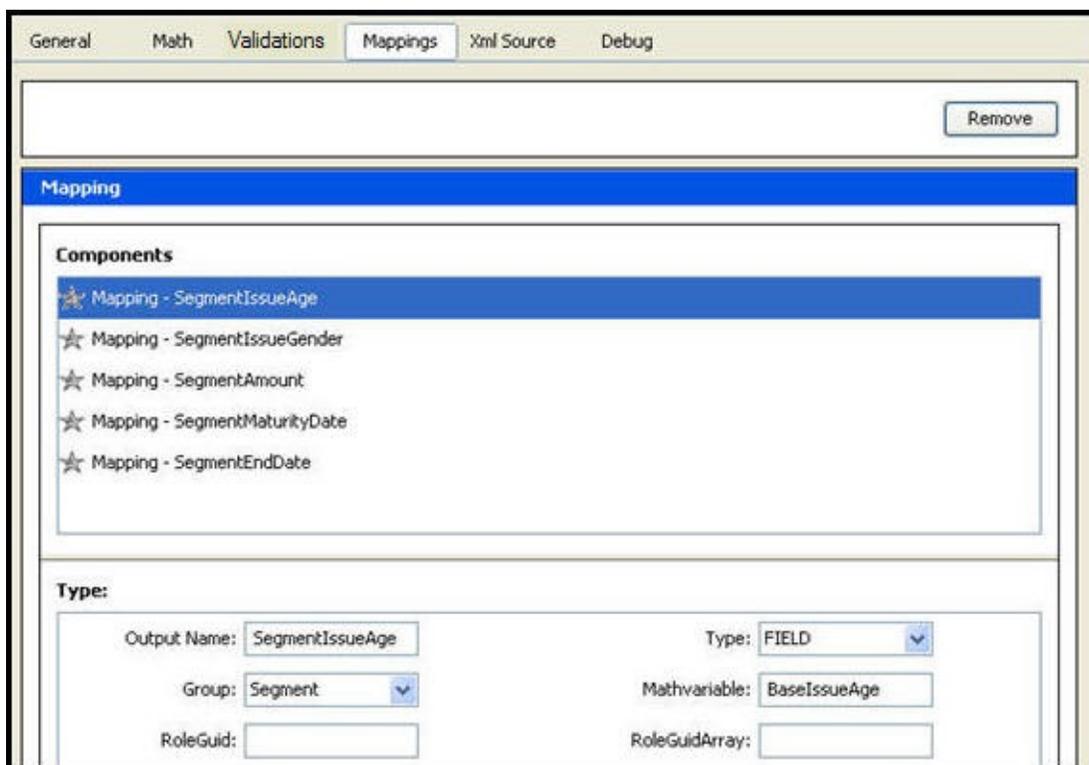
You are here: Global Explorer and Business Rules > Calculate Rules > Mapping Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Calculate Mappings Pane

The Mapping pane allows for the configuration of the output section of this business rule. Mapping updates a field on a screen with the value of a math variable that was calculated in this rule. The value of the math variable is written to a specified table that will display on the screen if the field is not hidden. To create a new mapping, drag the mapping component to the Mapping configuration window.

- **Output name** is the name of the field that will be updated. The name of the field must exactly match the name configured on the screen to update.
- **Type** should equal FIELD because that is what is being updated.
- **Group** is used to indicate the screen being mapping to. For instance, this value could be policy, segment or role.
- **MathVariable** is the name of the math variable in the Calculate Rule that holds the value to output.



The screenshot shows the 'Mappings' tab of a configuration window. At the top, there are tabs for 'General', 'Math', 'Validations', 'Mappings', 'Xml Source', and 'Debug'. A 'Remove' button is located in the top right corner. Below this is a 'Mapping' section with a 'Components' list containing five items: 'Mapping - SegmentIssueAge' (selected), 'Mapping - SegmentIssueGender', 'Mapping - SegmentAmount', 'Mapping - SegmentMaturityDate', and 'Mapping - SegmentEndDate'. Below the list is a 'Type:' section with the following fields: 'Output Name' (SegmentIssueAge), 'Type' (FIELD), 'Group' (Segment), 'Mathvariable' (BaseIssueAge), 'RoleGuid' (empty), and 'RoleGuidArray' (empty).

Calculate Mapping Pane

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Allocations > Configure Allocations > Calculate General Benefit Split Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Calculate General Benefit Split Pane

The Benefit Split pane allows the elements and attributes supporting Benefit Split to be visually configured. The elements configured in the Benefit Split section are used to calculate Benefit Split records based on specific allocations.

An explanation of the fields on the visual editor is explained below. For specific element and attribute explanations, refer to the **XML Configuration Guide** in the Help menu.

When setting up Benefit Split, an override of the CalculateGeneralBenefitSplit rule is used to create the Benefit Split records. Navigate to **Global Explorer | Business Rules | Calculate** to find the rule. Right-click the rule and create a new override.

- **Allocation:** This section contains the fields for capturing allocation information.
 - Type: allocation types are defined in AsCodeAllocationType.
 - Level: allows for the association to a specific policy or to a plan.
 - Merge: optional element that controls whether fixed fund(s) allocations can be pulled from outside the Allocation screen. Value should be Yes or No or a valid math variable. No is the default option.
- **Add Allocations:** This section defines the allocation information. It is required if the Merge field above is set to **Yes** or a valid math variable.
 - Type: allocation types are defined in AsCodeAllocationType.
 - Level: allows for the association to a specific policy or to a plan.
 - Percentage: defines the percent of allocation to be applied.
 - Final Allocation Type: this controls writing new AsAllocation records with the result of the merged allocations. Final Allocations are saved at the Segment level. The type selected determines the type of allocation that is written as the final allocation record in AsAllocation. Types are pulled from

AsCodeAllocationType.

- **Relation:** This section contains relation keys that are used to link benefit funds to the parent/child allocations. The NAME attribute identifies the Fund Field and the DATATYPE attribute specifies the data type of the criteria value and Fund Field.

A FixedBenefitFund sub element is required if segment supports fixed benefit payouts. The element holds a variable containing a 03 fund type code FundGUID that all fund type code 01 fixed allocation(s) will merge to

- Criteria Section: Click **Add** to open a new row. The Criteria field identifies the fund field that is matched to the Input math value. The value of the criteria is a math variable from the CalculateGeneral Input math or a literal value. The Data Type field displays a drop down box of available data types for the criteria value and Fund Field. The Value field is a drop down box populated with the available math variables that match the data type selected. Use **Delete** to remove a criteria row.
- Fixed Benefit Fund: this is required if the segment supports fixed benefit payouts. The element holds a variable containing a 03 fund type code FundGUID where all fund type code 01 fixed allocation(s) will merge.



The following element values are read in from Calculate General Math and applied to the Final Allocations calculated above

- Effective Date: this is a required field that contains a Math Variable date value used to look up the unit values of the benefit funds.
- Create Deferred Split: this is an optional field that holds a math variable or literal value of **Yes** or **No** indicating if type 51 benefit split records should be created. The default is No.
- Variable Benefit: this is a required field that holds a math variable amount (currency or decimal) representing the

calculated variable benefit amount. This is currently only used when 'Solve for Benefit' is selected. The currency is assumed to be the Plan default.

- o Fixed Benefit: this is a required field that holds a math variable amount (currency or decimal) representing the calculated fixed benefit amount. This is currently only used when 'Solve for Benefit' is selected. The currency is assumed to be the Plan default.

CalculateGeneralBenefitSplit(Global) ⓘ

General Math Mappings **Benefit Split** Xml Source Debug

Allocation

Type: Level: Policy

Merge: MergeAllocations

Add Allocations

Type: Plan Default - 01 Level: Plan

Percentage: Final Allocation Type : Segment Default - 05

Relation

Criteria	Data Type	Value
BandVariable	TEXT	PolicyBandVariable
InvestmentRate	TEXT	BenefitSplitBand

Add Delete

Fixed Benefit Fund : FixedBenefitFundGUID

Effective Date: UnitValueDate Create Deferred Split: DeferredSplit

Variable Benefit: VariableBenefitAmount Fixed Benefit: FixedBenefitAmount

Benefit Split Pane of Calculate Rule

You are here: Global Explorer and Business Rules > Calculate Rules > Validations Pane

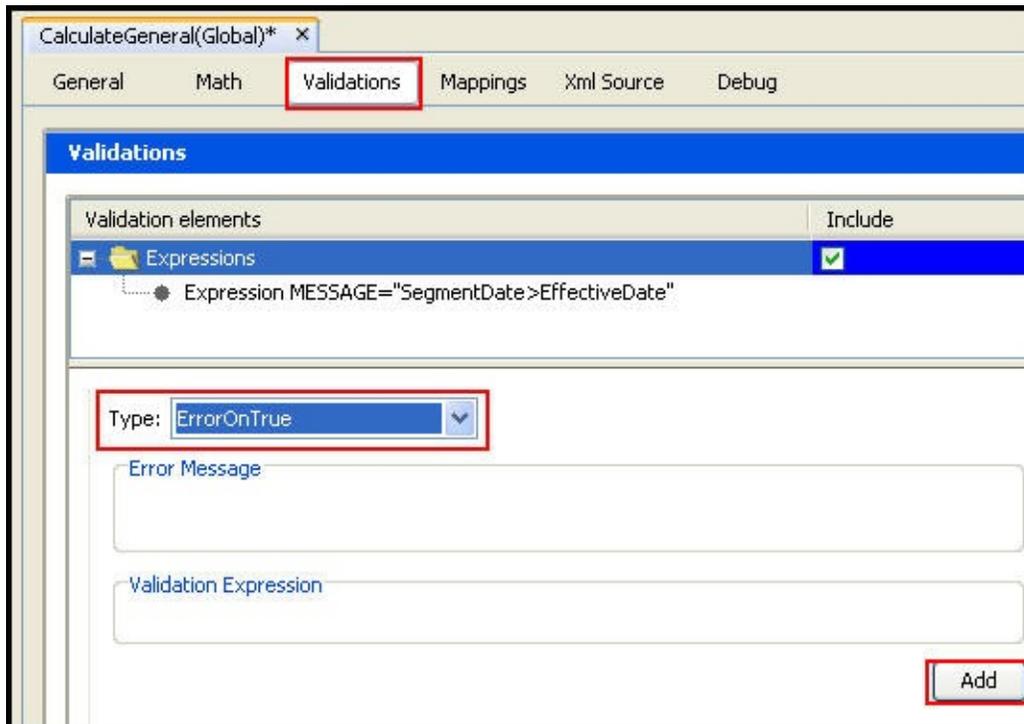
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Calculate Validations Pane

In this pane, validate the data a user inputs in a segment field to ensure it is the type of data required. The validation occurs when the user attempts to add the segment to the policy by hitting the **SUBMIT** button.

Steps to Add a Validation

1. Select the **Expressions Folder** under Validation Elements.
2. Select the type from the **Type** drop-down box.
3. Select the **Add** button.



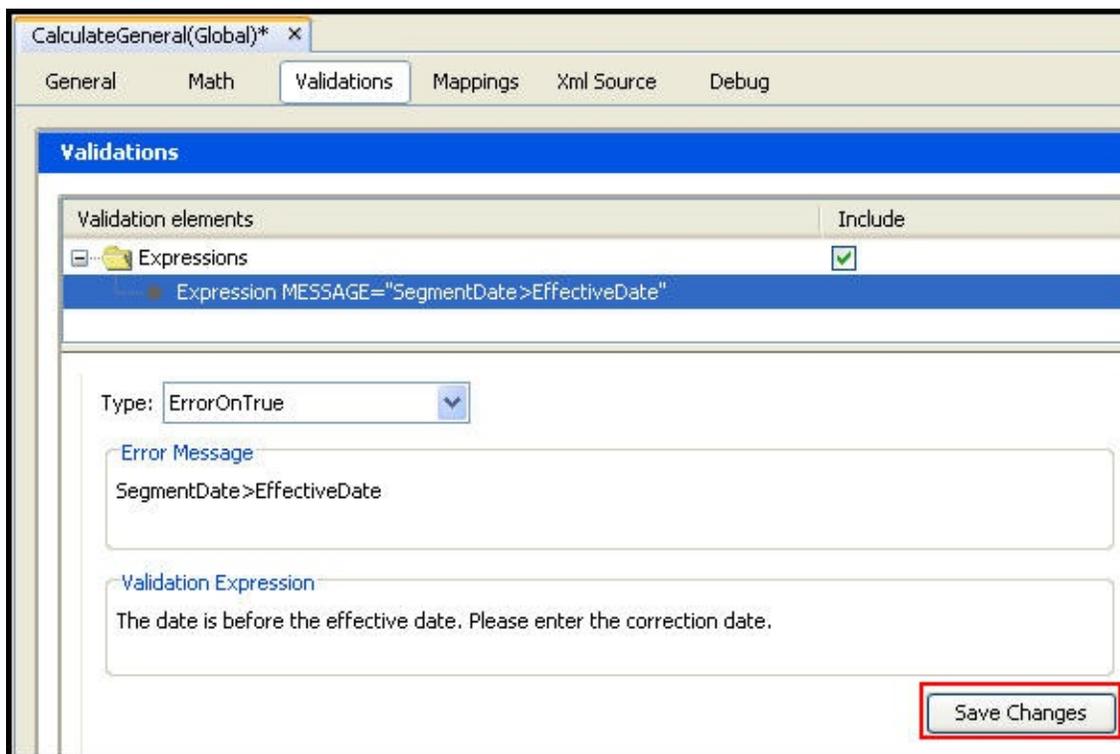
Calculate Validations Pane



To remove an expression, right-click on the expression name and select **Remove**.

Steps to Create the Error Message and Validation Expression

1. Select the **Expressions** Folder.
2. Click the new Expression Message you just created.
3. Enter the error message in the error message field. Use basic HTML formatting tags in the Error Message.
4. Enter the validation expression in the validation expression field. Use JavaScript functions and logical operators to validate conditions and check criteria. Please see the **V9 XML Configuration** topic in this help system. Additional information about Expression Writing is located in **Common Elements | Operators Available for Expression Writing**.
5. Click **Save Changes**.



Calculate Validations Pane Save Changes Button

You are here: Configuration > Configuration Panes > XML Source Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

XML Source Pane

Oracle Insurance applications use a specific custom XML syntax employing elements, attributes and values to define the data in application rules. This pane is used to configure the XML when a visual editor is not available.

Refer to the **XML Configuration Guide** topic in the **Help** menu for a complete list of all elements, attributes and values needed for configuration.

XML Components

Elements

The first element in an XML structure is the root element, which is usually the name of the rule being configured. All elements come in pairs and consist of a start tag and an end tag. Tags are surrounded by the two characters < and > . These are angle brackets. Every tag can have a value, which is what is listed between the start and end tag.

Sub-elements

Any elements that occur prior to the root elements end tag are called child or sub-elements. Child elements have start and end tags and may have children of their own.

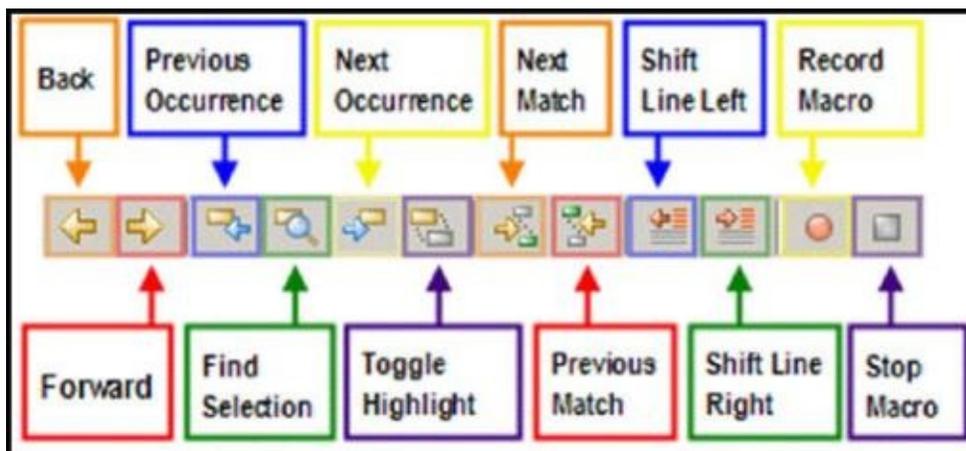
Attributes

Elements can have attributes that further define the data. Attributes are always listed in the start tag. Attributes consist of a name and value, where the value is in all capitals, followed by an equal sign, and the value is enclosed in quotes.

XML Editor Tools

The XML Editor tools are available to aid configuration. An explanation of each tool is provided below.

- **Back:** This moves to the last step that was performed.
- **Forward:** This moves forward from previous steps taken.
- **Previous Occurrence:** This finds the previous occurrence of the highlighted text.
- **Find Selection:** This finds each selection of the occurrence of the highlighted text.
- **Next Occurrence:** This finds the next occurrence of the highlighted text.
- **Next Match:** This inserts the next word.
- **Shift Line Left:** This shifts highlighted text to the left.
- **Record Macro:** This begins macro recording.
- **Toggle Search Highlight:** This turns on and off all highlighted instances of searched text.
- **Previous Match:** This inserts the previous word.
- **Shift Line Right:** This shifts highlighted text to the right.
- **Stop Macro:** This stops macro recording.



XML Editor Tools

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Global Explorer and Business Rules > CopyBooks > CopyBooks Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

CopyBooks Overview

A CopyBook is a type of business rule that allows for plans, transactions and other rules to share common functionality. Maintenance for configuration is easier as the CopyBook configuration is held in one place. This configuration can contain data for events such as the creation of fields for screens, actions for math and spawns.

When a transaction or rule references a CopyBook the contents of that CopyBook are [resolved](#) and inserted inside that rule or transaction at the specific place the CopyBook is referenced.

CopyBook Functionality

- [Create a new CopyBook](#)
- [Create a CopyBook override](#)
- [Call a CopyBook](#)
- [Resolving rules](#)
- [Delete a CopyBook](#)

CopyBook Tips

- The CopyBook must exist as a global business rule prior to an override being added.
- CopyBooks can be overridden at specific levels to support variations in calculations, fields, validations, etc. CopyBooks are usually resolved using the context applied to loading the rule where the copybook is referenced.
- Confirm the fields and/or math variables referenced but not defined within the CopyBook exist within the transaction that calls the CopyBook.
- In this release, entire sections of a CopyBook can be added on all panes (except the General). If you want to add multiple sections of a CopyBook or a partial section of a CopyBook, you can only do this on the Math and Fields pane. For all other panes, you will need to

use the [XML Source Pane](#).

- CopyBooks cannot be created for requirements.
- CopyBooks cannot be added to the following:
 - Debug pane
 - Events pane of a transaction
 - Events pane or buttons of segments
- When using CopyBooks with Assignments, the following combinations are supported:
 - Assignment alone
 - Assignment and Math
 - Assignment and Fields
 - Assignment and other sections as long as one of the other sections is Math and/or Fields.

You are here: Global Explorer and Business Rules > CopyBooks > CopyBook Override Levels

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

CopyBook Override Rules Contexts

CopyBook Override Rule Contexts are described in the section below. While some rules can be overridden at the Fund level, CopyBooks do not have an override level at Fund.

CopyBook Override Rules Contexts

Primary Company Context:

Global > Primary Company



The rule [DuplicateClient](#) presents an exception in that this rule only resolves to Company and cannot resolve to Global.

Subsidiary Company Context:

Global > Subsidiary Company

Plan Context

Global > Subsidiary Company > Product > Plan

State overrides are not available. If Product functionality is not enabled in the environment, then an override at that level will not be available.

Policy Context

Global > Subsidiary Company > Product > Plan

State overrides are available at all levels with the exception of Global.

Activity Context

Global > Company > Product > Plan > Transactions

In situations where a client chooses not to implement PlanGroup as a part of the business structure, the system will not search for a PlanGroup override when loading the best match for the rules. The check for Product override is dependent on the system property setting for Product. When Product is not enabled, the Activity context can be illustrated as follows:

Global > Company > Plan > Transactions

The following contexts have been added to further define the Activity override levels. Determination of specific activity context is made based on company, primary or subsidiary, to which the plan is aligned.

- **Activity Context (Policy):** Global > Subsidiary Company > Product > Plan > Transaction.

- State overrides are available at all levels with the exception of Global.
- Although not an exception to the Activity context (Policy), [TransactionTimes](#) is a departure from the other rules defined in this context in that it exists with transaction overrides in addition to Plan and Subsidiary Company overrides.
- **Activity Context (Client):** Global > Primary Company > Plan > Transactions
- **Activity Context (Company):** Global > Primary Company > Plan > Transaction.
 - Company level transactions are attached to the plan under the Primary Company.
- **Activity Context (Plan):** Global > Subsidiary Company > Plan > Transaction.

Fund Context

Plan > Fund

State overrides are available at all levels except the global level. The Fund context definition is reflective of rules that allow a Fund override. CopyBooks used within a rule allowing Fund override may resolve to Plan, Product, Subsidiary Company and Global, but Funds themselves are at a Plan level only.

Example of CopyBook Overrides Used in Configuration

The scenario below demonstrates how OIPA determines which CopyBook override to use when multiple CopyBooks are available.

CopyBook A is a Global CopyBook containing specific common term life fields. **CopyBook B** has a Global with a Subsidiary Company context containing unique fields for that Company. **CopyBook C** has a Global with a Plan context containing fields relating to that specific term life product. Lastly, **CopyBook D**, with a Global and a Transaction level context was referenced in the screen configuration.

CopyBook A was resolved at the Global level. **CopyBook B** was resolved at the Subsidiary Company level. **CopyBook C** resolved at the Plan level. **CopyBook D** was resolved at the Global level.

You are here: Global Explorer and Business Rules > CopyBooks > Create a new CopyBook

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create a New CopyBook

A CopyBook must exist as a global business rule before an [override](#) can be added. Create an override if a particular transaction is used by two different plans, but works differently for each plan. For example, create the Anniversary CopyBook as discussed below. Then create an override of the Anniversary CopyBook for each plan that uses it.

Create a CopyBook for an [entire transaction](#), [a section of a transaction](#) or [a business rule](#).

Steps to Create a New CopyBook

1. Go to **Global Rules Explorer** tab.
2. Right-click on the environment folder.
3. Select **New Global Rule for Category CopyBooks**.



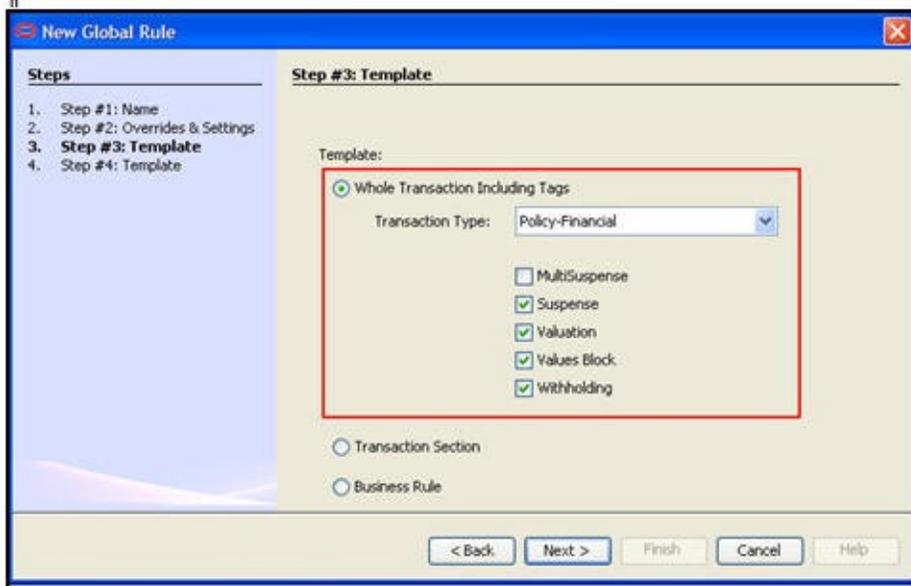
CopyBook Right-Click Option

4. Enter a name for the CopyBook and select **Next**. CopyBook names should always begin with *CopyBook-* then the name of the specific CopyBook. No spaces are allowed in the CopyBook name and the format should be in camel case.
5. Select **CopyBook** from the TypeCode drop down box and select **Next**.

At this point there is the option to copybook an entire transaction, a specific section of a transaction, a specific business rule or part of a business rule.

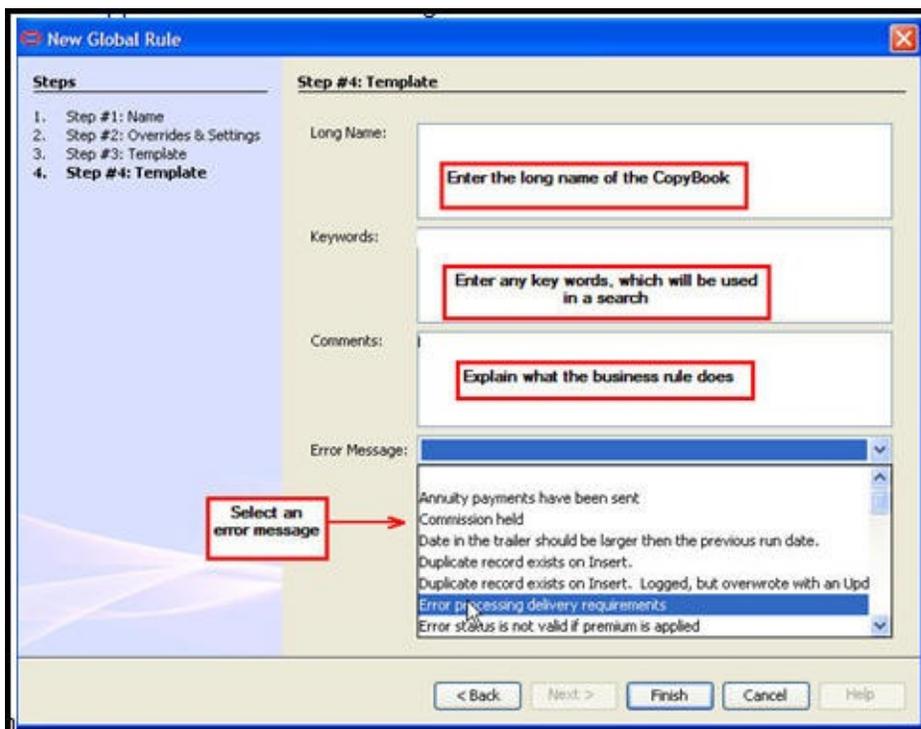
Copybook an Entire Transaction

1. Select the radio button next to **Whole Transaction Including Tags**.
2. Select the transaction type from the drop down box.
3. Select **Next**.



CopyBook Option to Copy Whole Transaction

4. Fill out the template information and select **Finish**.

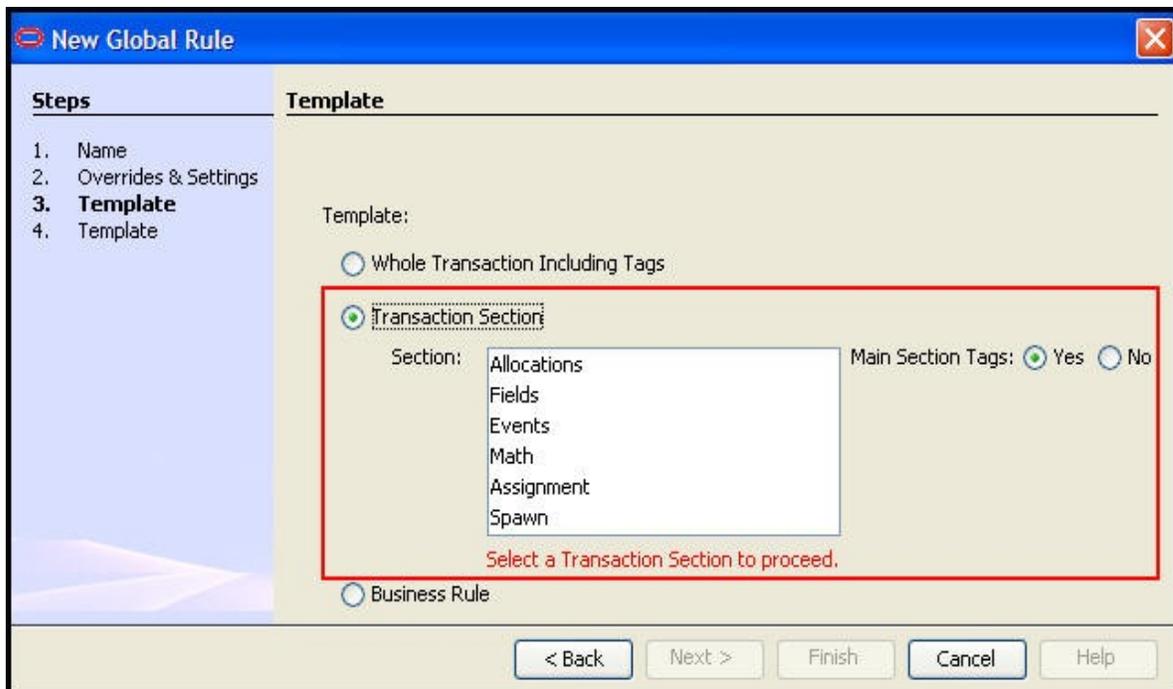


Step Four of the CopyBook Wizard

Copybook a Transaction Section

1. Select the radio button next to **Transaction Section**.
2. Select the section of the transaction to use. If multiple sections are needed, hold the CTRL key down and click the transaction sections.
3. Select **Next**.

Note: If only one section is selected, it can be included without tags. If multiple sections are selected, tags must be included.



CopyBook Option to Include Specific Transaction Sections

4. Fill out the template information and select **Finish**.

CopyBook a Business Rule

1. Select the radio button next to **Business Rule** and then select **Next**.
2. Fill out the template information and select **Finish**.

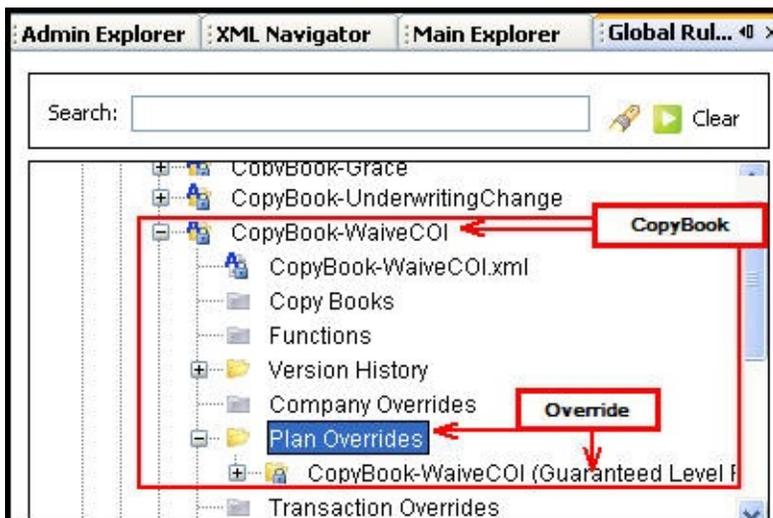
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Global Explorer and Business Rules > CopyBooks > Create a CopyBook override

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create a CopyBook Override

When a CopyBook needs to be updated to make it more specific for a particular plan, a CopyBook override can be created. From the Global Rules Explorer tab, open the **Business Rules | CopyBooks folder**. Right-click on any CopyBook to create the override. Once the override is created, it will reside in the corresponding Overrides folder. It will retain the original name of the CopyBook with the override level in parenthesis. So, an override for the WaiveCOI transaction for the Guaranteed Level Premium Term plan would be named CopyBook-WaiveCOI (Guaranteed Level Premium Term).

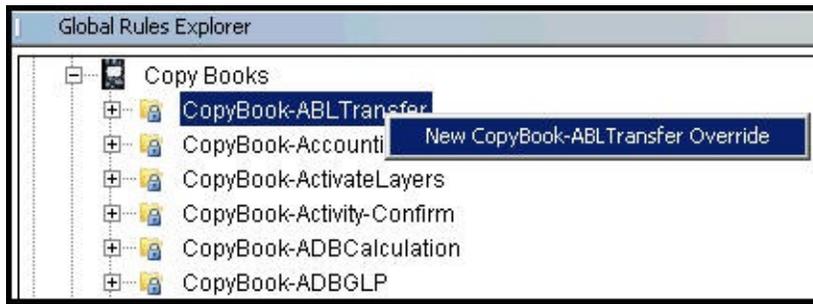


CopyBook and CopyBook Override

OIPA supports CopyBook overrides various levels. Refer to the [CopyBook Override Levels](#) section for details. The system will look for overrides at the lowest level and move up to global. When overrides are created, the Rules Palette Override Wizard limits the available levels according to what is supported in OIPA. Some rules have multiple contexts. The Rules Palette will not restrict those rules to any override levels.

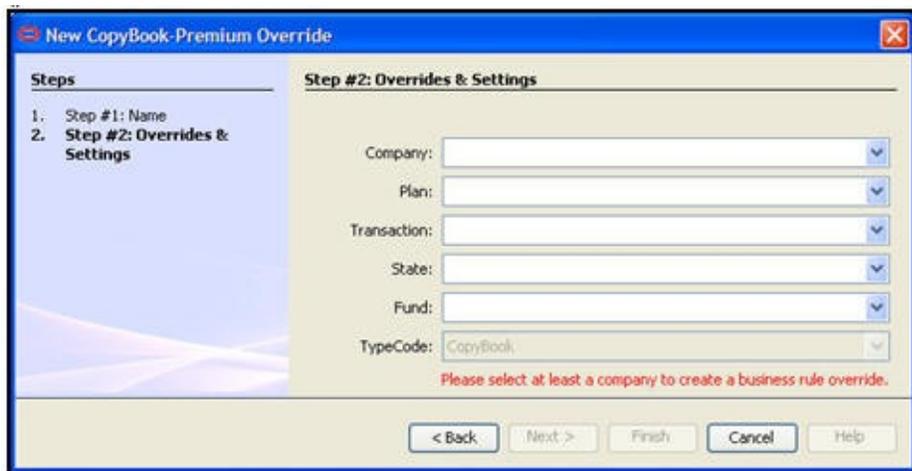
Steps to Create an Override of a CopyBook

1. Go to **Global Rules Explorer** tab.
2. Locate the CopyBook and right-click on the CopyBooks folder.
3. Select **New CopyBook Override**.



CopyBook Right-Click Option

4. Select **Next**. The name is already provided.
5. Select the appropriate override level and then select **Finish**.



Step Two of the CopyBook Override Wizard

6. Right-click on the CopyBook Override XML and select **Check-in**.
7. Right-click again and select **Check-out**.
8. Configure as necessary.
9. Right-click on the file and select **Check-In** to save the changes.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Global Explorer and Business Rules > CopyBooks > Call a CopyBook

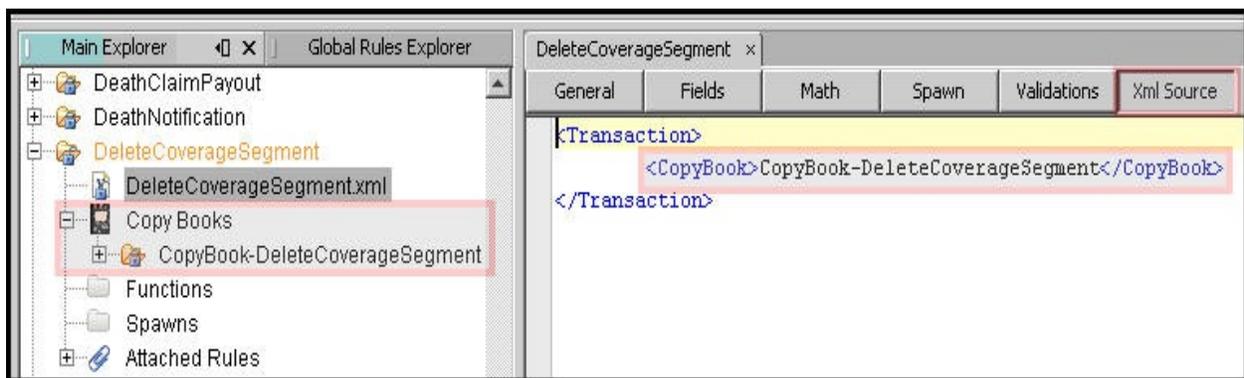
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Call a CopyBook

Steps to Call a CopyBook in XML Source

1. Navigate to the rule where the CopyBook will be called.
2. Right-click on the XML file.
3. **Check-out** the rule.
4. Open the XML Source Pane.
5. At the point of configuration the CopyBook should run, type the start and end `<CopyBook></CopyBook>` tags with the **CopyBook-Name** as the value.
6. Save  the rule.
7. **Check-in** the rule.

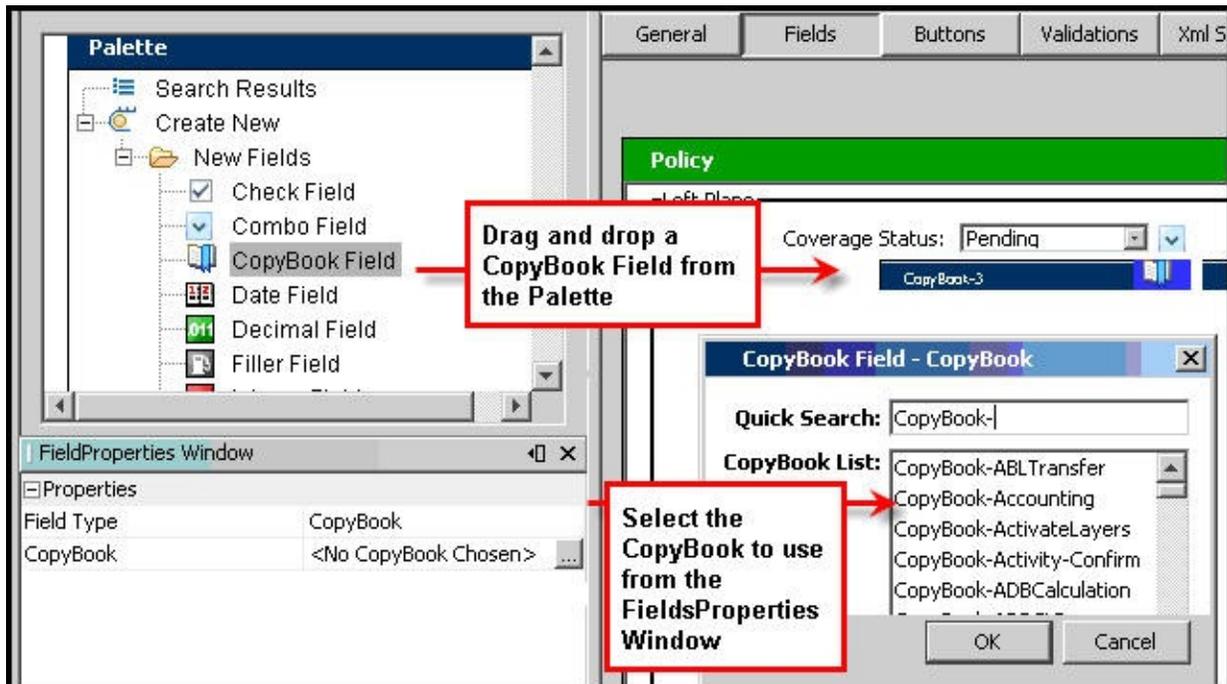
When rule's CopyBook folder is opened in the **Main Explorer** tab, the CopyBook being called will be listed.



CopyBook element in Transaction XML

Steps to Call a CopyBook in Fields

1. Navigate to the rule where the CopyBook will be called.
2. Right-click on the XML file.
3. **Check-out** the rule.
4. Open the Fields Pane.
5. Using the Palette, drag and drop a CopyBook field onto the screen.
6. In the **FieldProperties** Window select the  button.
7. Select the CopyBook from the CopyBook list and select **OK**.
8. **Save**  the rule.
9. **Check-in** the rule.



Drag CopyBook from Palette to Fields Pane

You are here: Global Explorer and Business Rules > CopyBooks > Delete a CopyBook

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Delete a CopyBook

Steps to Delete a CopyBook

1. Navigate to the Global Rules Explorer navigation tree and open **Business Rules | CopyBooks**.
2. Scroll through the list of CopyBooks and select one to delete.
3. Right-click on the CopyBook to delete and select **Delete Rule**.
4. Query the database to make sure the CopyBook is not referenced anywhere else in the system. If it is, delete those instances as well.

If the CopyBook override is not removed from the XML Source, then an error message will display when the transaction is compiled.

You are here: Configuration > Business Rules > Resolve Rules

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Resolve Rules

Rule resolution is limited to transactions, attached rules, segment name rules, calculate general rules, screen rules and plan rules. To view all the XML code in a rule's configuration, select **Resolve rule** from the rule's right-click menu. Resolving a rule will show CopyBook information instead of just the CopyBook call. This displays the configuration in its entirety so that code can be analyzed thoroughly.

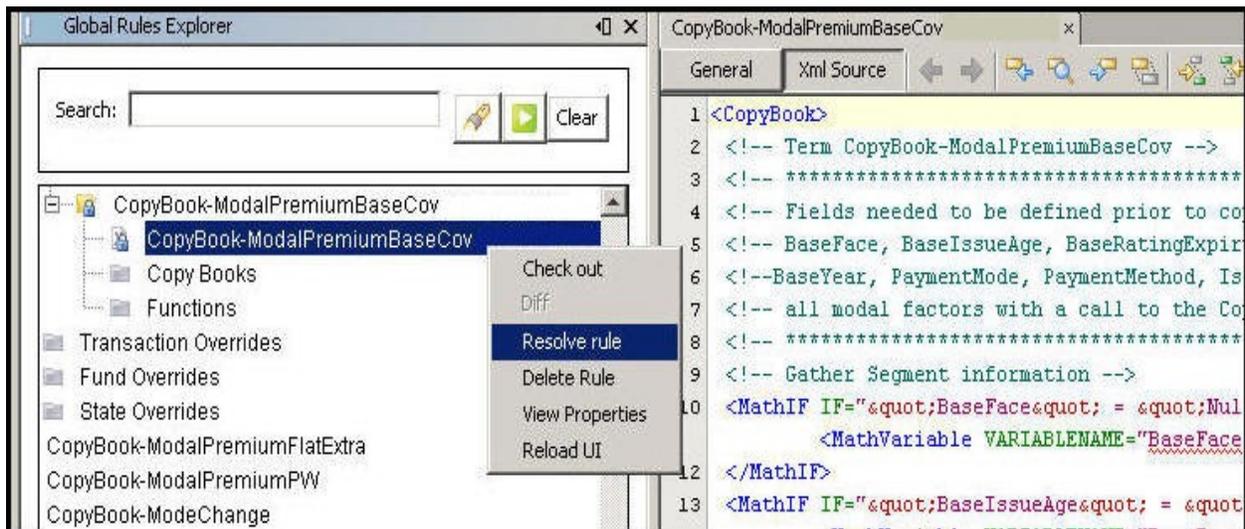
Rules That Do Not Resolve CopyBooks

There are two rules that do not resolve CopyBooks. They are [TransactionCosmetics](#) and [InquiryScreen](#).

Steps to Resolve Rules

1. Navigate to the rule and expand its folder.
2. Right-click on the rule's XML file.
3. Select **Resolve rule**. If a policy transaction is being resolved, a Context window will appear. A state context must be selected.

The resolved rule displays in the [Configuration Area](#).



CopyBook Resolve Rule Right-Click Option

You are here: Global Explorer and Business Rules > Functions > Functions Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Functions

In the Oracle Insurance Policy Administration (OIPA) system, functions contain common calculations that may be used multiple times. Functions can be defined at many different levels depending on the context of the container component (i.e. transactions, screen, etc.). It is best to place the function as high in the hierarchy as possible in order to be used by the most components.

There are two types of functions:

1. [Functions defined as rules](#) that are created to support business events.
2. [Pre-defined Java functions](#) that can be used to help with creating calculations.

Additional Function Considerations

When using Functions within CopyBooks:

- The override level of the contained functions does not need to match the override level of the contained CopyBook. The CopyBook may be defined at the global level and may contain a function that is overridden at a plan level.

Example: A global level CopyBook exists that validates that a premium paid is within the acceptable tolerance. A function is used in the CopyBook to look up the tolerance amount for a given plan. The function is overridden for each plan.

When using CopyBooks within Functions:

- When a CopyBook is used within a function, the override levels of the CopyBook and function do not have to match.

Example: A Premium Tax function exists at the global level. There are two state overrides, one for PA and one for NY. A Copybook exists within the Premium Tax function at the global level to look up the Federal tax information for the function. All three of the Premium Tax functions call the same Copybook.

When using Functions within Functions:

- Functions are available for use within other functions. A global function may contain calls to other global functions as well as to other override level functions. The override level does not need to match the level of the containing function. The levels can match, but OIPA will allow the levels to be different.

Example: In the global Base COI function there are several additional functions that are called. These include a Flat Extra calculation, which is a plan level override function, the lookup of current and guaranteed rates, which is a Global function, and a rate percent function, which is a plan level override.

When using Functions within Calculate General Business rules:

- Functions are available for use within Calculate General rules. The override levels of the functions do not need to match the level of the containing

CalculateGeneral.

When referencing a function multiple times:

- The function that is referenced multiple times in a specific activity, calculate general or screen, will always resolve to the same function override.

Example: FunctionTermPremiumCalc is defined with overrides at the Product level and plan level: TermPlanGroup and the AcmePlan. A BaseCoverage Calculate General defined for the AcmePlan defines function calls to FunctionTermPremiumCalc at lines, 15, 45, 75, and 110. The resolution of each call will always result in the same specific override within this Calculate General, the AcmePlan override.

You are here: Global Explorer and Business Rules > Functions > Create a New Function

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create a New Function

Functions are useful when a calculation or process needs to be repeated and made available for multiple transactions or business rules. Functions are also beneficial when only the output and result values are needed and not the intermediate values calculated by the function.

Function Configuration Standards

Functions are made up of four components. They are listed below along with a description.

- **Function names:** Best practice dictates that names should always begin with *Function-* then the name of the function. No spaces are allowed in function names. Ex: Function-ModalFactors.
- **Input parameters:** Passed from the transaction or business rule into the function configuration. Best practice dictates that names should always begin with *p*. Ex: pEffectiveDate.
- **Output parameters:** Passed from the function to the transaction or business rule that is calling the function. Best practice dictates that names should always begin with *o*. Ex: oFreeAmount.
- **Return variable:** The specific value the function is trying to obtain. Best practice dictates that names should always begin with *r*. Ex: rModalPremium.

Steps to Create a New Function

1. Go to **Global Rules Explorer** tab.
2. Right-click on the Functions folder.



Function Right-Click Option

3. Select **New Global Rule for Category Functions**.
4. Enter a name for the function and select **Next**. Function names should always begin with Function- then the name of the specific function. No spaces are allowed in the function name and the format should be camel case.
5. Functions are always created at a global level. The override fields will all be disabled; however, once a global function rule is created, it can be [overridden](#).
6. Select **Function** from TypeCode and then select **Next**.
7. Fill out the template information and select **Finish**.
8. Configure the function. Refer to the steps below for more details on configuring functions.

Steps to Configure a New Function

1. Make sure the Function is open in the Configuration Area. If not, double-click the .xml file to open it.
2. Click the Parameters tab.

General **Parameters** Math Xml Source Debug

Name:

Return:

Data Type:

Is Array
 Yes No

Input Parameters

Name	Data Type	Is Array
pEffectiveDate	DATE	<input type="radio"/> Yes <input checked="" type="radio"/> No
pPremiumPaymentArray	DECIMAL	<input checked="" type="radio"/> Yes <input type="radio"/> No
pPremiumDateArray	DATE	<input checked="" type="radio"/> Yes <input type="radio"/> No
pSurrenderAmount	DECIMAL	<input type="radio"/> Yes <input checked="" type="radio"/> No
pFreeAmountPercent	DECIMAL	<input type="radio"/> Yes <input checked="" type="radio"/> No
pMaxSurrenderYears	INTEGER	<input type="radio"/> Yes <input checked="" type="radio"/> No

Enter Input and Output Parameters

Output Parameters

Name	Data Type	Is Array
oFreeAmount	DECIMAL	<input type="radio"/> Yes <input checked="" type="radio"/> No
oPremiumsOutsideOfSurrenderPeriod	DECIMAL	<input type="radio"/> Yes <input checked="" type="radio"/> No

Function Parameters Tab

3. Enter the name of the function.
4. Enter the return value. Make sure it starts with the letter *r*.
5. Select the Data type.
6. Click the radio button for **Yes** or **No** if function is an array.
7. Click **Add** and click in the blank field to enter the input parameters.
 - input parameter name must begin with the letter *p*.
 - select the data type from the drop down box.
 - specify whether it is an array.
8. Click **Add** and click in the blank field to enter the output parameters.
 - output parameter name must begin with the letter *o*.
 - select the data type from the drop down box.

- specify whether it is an array.
9. Click the Math tab and configure the math section. Use all functionality available in the MathVariables section of Math with the exception of LOG.



:Refer to the [Math Pane](#) section for additional information on configuring in the Math Pane.

10. Check-in the XML file to save your changes to the database.

You are here: Global Explorer and Business Rules > Functions > Functions Defined as Rules

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Functions Defined as Rules

A function defined as a rule contains common calculations that can be broken out into logical pieces. Functions improve readability as the inputs and outputs are clearly defined. In addition, the majority of math variables used in a function do not need to be stored or used outside of the function. The math variables in a function are not stored in Math results, which improves system performance. Instead, the function is cached temporarily.

Functions Defined as Rules

- All math functionality available in transactions and business rule configuration is available within functions except LOG. Use LOG for the function's return value. Do not use LOG for math variables in the function itself.
- Functions may be global or they may be overridden. Overrides should be used as an exception to the more general higher level overrides.
 - The same functionality available in global functions is available in the override. Inputs and outputs must be clearly defined and math variables mapped to the input and output of the function must be defined prior to the function call or a stack trace will occur.
 - Every function has a signature. This signature is the name, parameters and the data types being passed in and out. There will be one signature that transcends all the overrides. All overrides are required to match a global signature.
- Functions are often chosen over CopyBooks for specific calculations since they offer increased readability.
- Functions can be defined at different levels. Container components using them will establish a context that finds the best matching level to execute for each individual function.
- If a function is used multiple times in a transaction/business rule it will be resolved and compiled once, no matter how many times it is used in that transaction/rule.

Important: Existing functions are stored in the Global Rules Explorer window under the function folder type. View a function's inputs and return value by double-clicking the function's XML file. In the XML source, the return value is written as an attribute of the function tag and pre-fixed with *r* for return. The inputs of the function are in the parameter tag and pre-fixed with *p* for parameter.

Steps to Call a Function Defined as a Rule

1. Right-click on the XML file to call the function in.
2. Select **Check out**.
3. Open the Math pane.
4. From the Palette window, scroll down to the folder labeled **rule** and drag and drop **FunCall** onto the Configuration Area.

Note: If the Palette window is not visible, open it by selecting **Open Search Palette Window** from the Window option on the main menu.

5. Name the math variable.
6. Click the **Data Type** field to select a data type from the drop down box.
7. Select whether to **LOG** or **Round** the function's output result.
8. Click the **Functions** field to select the function to call from the **Functions** drop-down box.
9. In the Input section, click the **Value** drop-down box to select each parameter's value.
10. In the Output section, click the **Value** drop down box to select each parameter's value.
11. Check-in the XML file to save the changes.

Math

Hierarchy	TYPE	DT	Attributes	Text	DT
Direction	VALUE	IXI		Next	Txt
XMLComment				Calculation o	
PolicyContractValue	FIELD	DEC		Valuation:Poli	011
AnnualContractFee	SQL	DEC	DEFAULT =		011

Name: **Enter MathVariable name**

Data Type: **Click field to select data type**

Variable Type: FUNCTIONCALL

Log: Yes No **Select Log or Round if needed**

Round: Yes No Decimal places:

Functions: **Click field to select function**

Input:

ParameterName	Data Type	Value
pEffectiveDate	DATE	<input type="text" value=""/>
pMaturityAge	INTEGER	<input type="text" value=""/>
pAnnuityAgeFullYear	INTEGER	<input type="text" value=""/>
pAnnuityAgeMonths	INTEGER	PolicyYear
pDateOfBirth	DATE	<input type="text" value=""/>

Output:

ParameterName	Data Type	Value
oAge	INTEGER	<input type="text" value=""/>
oAnnuitizationDate	DATE	<input type="text" value=""/>
oAnnuitizationAge	INTEGER	<input type="text" value=""/>
oRequiredDistributionDate	DATE	<input type="text" value=""/>

Palette

Search Results

Create New

- New Mathvariables
 - general
 - array
 - commutation
 - conditional
 - loop
 - ratelookup
 - rule
- CopyBook
- FunCall
- Function
- Process
- solve
- sql
- webservice
- xml

Select FunctionCall and drag onto Configuration Area

Click value field to select values

Math Section Displaying Function Parameters

Note: If the function's parameters are set-up incorrectly in the function, a red error will display in the bottom math variable text box.

You are here: Global Explorer and Business Rules > Functions > Pre-defined Java Functions

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Pre-defined Java Functions

Pre-defined Java functions are available to assist with calculation creation. Use these Java functions in the math section of rule configuration.

Note: View the **XML Configuration Guide** topic in this help system for a list of available Java functions. The Java functions are located in **Transaction Rule | Transaction Elements | Math Elements | Function**.

Steps to Use a Pre-defined Java Function

1. Check-out the rule to use a Java function in.
2. Open the **Math** pane.
3. From the Palette window, scroll down to the folder labeled **rule** and drag and drop **Function** onto the Configuration Area.

Note: If the Palette window is not visible, open it by selecting **Open Search Palette Window** from the Window option on the main menu.

4. Enter a name for the new math variable.
5. Click the **Data type** field to select a data type from the drop down box.
6. Select Log or Round if necessary.
7. In the Call text box, type the function and its parameters.
8. Check-in the rule to save the changes to the database.

Anniversary

General Allocations Fields Events **Math** Assignment Spawn Xml Source Debug

Up Down Delete Move Into Move Out Of Comment

Include
CopyBook ENTIRE Math Section:

Math

Hierarchy	TYPE	DT	Attributes	Text	DT
NextYearsPolicyAnnualPremium	EXP	DEC		NextYearsPolicyAnnualPremium + NextYear	911
NextYearsPolicyModalPremium	EXP	DEC		NextYearsPolicyModalPremium + NextYear	911
NextYearsPolicyAnnualPremium	EXP	DEC		NextYearsPolicyAnnualPremium + PolicyFee	911
NextYearsPolicyModalPremium	EXP	DEC		NextYearsPolicyModalPremium + NextYear	911
XMLComment				Calculate Days to Maturity from Activity D	
PolicyMaturit	POLFD	DAT		MaturityDate	116
DaysToMaturity	FUNCT	INT		DaysDiffOf(ActivityDate, PolicyMaturityDate)	123

Name: DaysToMaturity

Data Type: INTEGER

Variable Type: FUNCTION

Log: Yes No

Round: Yes No Decimal places:

Call: DaysDiffOf(ActivityDate, PolicyMaturityDate)

Math Pane with Function to Call

You are here: Global Explorer and Business Rules > Plan Rules > Plan Rules

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Plan Rules

Plan rules are used to organize and capture data needed to administer policies. Fields are configured to capture the data a company needs in order to administer a specific type of plan.



If the Plan Rules folder is empty there is a possibility that the XML Schemas are null in the business rules table. In this case, upload schemas using the Upload Schema function on the tool bar. Please refer to the section on [Upload XML Schemas](#) for more information.

When a new plan is created, the following plan rules are automatically attached to the plan.

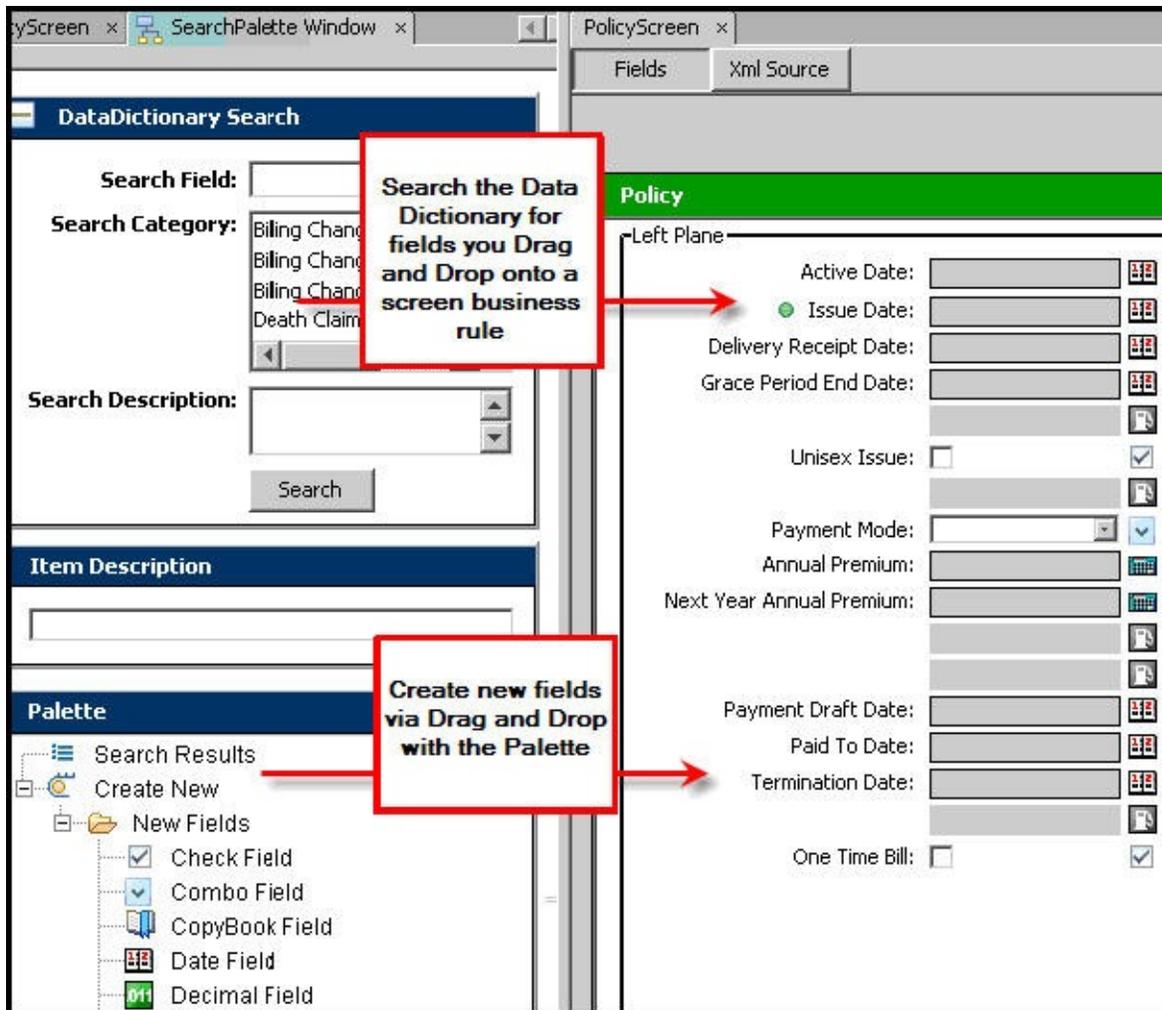
- [EligibleTransactionsByPolicyStatus](#): Controls the transactions that are available to the OIPA user depending on the [status](#) of the policy and whether in that status the transaction will be initiated by the user or the system.
- [EligibleRequirementsByPolicyStatus](#): Controls the requirements that are available to the OIPA user depending on the status of the policy, and whether in that status the transaction will be initiated by the user or the system.
- [Plan Screen](#): Defines the plan structure, such as maximum face amount, maximum age or grace to lapse days. [Plan Screen](#): Defines the configurable fields for the Plan Details.
- [PolicyAllocationScreen](#): This rule defines the allocation methods, funds and allocation models available for policy level allocations. It is used when [configuring allocations using models](#).
- [PolicyScreen](#): Defines the foundation of the contract, such as the issue date, premium mode, and free look end date.
- [PolicyValues](#): Used to display valuation on a policy and allows the user to configure other fields to run at the time valuation is executed in a transaction
- [Segment Screen](#): Defines the display of the columns in the segment

summary, which displays above the segment detail information. This is also where the ability to add new segments can be disabled, unless the segment is added by a transaction.

- [Values Screen](#): Defines the formatting of non-fixed fields on the Values screen.

When configuring any of these screen business rules, a comprehensive visual editor for field entry is available in the Rules Palette. The visual editor provides drag and drop functionality and allows the user to see the screen while editing. However, if XML is preferred, configuration can still be done in XML via the [XML Source pane](#). Configuration for the additional screen tags must still be done through the XML Source pane.

The [DataDictionary](#) or the Palette, which are located in the [SearchPalette window](#), can be used to drag and drop items into the Configuration Area.



SearchPalette Window

You are here: Configuration > Allocations > Configure Allocations > AllocationScreen Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Configure Allocations Using the Default Method

Clients upgrading from V7 and V8 who would like to maintain configuring the previous way should select the Default method when setting up a plan. Then configure the AllocationScreen business rule and/or allocations in transaction rules according to the plan requirements.



Only use the AllocationScreen business rule when performing an upgrade from v7 or v8 and do not use Fund Models or Classes. V9 introduces new rules for configuring funds and allocations for the first time or when upgrading products that use Fund Models or Classes.

Configuring Plan or Policy Level Allocations

The first step is to select **Default** as the plan allocation method when creating a new plan with the Plan Wizard. If the plan already exists, the allocation method can be edited by right-clicking on the plan in the Main Explorer and selecting **Edit Plan**. Make sure that **Default** is selected for the allocation method.

The AllocationScreen business rule defines plan, policy or segment level allocations. When configuring this rule, different allocation structures will be defined for different types of allocations at the plan, policy or segment level. Funds can limited to specific allocation structures.



When setting up allocations at the policy level, update the PolicyScreen business rule. This rule controls the allocation selection in OIPA via the Buttons pane.

Steps to Configure Allocation

1. Navigate the **Plan | Plan Rules | AllocationScreen**.
2. Check out the AllocationScreen.xml file. The configuration information in the steps below will go between the opening and closing `<AllocationScreen>` elements.
3. Add an `<Allocations>` sub-element with a TYPE attribute of Plan or Policy. Multiple allocation structures can be identified at this level.
4. Enter any restrictions, formatting, or other guidelines for the allocation structure using the following attributes in the `<Allocations>` element.
 - **AMOUNTPRECISION**: used to set the number of decimal places for the AllocationMethod(s). This is used when allocating money.
 - **UNITPRECISION**: used to set the number of decimal places for the AllocationMethod(s). This is used when allocations are unit values.
 - **PERCENTPRECISION**: used to set the number of decimal places for the AllocationMethod(s). This is used when allocations are in percentages.
 - **USEEFFECTIVEDATE**: used to display different effective dates for a

model and allows the selection of a group of allocations by model and effective date. The value Yes allows effective dates and No indicates effective dates aren't used.

- **ALWAYSEQUALPERCENT**: when Yes is selected then allocation percents will be calculated irrespective of the number of funds. When No is selected, then allocation percents will be calculated only for the even number of funds. If there is an odd number of funds, the percentage boxes will be populated with zeros.
5. Configure an actual allocation structure using the <Allocation> element.
 6. Each <Allocation> sub-element must use the TYPECODE attribute to identify the type of allocation. The value of the TYPECODE attribute is the typecode that is associated with the code located in AsCodeAllocationType. [Add the code](#) if it is missing.
 7. In the <Allocation> sub-element exclude any funds that should not be available to a user in OIPA. The following attributes are notable, but a complete list is in the XML guide.
 - **EXCLUDEFUNDSTATUS**: used to prevent the funds (that are in the specified fund status) from being available for allocation. For example: EXCLUDEFUNDSTATUS = "01,03" indicates that the funds that are in these statuses should not be available for allocation. 01 & 03 may indicate active and closed fund status, respectively.
 - **EXCLUDEFUNDTYPE**: used to identify fund types (comma separated codes) that should be excluded from the Allocation Fund drop down list.
 - **FUNDLIMIT**: used to set the limit on the number of funds that can be made available for allocation.
 - **ALLOWMIXEDMETHOD**: used to allow different method codes to be specified for each allocation.
 8. Add an <AllocationMethods> sub-element. In the value, enter the codes for the specific allocation methods that will be used. The codes are found in AsCodeAllocationMethod. For example, percent, amount or units.
 9. Check in the configuration to save the changes.

Configure Allocations that Transfer Money from One Fund to Another

Allocations can be configured to move money from one fund to another. Use the **<AllocationTransfer>** element along with the **<AllocationFrom>** and **<AllocationTo>** sub-elements to configure this feature. The **XML Configuration Guide** provides detailed explanations of all configuration syntax. It is located in the Help menu in the Rules Palette.

Schema Example

```
<AllocationTransfer REPEATFUNDS="[Yes|No]">  
<AllocationFrom EXCLUDETYPE="[FundTypeCode]" PERCENTPRECISION="0.01">  
<AllocationTo EXCLUDETYPE="[FundTypeCode]" PERCENTPRECISION="0.01">  
</AllocationTo>  
</AllocationFrom>  
</AllocationTransfer>
```

Disabling the Allocation Structure Fields By Policy Status

Allocation structures can be locked down after the policy moves into a certain status. For instance, if a policy is in canceled or death status then allocations cannot be set. Allocations can be disabled for a plan according to policy status or at the individual allocation level.

To disable allocations for a plan, configure the **<DisableAllocationFields>** element and **<DisablePolicyStatus>** sub-element. The value of **<DisablePolicyStatus>** should be the policy status codes where allocations are disabled.

Schema Example

```
<DisableAllocationFields>  
<DisabledPolicyStatus>CodeValue, CodeValue</DisabledPolicySt.  
</DisableAllocationFields>
```

Transactions that Use Allocations

Configuring transactions that use allocations uses a different syntax. The allocation structure is still created, but there are additional tags for moving money in funds to other funds and setting default allocations. Also transaction rules have an Allocations pane where the configuration is performed. Refer to the [Transactions That Use Allocations section for additional information](#).

Configuring Allocations for Fund Models

This is not backwards compatible from v7 and v8. This must be done using the Allocation Model method.

Configuring Allocations for Asset Classes

This is not backwards compatible from v7 and v8. This must be done using the Allocation Model method.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Global Explorer and Business Rules > Plan Rules > EligibleRequirementsByPolicyStatus
Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

EligibleRequirementsByPolicyStatus Rule

The **EligibleRequirementsByPolicyStatus** rule is a plan rule that enhances and supports the processing of events in the system. This rule can only be [configured using XML](#).

The EligibleRequirementsByPolicyStatus business rule controls the requirements that are available to the OIPA user depending on the status of the policy and whether in that status the transaction will be initiated by the user or by the system.

Statuses

The statuses are defined in the AsCode database table. New statuses can be added or existing statuses can be changed using the [Code Names editor](#) in the Administration folder of the Admin Explorer.

Note: When configuring a shadow status, the status should be configured to require that no activities, no pending or recoverable disbursements, no roles and no segments exist on the policy.

General Pane

The **General pane** displays the rule properties. The fields on the General pane include:

- Version: the version of the rule.
- Type: the type of rule. In this case it is a system rule.
- Long name: the complete name for the rule.
- Keywords: any words that might be helpful for locating the rule during a search.
- Comments: any comments that might explain the function of the rule.
- Error message: any pertinent error messages.

Configure Using XML Source Pane

A requirement's availability by status can be updated or removed from the XML Source pane. The requirement's name must be the value between the <Requirement> start and end tags. All <Requirement> tags must be sub-elements of the <PolicyStatus> tag. The requirements are also identified as system generated or user generated with the TYPE attribute. The <PolicyStatus> element has an attribute called NAME that lists the name of the policy status that the policy must be in for the requirement to be available. The NAME attribute is only used for readability. The CODE attribute controls in what policy status the requirement will be available. CODE values are available in the [Codes Names](#) folder in Admin Explorer.

Important: Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Plan Rules | EligibleRequirementsByPolicyStatus**.

You are here: Global Explorer and Business Rules > Plan Rules > EligibleTransactionsByPolicyStatus Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

EligibleTransactionsByPolicyStatus

The **EligibleTransactionsByPolicyStatus** rule is a plan rule that enhances and supports the processing of events in the system. This type of rule can be [configured using XML](#) or it can be [configured using visual editing tools](#).

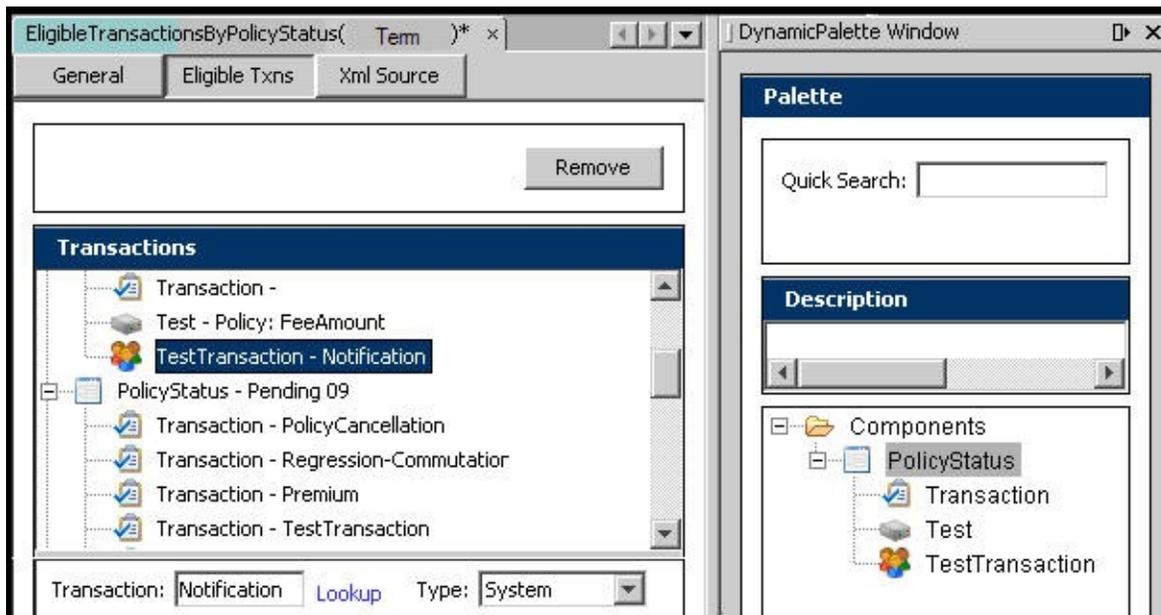
The EligibleTransactionsByPolicyStatus business rule controls the transactions that are available to the OIPA user depending on the status of the policy and whether in that status the transaction will be initiated by the user or by the system. For example, the Issue activity that issues a policy is available only when the policy is in pending status. If the policy was already in active status, there would be no reason to have the Issue activity available, as the policy has already been issued.

Statuses

The statuses are defined in the AsCode table. New statuses can be added or existing statuses can be changed using the [Code Names editor](#) in the Administration folder of the Admin Explorer.

Any transaction that needs to be available in all policy statuses must be added to each status individually. For example, the StatusChange transaction needs to be available regardless of the status a policy is in. Therefore, it is added to each status listed in the EligibleTransactionByPolicyStatus rule so that OIPA makes it available to the user regardless of the policy status.

Note: When configuring a shadow status, the status should be configured to require that no activities, no pending or recoverable disbursements, no roles and no segments exist on the policy.



Eligible Transaction Pane of Business Rule

The **General pane** displays the rule properties. The fields on the General pane include:

- Version: the version of the rule.
- Type: the type of rule. In this case it is a system rule.

- Long name: the complete name for the rule.
- Keywords: any words that might be helpful for locating the rule during a search.
- Comments: any comments that might explain the function of the rule.
- Error message: any pertinent error messages.

The **Eligible Txns Pane** allows for the visual editing of this rule. Drag and drop from the Components listed in the Palette window to the Transactions section of the Configuration Area.

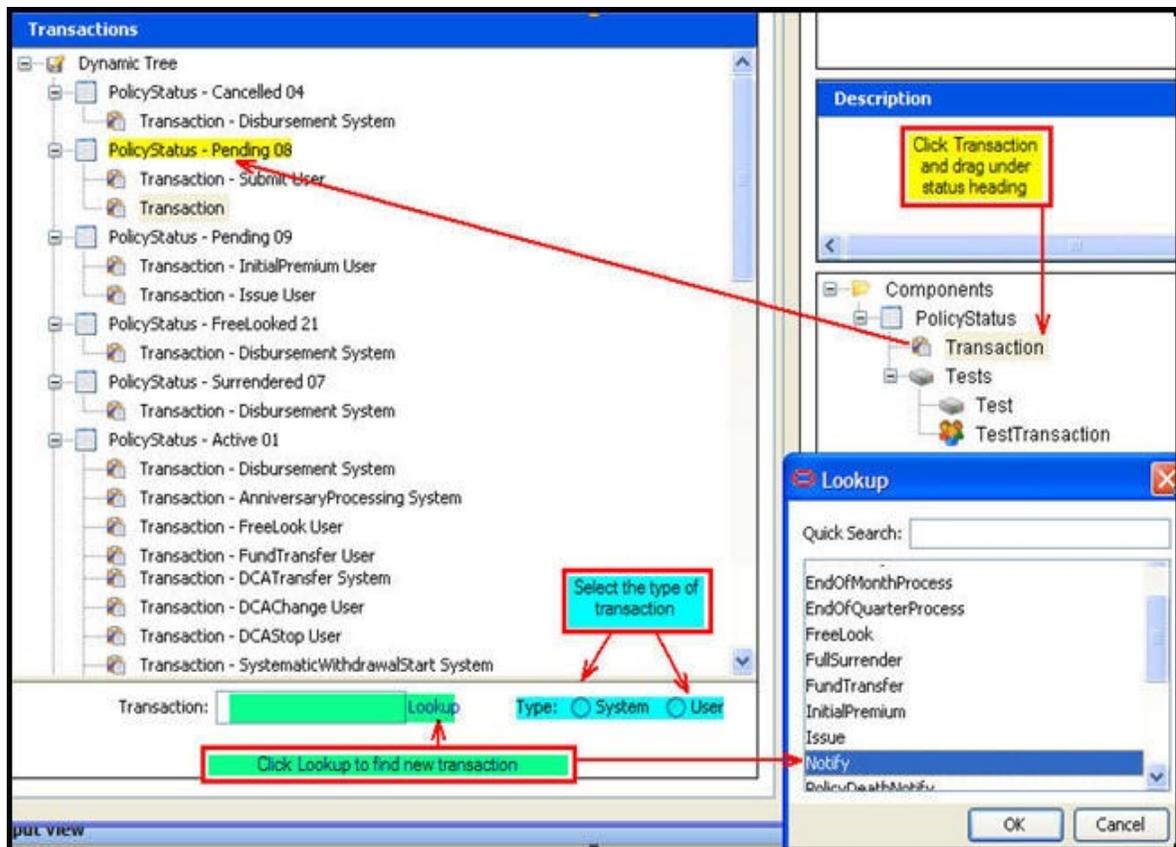
- PolicyStatus can be dragged onto the Transaction section to create a new policy status. Enter the code value and name of the policy status manually or use the Lookup link to select one from records in AsCodeStatus. If a [new policy status code](#) is entered, then it is important to make sure it exists in AsCodeStatus.
- Transactions can be dragged onto any PolicyStatus in the Transactions section. Identify the transaction that should be available in the associated policy status either by entering the name manually or using the Lookup link to select the transaction from the AsTransaction table.

Note: System transactions are added to the Activity screen in OIPA only by [spawns](#) and will not be available from a drop-down list. User transactions can be added manually (through drop-down list selection) or through spawns.

- Test should be used in conjunction with the TestTransaction component. Test is available to validate a screen field against a value in a transaction. Write the validation in the Test field. There may be multiple Test validations per TestTransaction.
- TestTransaction is used in conjunction with Test and identifies the transaction that will be available if the test condition result is true.

Steps to Use Visual Editing to Add a Transaction to the EligibleTransactionByPolicyStatus Business Rule

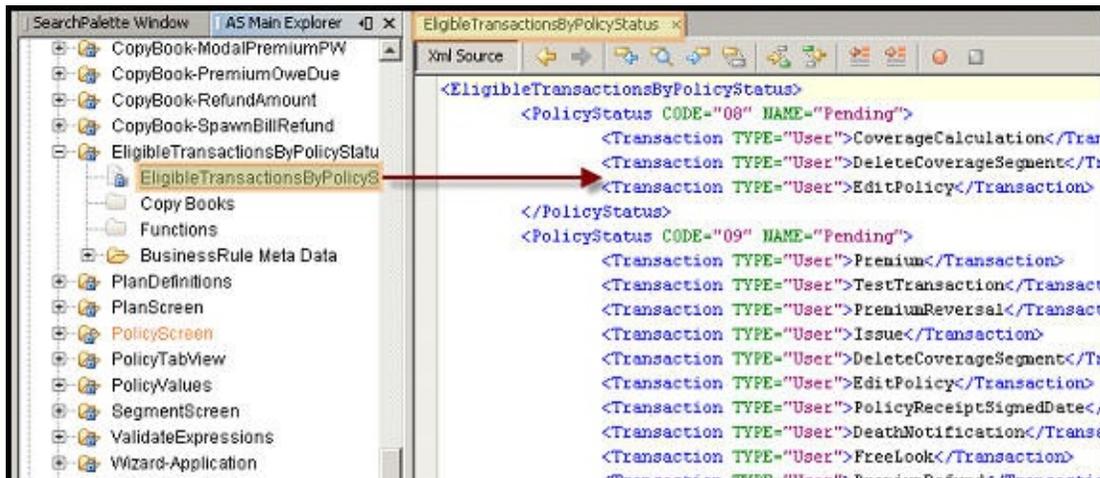
1. Navigate to the **Main Explorer** tab and open the Company and Plan folders where the new transaction resides.
2. Open **Business Rules | Plan Rules** and double-click on the EligibleTransactionByPolicyStatus folder. Right-click on the XML file and select **Check-out**.
3. Click the **Eligible Txns** tab. This will open the window in the Configuration Area where the new transaction can be added using drag and drop features.
4. Locate the Policy Status in the dynamic tree. If the policy status does not exist, then click **PolicyStatus** in the Components window on the right and drag it under the dynamic tree. Click the **Lookup** link next to the Code box under the dynamic tree window and select a policy status.
5. Click **Transaction** from the right-side window and drag it under the appropriate policy status heading.
6. From the bottom of the Configuration Area, click the Lookup link to find the name of the new transaction. When the Lookup window appears, highlight the new transaction name and click **OK**.
7. Click the radio button for **user** to identify the new transaction as user generated or click the radio button for **system** to identify the new transaction as system generated.
8. Check-in the EligibleTransactionByPolicyStatus business rule to save the changes to the database.



Drag and Drop New Transaction

Configure Using XML Source Pane

A transaction's availability by status can be updated or removed from the XML Source pane. The transaction's name must be the value between the <Transaction> start and end tags. All <Transaction> tags must be sub-elements of the <PolicyStatus> tag. The transaction is also identified as system generated or user generated with the TYPE attribute. The <PolicyStatus> element has an attribute called NAME that lists the name of the policy status that the policy must be in for the transaction to be available. The NAME attribute is only used for readability. The CODE attribute controls what policy status the transaction will be available in. CODE values are available in AsCodeStatus, which are in the [Codes Name](#) folder in Admin Explorer.



EligibleTransactionByPolicyStatus XML in XML Source Pane

Important: Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Plan Rules | EligibleTransactionsByPolicyStatus**.

You are here: Global Explorer and Business Rules > Plan Rules > PlanScreen Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

PlanScreen Rule

The PlanScreen rule has four panes available for configuration. The General, Fields, and Events panes offer visual configuration editing tools. The XML Source pane can be used to configure the Plan screen using XML. This rule defines plan values that can be accessed by other business rules.

PlanScreen General Pane

The General pane displays the name and properties of the Plan screen rule.



PlanScreen General Pane

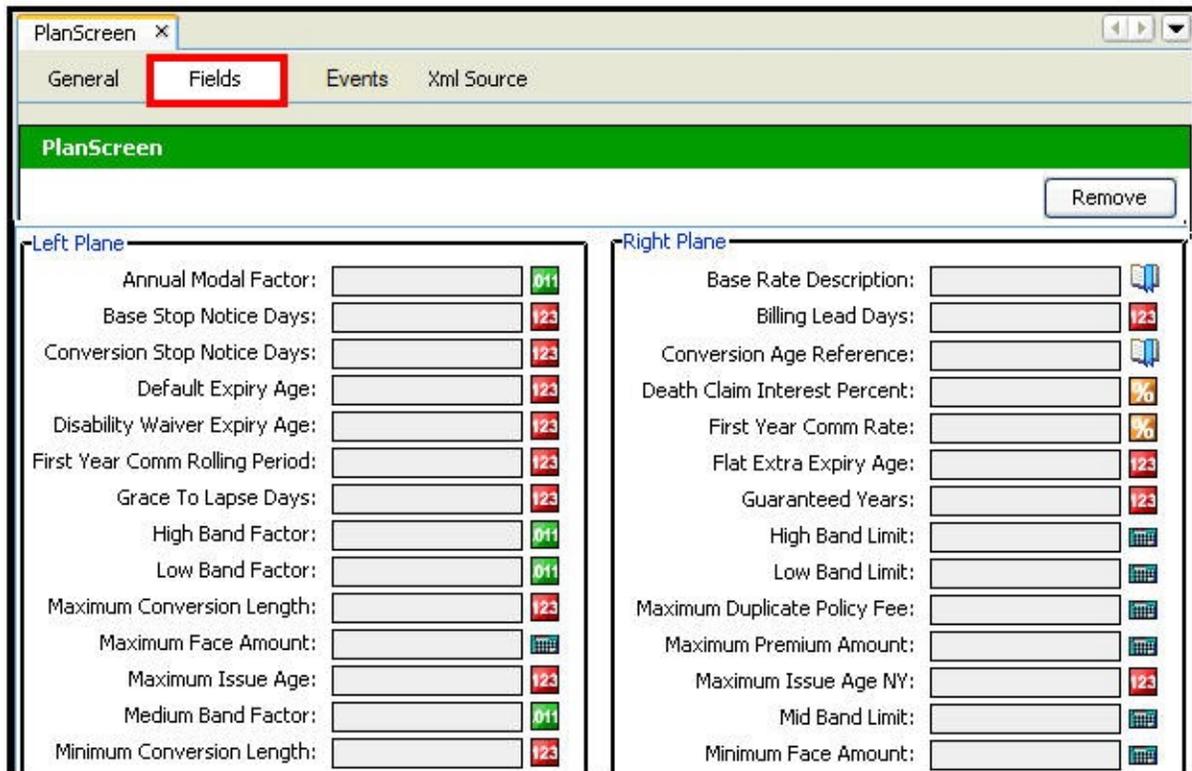
A miscellaneous section at the bottom of the General pane allows the user to control whether the plan supports state approvals. Select **Yes**, if state approvals are supported on the plan.



Use State Approval Option on General Pane of PlanScreen Rule

Fields Pane

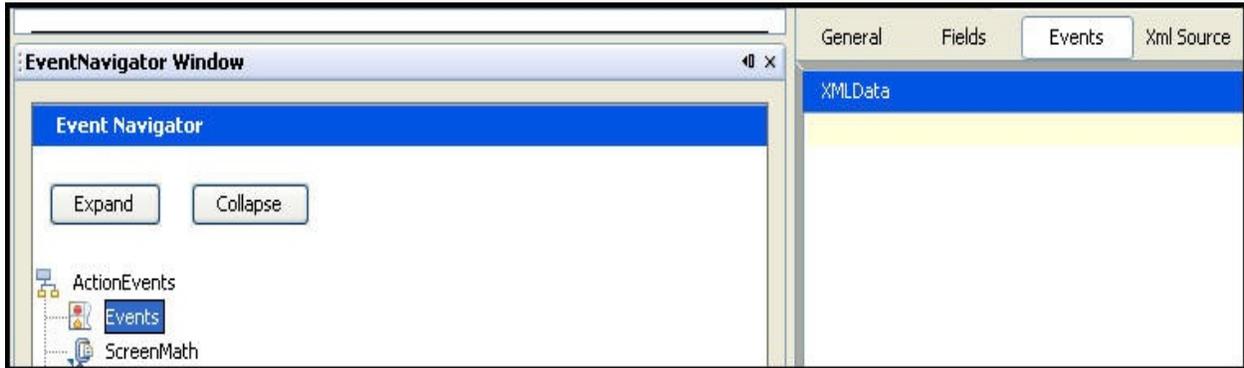
This pane is used to create fields that will display on the Plan screen. The functionality is exactly the same as the Fields pane described in the transaction section. Please see the [Fields Pane](#) for more information.



PlanScreen Fields Pane

Events Pane

This pane is used to create validations that will display on the Plan screen. The functionality is exactly the same as described in the transaction section. Please see [Events](#) for more information.

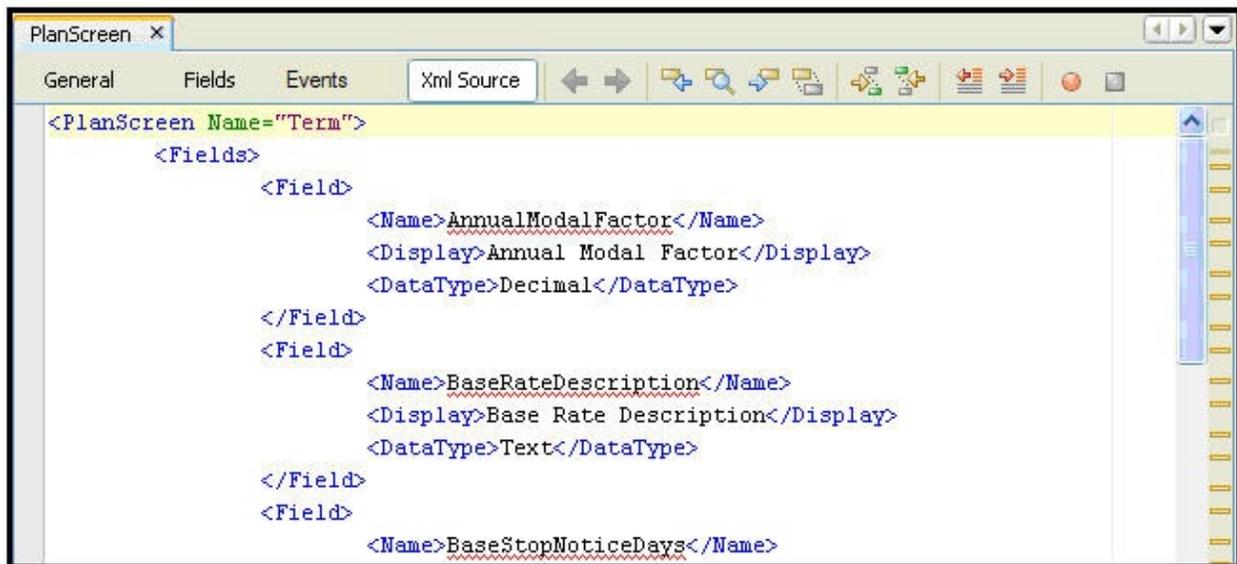


PlanScreen Events Pane

XML Source Pane

Configuration can also be done in XML via the XML Source pane. The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the [XML Source Editing](#) section. Configuration for the additional screen tags must still be done through the XML Source pane.

Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Plan Rules | Plan Screen**.



PlanScreen XML Source Pane

You are here: Group Information > Group Customer Plans > Plan Screen

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Plan Screen

The PlanScreen business rule defines the elements and field values associated with a selected plan. Since plans are linked to a child product or product, this rule supports the sharing of elements between the product, child product and plan.

The PlanScreen rule can be overridden at any combination of product, child product and plan levels. When multiple overrides exist, OIPA will build a single unified rule from the multiple overrides. A single fields section containing all of the fields defined in each of the rules will be built and displayed on screen. When there are conflicts, meaning a field with the same name is defined in multiple instances of the rule, the field defined at the lowest level in the hierarchy takes precedence. The hierarchy is as follows: Product > Child Product > Plan where “best match” starts at plan and works backwards (i.e. from bottom up) in completing the rule/fields as applicable.

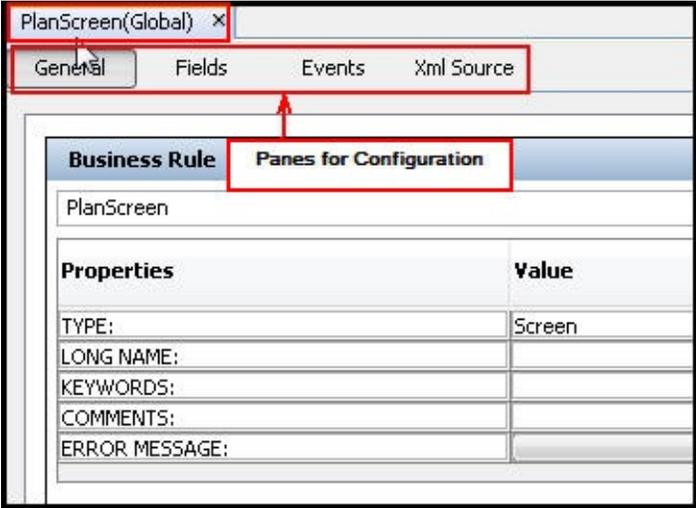
The ability to override plan screens fields by reconfiguring fields with the same name, and when doing so, the Fields pane also has the ability to show the inherited field and the overridden field

The ability to prevent a field that is configured in multiple override levels from being overridden is available. For configured fields, the new <**Final**> element prevents a field from being overridden at a lower level when set to **Yes**. When this element is not present, the default behavior will be **No**.

Please see the **V9 XML Configuration** topic in the **Help** menu for a list of all elements, attributes and values needed for configuration. View **Business Rules | Screen Rules | PlanScreen**.

General Pane

The Plan Screen General pane displays the name of the rule, the properties of the Plan Screen screen and their associated values.



Plan Screen With Configuration Panes Highlighted

Fields Pane

The Fields pane is used to create fields that will display on the Class Group screen. The top section of the visual configuration area is for fixed fields. Drag and drop the available fixed fields from the Palette. Custom fields cannot be created for this section, but the display names of fixed fields can be changed as needed. The display names of the Fixed Fields is controlled by AsTranslation. Configure dynamic fields in the section under fixed fields. The functionality is exactly the same as the configuration for other fields panes. Please see the [Fields Pane](#) for more information.

Events Pane

This pane is used to create validations that will display on the Plan screen. The functionality is exactly the same as described in the transaction section. Please see [Events](#) for more information.

XML Source Pane

Configuration for this rule is done in XML via the XML Source pane. The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the [XML Source Editing](#) section.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Global Explorer and Business Rules > Plan Rules > PlanCoverageScreen_Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

PlanCoverageScreen_Rule

The specific level or subset of Plan benefits / features available for election and enrollment by a Participant is a Sub-Plan, also known as Plan Coverage.

A Sub-Plan can be attached to one or more Class within a Class Group as well as remove associations between a Sub-Plan and Class.

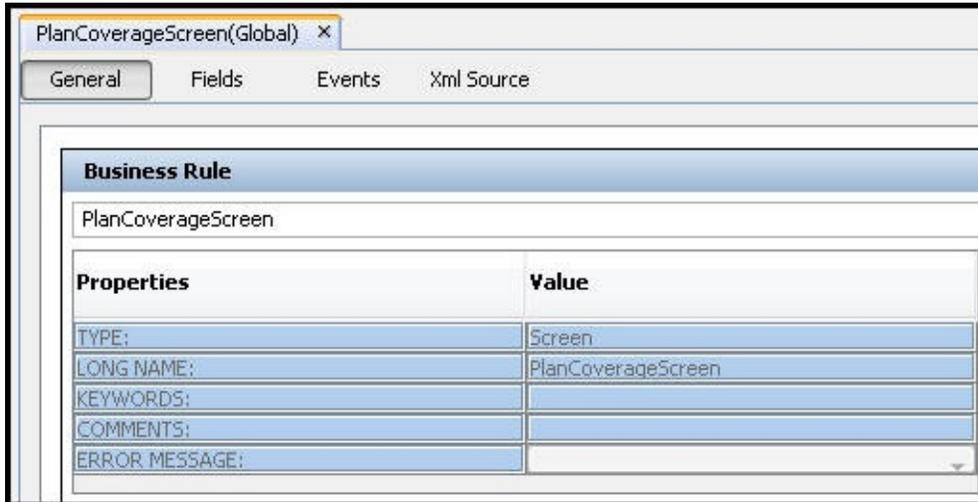
When attaching a new Sub-Plan to a class, you will be presented with the PlanCoverageScreen in order to view eligible Sub-Plans based on Effective Date. Any Sub-Plans already attached to the Class are excluded from the listing.

You can view the Plan Coverage Name and Effective Date within the listing. You can filter the listing by Effective Date.

You will be able to select a checkbox next to each Sub-Plan(s) to associate to the Class and Active From Date to indicate the start of the association. An **OK** button will confirm the selections and implement the attachment of the Sub-Plan(s) to the Class. Any Sub-Plans that will be associated to a Class with varying Active From dates will need to be selected individually.

General Pane

The General Pane displays the name and properties of the PlanCoverageScreen business rule



PlanCoverageScreen General Pane

XML Source Pane

Configuration can be done in XML via the XML Source pane. Configuration for the additional screen tags must still be done through the XML Source pane.

The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the [XML Source Editing](#) section of this document.

Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Screen Rules | PlanCoverageScreen**.



PlanCoverageScreen XML Source Pane

Miscellaneous Screen Details: Security

Page: Plan Coverage
Buttons: Save, Cancel

You are here: Global Explorer and Business Rules > Plan Rules > PolicyScreen Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

PolicyScreen Rule

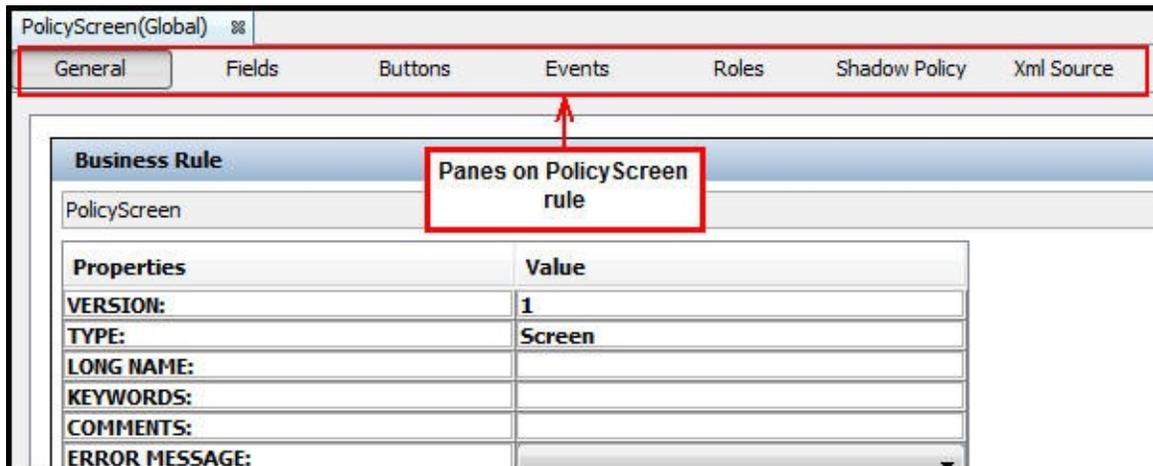
This business rule controls the layout and functionality of OIPA's Policy screen, which is used to define the foundational information on which a policy is based. PolicyScreen can also be used to configure OIPA's Application screen by using a system code-level override. The policy summary screen is also introduced in the policy overview screen to make it consistent with other policy related pages and compatible with prior versions. In addition, prior versions displayed the policy summary with all of the fixed fields presented in the same order and format as they present on the PolicyScreen. This is to provide a default presentation of the policy summary for all policy related pages except the PolicyScreen which retains the current fixed field and dynamic field presentations alone and no summary. In addition, configurability for policy summary will be added so each customer may specify the fixed fields and presentation order they wish.

Configuration Panes

The **PolicyScreen** rule has seven panes available for configuration and is a required rule for a plan to run in OIPA. The [General](#), [Fields](#), [Buttons](#) and [Events](#) panes offer visual configuration editing tools.

There are also two unique visual configuration panes: Roles and Shadow Policy. [The Roles](#) pane is used to define the roles that are available to assign to a specific plan. The [Shadow Policy](#) pane defines the criteria OIPA uses when shadowing (or canceling) a policy.

The final pane, the XML Source pane, can be used to configure the Policy screen using XML.



PolicyScreen Panes for Configuration

General Pane

The General pane displays specific properties that are configurable for the Policy screen. The Policy screen General pane Segments section lists the properties and an editable value field to use for configuration.

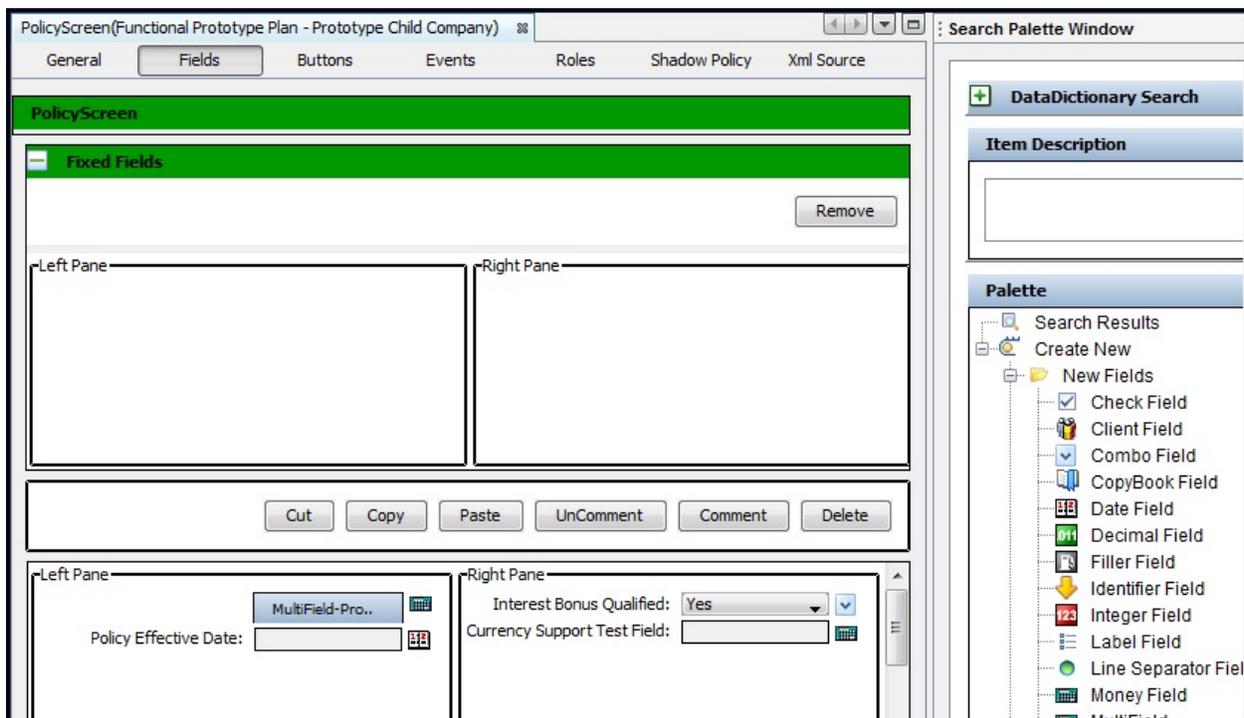
Definitions for the Policy Screen General Pane

Property	Definition
Segments Allowed	Specifies the number of segments that are allowed on the policy.
Minimum Segments	Specifies the minimum number of segments that are required for the policy.
ShowSegmentRoles	Yes allows Segment roles to display on the Segment tab of the policy and in the Client drop down box. No restricts the Segment roles from displaying on the Segment tab of the policy or in the Client drop down box.
Automatic Policy Number	System will generate an automatic policy number.
Disabled Status	Indicates which applicable status(es) will disable otherwise editable fields.
Segment Count	Total number of segments that are assigned to a policy.

Fields Pane

The Fields pane is used to create fields that will display on the Policy screen. The top section of the visual configuration area is for fixed fields. Drag and drop the available fixed fields from the Palette. Custom fields cannot be created for this section, but the display names of fixed fields can be changed as needed. Configure dynamic fields in the section under fixed fields. The functionality is exactly the same as the configuration for other fields panes. Please see the [Fields Pane](#) for more information.

Note: The Cut, Copy and Paste buttons available in Fields and Math panes are not supported for the PolicyScreen's Fixed Fields.



PolicyScreen Fields Pane

Buttons Pane

The Buttons pane allows the selection of the type of button displayed on the Policy screen.

The Preview section along the bottom of the Button Pane provides a preview of the appearance of the buttons in OIPA.

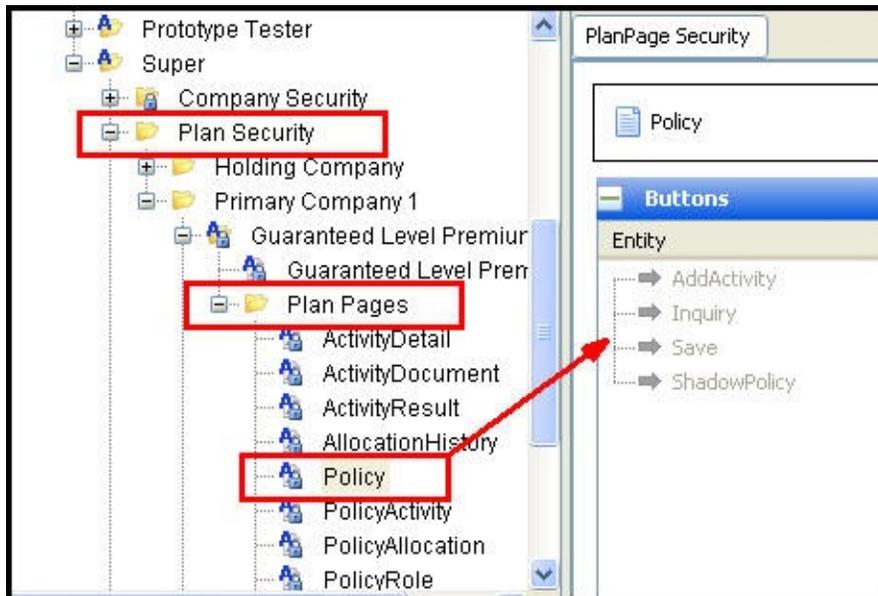
The Order section along the right side of the Button Pane allows the user to control the order the buttons will display in OIPA.

The following buttons are supported on the Policy screen.

- Activity
- Add
- Allocate
- Calculate
- Close
- Find
- Inquiry
- New
- Policy Overview: this is the first option on the Left Navigation menu of the Policy screen in OIPA when this button is added. The [PolicyOverviewScreen business rule](#) must also be configured.
- Process
- Save
- Shadow Policy: this is configured through the [Shadow Policy pane](#).
- Values
- Withholding

The Allocation, Values, Activity and Withholding buttons are configured through the Button pane on the Policy screen business rule in the Global Explorer. The Save, Inquiry and Add Activity buttons will only become visible on the Policy screen if security is set for them through the Admin Explorer. The Shadow Policy button will only become visible if the

Shadow Policy section is configured and security is applied through the Admin Explorer.



PolicyScreen Button Security in Admin Explorer

Button Security

Security is applied to the buttons on the Policy screen in the Admin Explorer. Open the Security folder to reveal the options for applying security.

Steps to Add Security to Buttons

1. Open **Admin Explorer | Security | Application Security | Security Groups**.
2. Select the Security Group that has access to the Policy screen.
3. Open **Plan Security | Name of the Company | Plan Pages**.
4. Right-click the Policy node and select **Check-out** to open it in the Configuration Area.
5. Check the box next to each button the users in the security group will have access to in OIPA.
6. Check-in the file to save the changes to the database.



The Save, Inquiry, and Add Activity buttons are dependent on security and also on what has been entered for the policy. The security group has four buttons for the Policy screen. If all of these buttons are checked in the security group, then before the Policy screen is saved in OIPA, only the Save button are present. Once it has been saved, then the Inquiry Screen button is present if a policy level Inquiry screen has been configured. The AddActivity button is present after the segment is calculated if the MinimumSegments configuration tags are present and the value is greater than zero. The Shadow Policy button will only become visible if the Shadow Policy section is configured and security is applied through the Admin Explorer.

Events Pane

This pane is used to identify the type of event that needs to occur in order to invoke a set of actions. It also identifies the field that is connected to the event. The functionality is exactly the same as described in the transactions section. Please see [Events](#) for more information.

Shadow Policy Pane

This pane is used to set up conditions for shadowing (deleting) a policy. It allows a user to identify the policy statuses that are valid for shadowing, as well as the status a shadowed policy is assigned. Please see [Shadow Policy Pane](#) for more information.

Roles Pane

The Roles pane is used to add, edit, or remove roles that are associated with a policy.

Steps to Add a New Role

1. Select the **Roles** pane.
2. Right-click on the Roles node in the Navigator menu and select **Add Role**.



Role Node in Navigator Window

3. Select the desired role name from the available list in the **Role Code** drop-down box.



Role options are populated by AsCodeRole, which can be updated through the [Codes Names](#) folder in Admin Explorer.

4. Select **Yes** or **No** in the **Allow Zero Percent** drop down box. If Yes is selected, then the Allow Percent drop down box will be disabled.
5. Select **Yes** or **No** in the **Allow Percent** drop down box. If No is selected then the AllowZeroPercent box is disabled.
6. Enter a role count in the **RoleCount** field if desired. Role count is the maximum number of times a role can be assigned to a particular policy.
7. If applicable, select a **ClientType**. This selection defines the client types that can be assigned to the role. Only these client types will be displayed in the client type drop-down box on the Client screen for the specific role. For example, an insured may need to be an Individual client type for a particular plan.
8. Enter a role percent in the **Role Percent** field. Percentages can range from 0% to 100%, or an asterisk ("*") can be used to indicate

that the total percent for the role can exceed 100%. Each individual record will still have a maximum of 100%, but the total for multiple records on the same Policy Role will be able to exceed 100%.

Using an asterisk in the Role Percent field will generate a warning message at the bottom of the Roles pane.

4. Specify whether the role has an external client by selecting the **Yes** or **No** radio button in the External Client field. An external client is a client whose data is stored in an external database that cannot be modified within OIPA.
 - **Yes:** Indicates that client data is stored in an external database. If this option is selected, the EXTERNAL attribute with a value of "Yes" is added to the <Role> element in the XML source. Additionally, if the CustomScreen field is populated, then the Client Type field can be left blank. If the CustomScreen field is not populated and no Client Type is chosen, then the system will generate an error message upon saving the role. These conditions apply to both visual and XML source configuration.
 - **No:** The default selection—indicates that client data is not stored in an external database, and can be modified within OIPA. The Custom Screen and External Client Row Retriever fields will be disabled, and a client type must be chosen, or the system will generate an error message upon saving the role. This condition applies to both visual and XML source configuration.
5. If the External Client radio button is set to Yes, type a class name in the External Client Row Retriever field. This should be the name of a class that has been created to implement the external client search.
6. If the External Client radio button is set to Yes, and if applicable, type a class name in the Custom Screen field. This should be the name of a class that has been created to implement a customized display of external client search results. Note: Refer to the Extensibility document on OTN in the library for this release to access additional information on extending OIPA to include external clients.
7. If applicable, enter an expression in the **Test** field. This allows

configuration of a test condition. The evaluation of the condition determines whether the particular role section can be executed or not. Enter an expression to determine if the role should be displayed/allowed for the policy.

8. Select the **Save** button at the bottom of the Roles pane when finished.



To add multiple new roles, select **Add Role** from the Navigator window before each new role is entered. This clears the fields. **CTRL** + click deselects the options if they are already selected in a drop down list.

Steps to Remove a Role

1. Click the **Roles** pane.
2. Click the role to be removed from the Roles section of the Navigator window. The role will appear highlighted in blue.
3. Right-click the role and select **Delete**.

Steps to Edit a Role

1. Select the **Roles** pane.
2. Select a role from the Navigator window. The role will appear in the Configuration Area.
3. Make the necessary changes.
4. Click **Save** at the bottom of the Roles pane.

XML Source Pane

The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the [XML Source Editing](#) section.

Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Plan Rules | Policy Screen**.



The screenshot shows a software interface with a menu bar at the top containing 'Fields', 'Buttons', 'Events', 'Shadow Policy', 'Roles', and 'Xml Source'. The 'Xml Source' tab is selected and highlighted with a red box. Below the menu bar, the XML source code is displayed in a text area. The code is as follows:

```
<PolicyScreen NAME="PolicyScreenTerm">
  <Fields>
    <Field>
      <Name>ActiveDate</Name>
      <Display>Active Date</Display>
      <DataType>Date</DataType>
      <Disabled>ReadOnly</Disabled>
    </Field>
    <Field>
      <Name>StatutoryCompany</Name>
      <Display>Statutory Company</Display>
      <DataType>Combo</DataType>
      <DefaultValue>ALIC</DefaultValue>
      <Disabled>Yes</Disabled>
      <Query TYPE="FIXED">
        <Options>
          <Option>
            <OptionValue:

```

PolicyScreen XML Source Pane

Use for Applications

In addition to its use in configuring the Policy screen, the PolicyScreen business rule can also be used to configure the Application screen in OIPA. To configure the Application screen, the PolicyScreen rule needs to be overridden at the system code level, using the NBU (New Business Unit) system code. See [Creating the Application Screen](#) for more information.

In OIPA, the Application screen is accessible through the **Application** node in the menu on the left side of the Case screen.

You are here: Global Explorer and Business Rules > Plan Rules > PolicyValues Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

PolicyValues Rule

The **PolicyValues** business rule is used to display valuation on a policy and allows the user to configure other fields to run at the time valuation is executed in a transaction. Fields are also displayed on the PolicyValues screen.

Note: If fields are only needed for display purposes, it is advisable to use the [Inquiry](#) screen instead.

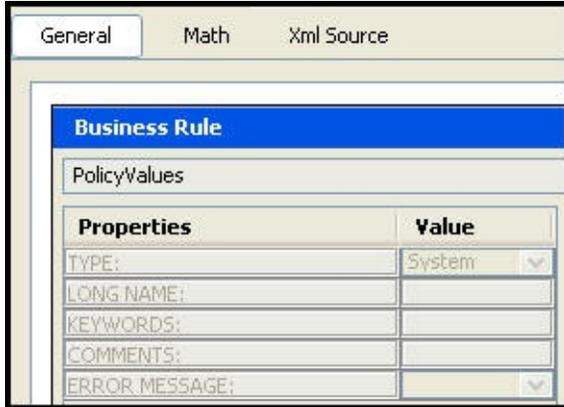
Other rules may reference the policy value math variables in this rule in order to obtain the policy values for displaying or processing. This rule may be configured to perform the actual policy value calculation or may pull the policy value from the valuation results. If this rule is not configured at the plan level, the system will try to use global rules using the same best-match lookup that is used for other rules.

Rules that are used in conjunction with the PolicyValues business rule:

- [ValuesScreen](#) business rule, which is used to format the Values screen, can display the policy value using the value stored in the PolicyValues business rule. Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Plan Rules | Values Screen**.
- **Transaction rules** can execute the PolicyValues business rule if the valuation tag is attached and the attribute policy value is set to **Yes**.

PolicyValues General Pane

The General pane displays the name of the rule and the properties.



PolicyValues General Pane

PolicyValues Math Pane

The Math pane is used to calculate the policy value(s). For further information on how to use the Math pane, see the [Math Pane](#) section.

The screenshot displays the 'PolicyValues(Math)' pane and the 'SearchPalette Window'. The 'PolicyValues(Math)' pane has tabs for 'General', 'Math', and 'Xml Source'. The 'Math' tab is active, showing a table of variables and their properties.

Hierarchy	TYPE	DT	Attributes	Text	DT
Blank	VALUE	TXT			Txt
Yes	VALUE	TXT		Yes	Txt
No	VALUE	TXT		No	Txt
Zero	VALUE	INT		0	INT
Checked	VALUE	TXT		CHECKED	Txt
Segment...SQL	SQL	TXT		SELECT AsSegment	Txt
PolicyCa...	FIELD	DEC		Valuation:Policy:Ca	DEC
Contract...	FIELD	DEC		Valuation:Policy:Ca	DEC

Below the table, there are input fields for 'Name' (Blank), 'Data Type' (TEXT), 'Variable Type' (VALUE), 'Log' (Yes/No), 'Round' (Yes/No), and 'Value'.

The 'SearchPalette Window' is on the right, containing a 'Search Description' field, an 'Item Description' field, and a 'Palette' section with a tree view of search results.

PolicyValues Math Pane

XML Source pane

The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the [XML Source Editing](#) section.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Global Explorer and Business Rules > Plan Rules > RoleScreen Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

RoleScreen Rule

The Role screen rule allows the configuration of fields specified on the Role Details screen in OIPA. Each role, such as beneficiary, insured, owner or payer, can have configured fields for capturing relevant data required by the business.



If you want to change the roles that are available to users in OIPA for a particular company, then you must edit the Roles section of the [PolicyScreen](#) business rule.

Note: In the RoleScreen action/events/screenmath sections, all Plan, Policy, Client and Address details are accessible through PLANFIELD, POLICYFIELD, CLIENTFIELD and ADDRESSFIELD type math variables.

General Pane

The General Pane displays the name and properties of the RoleScreen business rule.

RoleScreen Pane

Roles are managed through the RoleScreen pane of the RoleScreen business rule. The roles that are currently available will be displayed on the left side of the screen in the navigation tree. Click on a node to open it in the Configuration Area. There are three nodes under each rule:

1. **Ordinal:** An optional element that contains the Order field. In the Order field, a user can enter an integer value specifying the order in which roles should display on the policy's Role screen in OIPA. For example, a role with an Order of 01 will be the first role displayed, a role with an Order of 02 the second, and so on. The Rules Palette will return an error if a negative or non-integer value is entered into the Order field.
2. **Fields:** An element that allows for visual configuration of fields on a policy's Role screen in OIPA.
3. **Events:** An element that allows for the configuration of ActionEvents, which provide the ability to perform actions on a field when an event occurs on the screen. This node functions the same way as other configurable ActionEvents—see the [Events Pane](#) page for more information on configuring events.

Steps to Add a New Role

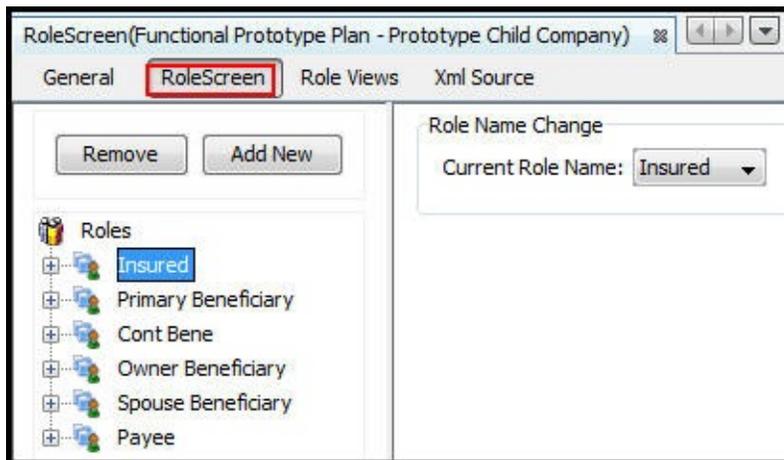
1. Navigate to the **Global Rules Explorer | Business Rules | Plan Rules** folder. Open the RoleScreen business rule, taking care to find the appropriate override level.
2. Click the **RoleScreen** pane.
3. Click **Add New**. The New Role field will display in the right-hand side of the pane.
4. Select a role from the drop-down box. This box is populated with roles from AsCodeRole.
5. Click the **Add New** button below the drop-down box. The new role will display in the left navigation tree.
6. Expand the new role to view the subnodes.
7. Click **Ordinal** to assign a display order to the role. This determines its

position in the list when available roles are displayed in OIPA.

8. Click **Fields** to add fields to the Role screen for this role.
9. Click **Events** to configure events for this role.

Once roles have been added to the Role screen, they are available for use in OIPA.

A role can be removed by selecting the role and then clicking **Remove**.



RoleScreen Pane on the RoleScreen Business Rule

Role Views

Roles can be grouped into specific views for display in OIPA. When a role view is selected in OIPA, only those roles assigned to that view will be visible to the user.

Role views can be created using the Role Views pane, or they can be created directly in the XML Source Pane. Refer to the XML Configuration Guide in the Help menu of the Rules Palette for additional details on the elements and attributes supporting views. Navigate to **Business Rules | Screen | RoleScreen**.

Steps to Manage Role Views Using the Role Views Visual Editing Pane

1. Navigate to the **Global Explorer | Business Rules | Plan Rules** folder. Open the RoleScreen business rule, taking care to find the appropriate override level, and check out the rule.
2. Click the **Role Views** pane. The **Navigator** window will display in the bottom left corner of the screen, directly below the Global Explorer window.
3. Click the **Display Status** node in the Navigator window. The role statuses will display in the Role Views pane. Select the statuses a role must be in to be eligible for a role view. Hold down the Ctrl key to select multiple statuses. The Display Status section is optional, so if no display statuses are selected, then role views will be eligible for all role statuses.
4. Click the **Role Views** node in the Navigator window. All existing role views will display under the node.
5. To delete an existing role view, right-click on the role and select **Remove Role View**.
6. To add a new role view, right-click on the Role Views node and select **New Role View**. This will open the Role Views pane.
7. Type a name for the role view in the View Name field. Acceptable name values can be any combination of text and numbers. The role view name is a required item and if it is not entered then a business error will be

produced.

8. Select the roles that will display when the view is selected in OIPA. This drop-down box is populated with the roles that are defined in the <Role> section of the RoleScreen business rule. At least one role must be selected. Hold down the Ctrl key to select multiple roles for the view. If all roles will be available in this view, then click the **All Roles** checkbox above the Roles drop down list.
9. Type the optional Table XML data in the box provided for that purpose.
 - <Table> is the opening/closing element for the table. This is the table in OIPA that displays the role information.
 - <Column> is the opening/closing element for each column in the table.
 - <Display> defines the heading for each column that will appear in the table.
 - <Name> identifies the field that holds the information that should be populated in the defined column.
 - <Group> identifies the screen that contains the field defined in the <Name> element.
 - <DataType> defines the type of data that will be displayed.
10. Check in the RoleScreen business rule to save all changes.

To edit an existing role view, make sure the RoleScreen business rule is checked out, then click the name of the view in the Navigator window. The existing information will display in the Role Views pane. All information is available for update. Click the **Save** button at the bottom of the pane when all updates are complete.



If a role is removed from the RoleScreen business rule, meaning it is deleted from the <Role> section, and that role was included in a role view, then it should be manually removed from the role view in the XML Source pane.

Admin Expl... Global... XML Navigator

Search: Clear

Functions
Interface
Plan Rules
Screen
ActivityResultsScreen

Navigator

Display Status
Role Views (New Role View)
Owner Role View
Insured Role View
All Roles View

RoleScreen(Functional Prototype Plan - Prototype Child Company)

General RoleScreen **Role Views** Xml Source

<Use Save button on the bottom of this page to save changes>

View Name:

All Roles:

Roles: Cont Bene
 Insured
 Owner Beneficiary
 Payee
 Primary Beneficiary
 Spouse Beneficiary

Table XML Data

```

<Display>Role</Display>
<Name>RoleCode</Name>
<Group>Role</Group>
<DataType>Text</DataType>
</Column>
<Column WIDTH="100" ALIGN="Left">
<Display>Tax ID</Display>
<Name>TaxID</Name>
<Group>Client</Group>
<DataType>Text</DataType>
</Column>
</Table>

```

Save Clear

Role Views Pane of Role Screen

Button Security

Security is applied to the buttons on the Role screen in the Admin Explorer. Open the Security folder to reveal the options for applying security.

Steps to Add Security to Buttons

1. Open **Admin Explorer | Security | Application Security | Security Groups**.
2. Select the Security Group that has access to the Role screen.
3. Open **Plan Security | *Name of the Company* | Plan Pages**.
4. Right-click the PolicyRole node and select **Check out** to open it in the Configuration Area.
5. Check the box next to each button the users in the security group will have access to in OIPA.
6. Check in the file to save the changes to the database.

XML Source Pane

Configuration can also be done in XML via the XML Source pane. Configuration for the additional screen tags must still be done through the XML Source pane.

The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the [XML Source Editing](#) section of this document.

Among the tags that are configured on the XML Source pane is the <DisplayRoleFields> element, which allows you to control the display of roles on OIPA's Role screen. If this element has a value of **Yes**, then the roles will display in a drop-down list on the Roles screen. If this element has a value of **No**, then the roles will display as a series of checkboxes. Additionally, if this element has a value of Yes, the information entered on the Role screen when adding a new role will be validated against other policy and role information. Validations will also be performed upon deleting and updating roles, with error or warning messages displayed when a role that is not allowed to be updated/deleted is updated/deleted. Refer to the [RoleScreen Actions and Events Prototype](#) for a complete explanation of configuring Role screen validation.

Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Plan Rules | Role Screen**.

RoleScreen XML Source Pane

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Segments > SegmentScreen Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

SegmentScreen Rule

The **SegmentScreen** business rule defines the display of the columns in the segment summary. The summary displays above the segment detail information. The SegmentScreen business rule has three panes available: General, SegmentScreen and XML Source, which can be used to configure the Segment screen layout.

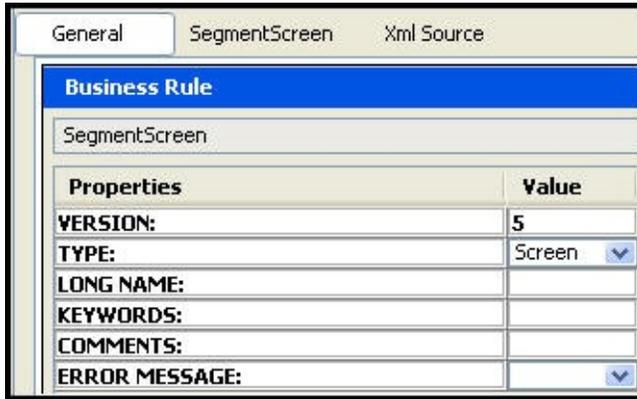
The **Global Explorer** contains the SegmentScreen business rule, which can be overridden for each plan. Open **Business Rules | Plan Rules** to locate the SegmentScreen business rule file.



Actual segments on a policy are configured in the Segment folder. For more information on creating and configuring segments, refer to the [Segments](#) section.

General Pane

The General Pane displays the name and properties of the SegmentScreen business rule.



SegmentScreen General Pane

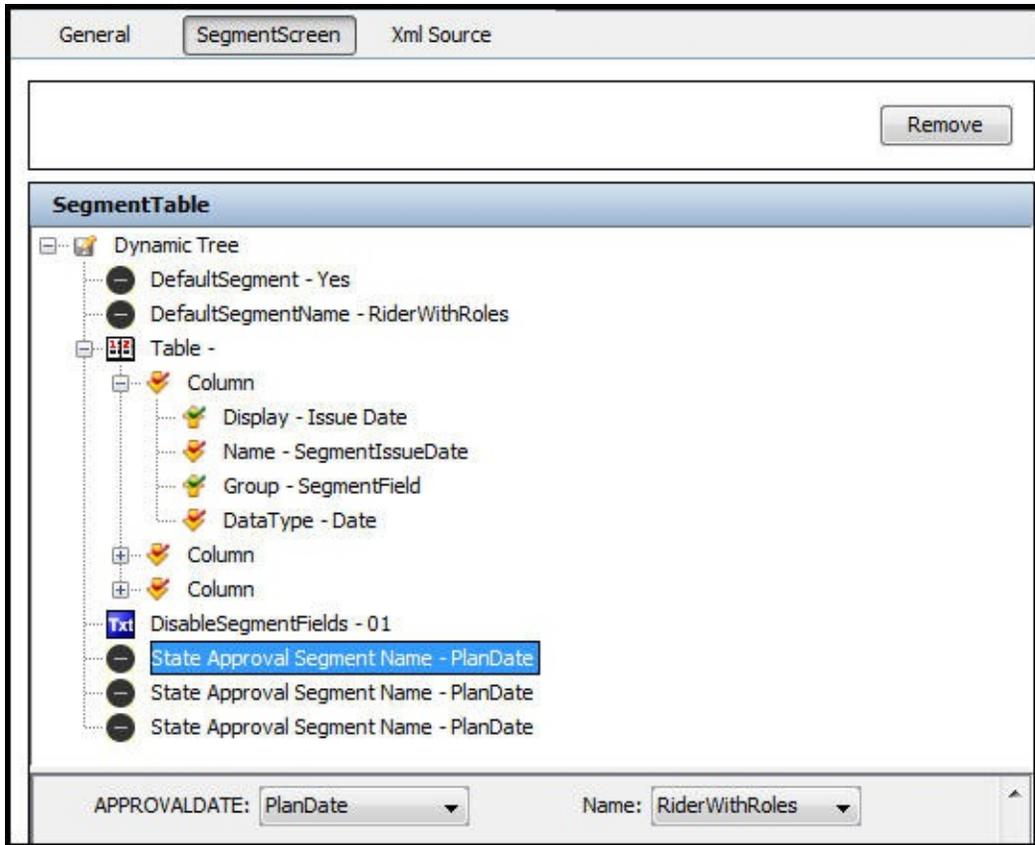
SegmentScreen Pane

The **SegmentScreen** pane allows for the visual configuration of segments and state approval support. Notice the options in the Palette are now specific to SegmentScreen configuration. Drag and drop components from the Palette window onto the Configuration Area.

- **DefaultSegment** determines if a particular segment will appear first in the Add Segment drop down list in OIPA. If set to **Yes**, then the segment name should be defined using the DefaultSegmentName element.
- **DefaultSegmentName** identifies the segment that should be listed first in the Add Segment drop down list in OIPA when a new segment is added to a policy. This element is only used if DefaultSegment is set to **Yes**.
- **Table** sets the width of the column in pixel size and justifies its position. Define the display name of the column heading and enter the name of the field whose value will display under the column heading. The exact name of the field must exist for each segment configured. To display the name of the segment, use the Lookup link.
- **DisableSegmentFields** will disable segment fields from being edited when the policy is in a specific status. Enter the [Code Value](#) from AsCodeStatus for the policy statuses where the segment should be read-only(disabled).
- **StateApprovalSegmentName** will grant state approval access to the segment identified in this element. Each segment must have a separate State Approval <SegmentName> element granting state approval access. The APPROVALDATE drop down box is populated with policy fields with datatype=Date. The name of the segment that will be granted state approval support is selected from the Name field.



If State Approval support is not granted to segments in the SegmentScreen business rule, then the Segment State Approval node in the [State Approval editor](#) will be empty.



SegmentScreen Pane in SegmentScreen Business Rule

XML Source Pane

Configuration can also be done in XML via the XML Source pane. The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the [XML Source Editing](#) section.

Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Plan Rules | Segment Screen**.

SegmentScreen XML

You are here: Configuration > Valuation > Valuation Business Rules > ValuesScreen Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

ValuesScreen Rule

The ValuesScreen business rule is configured at the Plan level. This rule is used with the PolicyValues business rule to display the policy's values on the Values window (invoked by clicking the Values link in the navigation section of the application). The PolicyValues business rule must be configured prior to configuring the ValuesScreen business rule. The ValuesScreen business rule dictates how to display (format) the item/variable on the Values screen.

In addition, ValuesScreen rule can also perform the following:

- Display the valuation date (from PolicyValues) on the Values screen (policy level).
- Display the fund details and pie charts for Allocations and Assets.

Explanation of Key Elements

The <PolicyValues> section defines the screen's display of PolicyValues math variables. Each MathVariable to be displayed should be listed as a separate <Field> subelement. Any policy value that is desired to be displayed on the ValuesScreen must be predefined in the PolicyValues business rule.

The <FundDetails> section allows for the display of fund information. Funds that should not be displayed on the ValuesScreen can be disabled in the <Exclude> subelement of <FundDetails>, with a separate <Fund> subelement for each fund to be excluded.

Fund data that should be displayed on the ValuesScreen is configured in the <Columns> subelement. Each fund field to be displayed should be listed in a separate <Column> subelement. The ValuesScreen also allows for two graphing options. The <Graphs> element can be used to include a pie chart of assets and/or allocations. To enable the pie charts, the respective ASSETS or ALLOCATIONS attribute should be set to **Yes**. If set to **No**, pie charts will not be displayed.

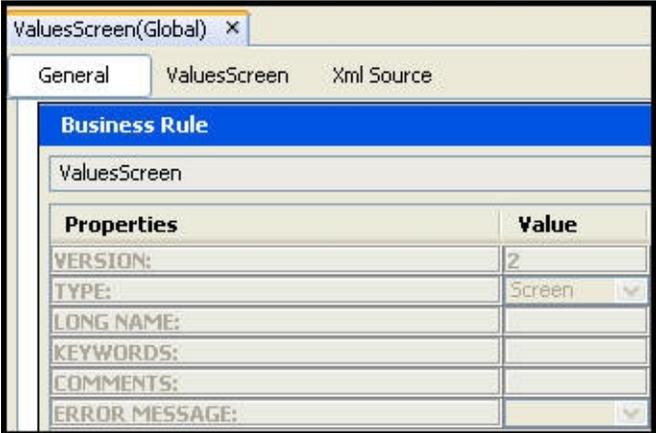
"Note: There should be atleast one allocation saved under the PolicyAllocation tab when the 'Graphs' section is enabled in values Screen."

The ValueScreen is located under the Plan Rules folder. Access it from the Global Rules Explorer under **Business Rules | Plan Rules** or access the ValueScreen overrides from the Main Explorer under the **Plans | Name of Plan | Plan Rules folder**.

Note: The PolicyValues business rule must be configured prior to configuring the ValuesScreen business rule.

General Pane

The General pane displays the name of the rule, its version, type and several other properties fields that allow specific information to be entered concerning the business rule.

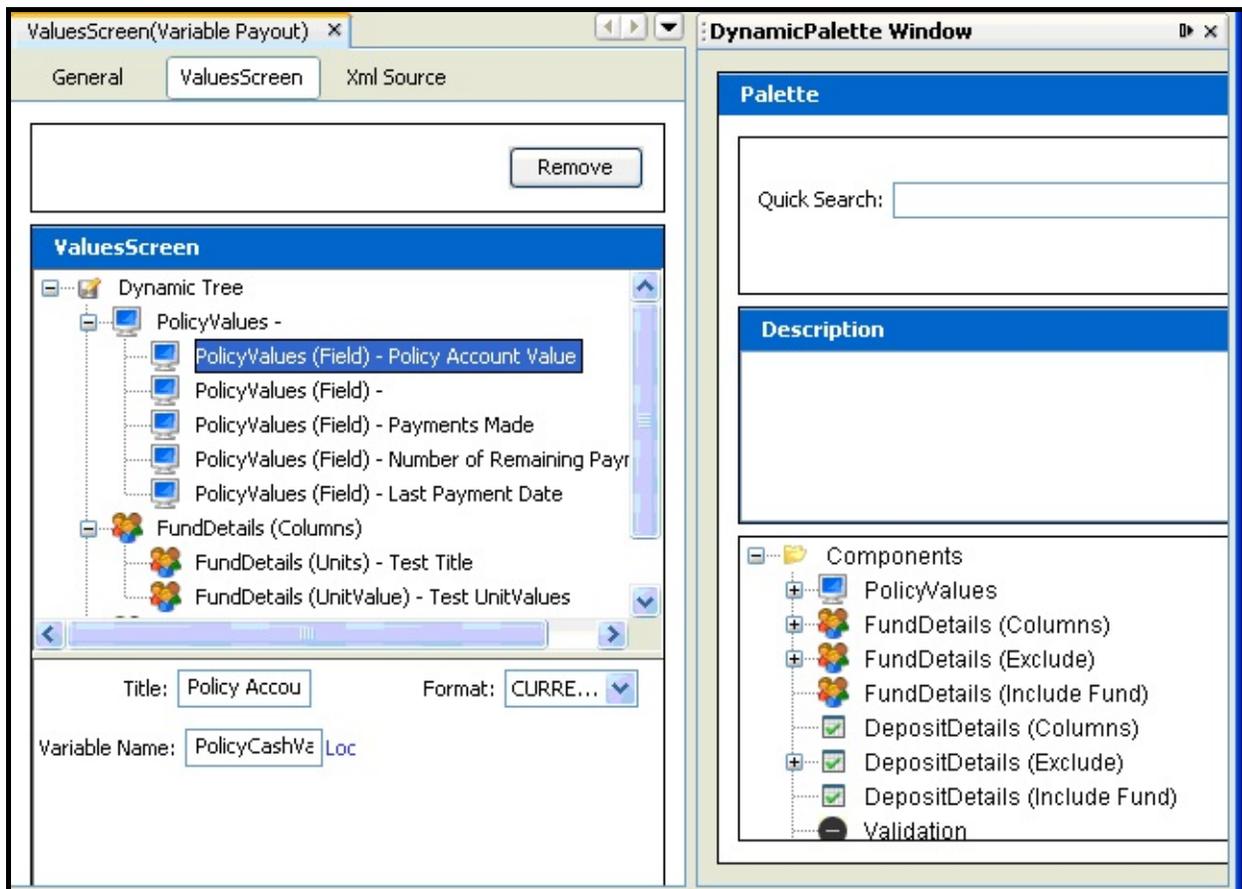


ValuesScreen General Pane

ValuesScreen Pane

The ValuesScreen pane allows for the visual configuration of the Values screen.

- PolicyValues
- FundDetails



The ValuesScreen Pane of the Values Screen

Button Security

Security is applied to the buttons on the Value screen in the Admin Explorer. Open the Security folder to reveal the options for applying security.

Steps to Add Security to Buttons

1. Open **Admin Explorer | Security | Application Security | Security Groups**.
2. Select the Security Group that has access to the Value screen.
3. Open **Plan Security | *Name of the Company* | Plan Pages**.
4. Right-click the PolicyValue node and select **Check-out** to open it in the Configuration Area.
5. Check the box next to each button the users in the security group will have access to in OIPA.
6. Check-in the file to save the changes to the database.

XML Source Pane

Configuration can also be done in XML via the XML Source pane. The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the [XML Source Editing](#) section of this document.

Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Plan Rules | Values Screen**.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Global Explorer and Business Rules > Screen Rules > Overview of Screen Rules

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Overview of Screen Rules

Screen rules are used to control the display of screens in OIPA. Screen rules exist as global rules but can be overridden to meet the needs of individual plans. Global screen rules are located in the **Global Explorer** tab in the **Business Rules** folder. Any overrides (except company level overrides) created will be stored in the **Main Explorer** tab under the **Company | Plan | Business Rules | Screen** node.

Most screen rules are configured using the XML Source pane. Visual editing is not available for these rules at this time. An explanation of each rule is provided below. There are two exceptions: RoleScreen and CompanyScreen. These two rules can be configured using visual editing tools.



Please see the **XML Configuration Guide** topic in this help system for a complete list of all elements, attributes and values needed for Screen rule configuration. View **Business Rules | Screen Rules**.

Screen Rules

ActivityRequirementScreen: The ActivityRequirementScreen business rule must be configured in order for OIPA to handle requirements properly. This rule will need to be configured as a screen rule. The global rule should only have an empty opening and closing tag. The actual rule is configured as a company level override.

ActivityResultsScreen: This rule defines the configuration of the Activity Result screen in OIPA. The screen displays when the Activity Detail icon to the left of a processed activity is clicked. Configuration allows the user to control whether parent and/or child funds display and whether the original allocation amount displays. The rule should be overridden at the transaction level if either one of these display enhancements is required. If one or both of these display enhancements is needed, the rule configuration should be updated to reflect the display required and should be attached to a specific transaction. A transaction override of the rule is not included in the TransactionBusinessRulePacket.

The [Verification Screen](#) provides a different view, completely configurable, with a section for allocation. The VerificationScreen makes the optional SHOWORIGINAL attribute available.

AddressScreen: This business rule allows the user to configure the non-fixed fields and validations for the address roles on the Address screen. Fixed field values can also be controlled through configuration of this rule. Mailing addresses can be set to expire based on date criteria and a contingent value established to introduce an active or inactive status. Configuration supports foreign addresses and dates. The AddressType is controlled by the AsCode table > **AsCodeAddressType**.

If phone number fields are required on the Address screen, then the **<DisplayPhoneScreen>Yes</DisplayPhoneScreen>** configuration must be present in the AddressScreen business rule.

An address change letter can be automatically generated when a change is made to an existing address. The Spawn IF logic will determine when

and if a letter is generated. Configuration has the option to spawn a client level activity and create messages as needed through events/actions. To make use of this functionality the following configuration requirements must be met.

- The activity must be a client level activity.
- Spawning is only available from the Address screen.
- The only spawn code supported is 03.

Refer to the [Address Change Letter Prototype](#) for configuration steps to accomplish this task.

Agreement Screen: The Agreement screen rule allows the user to configure the Agreement Screen on the Group Customer navigation options to either show the summary level Group Customer information or skip it and also define the number of levels of agreement hierarchies to be displayed ONLOAD.

AgreementRoleScreen: This rule allows the configuration of agreement roles as they relate to a specific Agreement Role Type (AsCodeAgreementRoleType). This includes the configuration of fields or field subsections specified on the Agreement Role Details tab for the Agreement Role screen in OIPA. Each agreement role, such as a Contact, Administrator, Bank Representative or TPA resource, can have configured fields for capturing relevant data required by the business.

- The TYPECODE attribute defines the role types that are available as an agreement role. If this attribute is not present in the <AgreementRole> element, then roles cannot be added to the agreement.
- The ALLOWGROUPECUSTOMER attribute controls whether the Find Customer tab displays at the top of the Add Agreement Roles window. The Find Customer tab will appear on the screen if at least one of the agreement role types has this attribute set to **Yes**. Only the agreement role types that have the attribute set to **Yes** will be available in the Agreement Role drop down list under the Find Customer tab.
- The DISABLEBYAGREEMENTSTATUS attribute restricts the use of

a role by agreement status. Agreement statuses are defined in the Rules Palette in the Admin Explorer. Click **Admin Explorer | Administration | Code Names | AsCodeAgreementStatus** to add and update agreement statuses.

- A CLIENTTYPE attribute allows the role to be restricted to specific clients. This is a required attribute. If a client is not identified in this attribute, then that client cannot be associated with the role. Client Type drop down box for “The Find Client” and “New Client” are controlled by this attribute. If this attribute is omitted or no client types are defined for a particular role type, that particular role type will not be listed in the role type dropdown for New Client and Find Client tabs.

The [AddAgreementRoles](#) business rule can be attached to a transaction to add roles to agreements via activity processing.

AllocationScreen: This business rule defines the allocations that are assigned for the plan and policy. This rule does not have visual editing support and must be configured directly in the XML Source pane. It is used when [configuring allocations using the default method](#).

Alternate Name Screen: This business rule defines an alternate naming convention for a Group Customer. The [alternate name](#) will be a field used when search criteria is entered for the Customer Search screen and can be used for identifying the Group Customer and additional information housed under the alternate name.

CaseScreen: This business rule allows a user to create and edit case records, part of the New Business Underwriting process. In OIPA, the Case screen is accessed by selecting **Case** from the Main Menu and clicking on **New Case**, or by selecting **Search Case** from the Main Menu, then searching for and selecting an existing case.

Information displayed on this screen includes the case name, case number, case status, creation date and date last updated, all of which correspond to columns within the AsCase database table. Additionally, dynamic fields can be configured to display other information on the screen. A case's status is represented by a two digit role code, and these

role codes are defined in the AsCodeCaseStatus code name.

The screen has three sections:

- The Case General Info section displays the screen's fixed fields, such as case name and number.
- The Policy/Application Table section displays the table defined in the <Table> tags within the <Policies> tags of the CaseScreen rule.
- The Case Detail section displays the dynamic, configurable fields defined in the CaseScreen rule's configuration. These fields' values are stored in the AsCaseField database table.

Masking and field-level security is supported on the Case screen.

CaseSearchScreen: This business rule is used to configure the CaseSearchScreen, which allows an underwriter or CSR to search for cases or applications as part of the New Business Underwriting process. The CaseSearchScreen functions similarly to the Policy Search screen. The CaseSearchScreen's configuration defines the fields that are used to store the results of a case or application search. In OIPA, the Case Search screen is accessed by selecting **Case** from the Main Menu and clicking on **Search Case**.

The screen has two sections:

- The Case Search Criteria section contains the fields into which case search criteria can be entered.
- The Case Search Results section displays the results of a case search.

ChartOfAccountsScreen: This rule determines the set-up of the dynamic fields for the ChartOfAccounts screen (only for the criteria section) and determines how validation can be performed.

ChildFundScreen: This rule has been deprecated as of the 9.5.0.0 release. Refer to the [Child Fund](#) page for information on setting up child funds.

Class Screen: This business rule defines the characteristics for each

class within a Class Group. Information such as the class type, class name and class description displays as well as any dynamic fields captured at the class level. In addition, the screen provides a tabbed view that allows a user to drill down to the plans associated to each class (if applicable). Class rules and Class Rule Variables used for defining membership are also available on this screen. Any members enrolled in the plans associated with the class are available as well.

This screen rule has an upper section that allows for the display of Fixed Fields for class type, class name, description, and number of members. In addition, there are configurable dynamic fields to further define the class. The screen layout contains a subsection that can be expanded and collapsed by clicking on the section header for Class Detail and the expanded subsection tab will be highlighted in red. Tabs are available for Class Details, Class Plans, Class Rules and Class Members. Refer to [Classes and Class Groups](#) summary section and the [Classes and Class Groups prototype](#) example for understanding this rule in more detail.

Class Group Screen: This business rule defines the collection or arrangement of classes for the purpose of viewing and editing Class Groups. The Class Groups are arranged by type and can be copied and created in OIPA. The Class Group Screen also has options to define whether Business Status should be used for class group records and also to define the transaction that needs to be spawned in case a class group time slice is submitted. Refer to [Classes and Class Groups](#) summary section and the [Classes and Class Groups prototype](#) example for understanding these rule in more detail.

Client Relationship Screen: This business rule defines the primary and secondary relationships associated with a Group Customer. The screen provides the connection between a client entity through a primary relationship to the Group Customer. The relationship can be further defined by a secondary relationship type. The optional **<MaximumDuplicate>** attribute specifies the number of times a relationship can be duplicated for a client. A validation will prevent the user from adding more than the maximum number designated and presents the user with the following error message, "Maximum count

exceeded for this relationship.” If the <**Maximum**> attribute is not configured, then an infinite number of same client/same primary/same secondary relationships may exist for the Group Customer.

ClientScreen: This business rule allows the user to configure fixed and dynamic fields on the Client screen. A separate section is configured for each Client Type, which is identified by its typecode. This rule is also used to control the display of policy roles, individual fields, address table, the TaxID field on Client screen, and the process button for future activities on the Client Activity screen.

Configuration supports the display of foreign calendars and dates as well as numerous formats of the client name. The LegalResidenceCode field and the TypeCode are the only required fields and they determine the display of all information on the screen.

If this screen is called as a result of the use of a [client field](#) in an activity, then it will display in a popup window. Security for fields and buttons displayed in the popup window tabs will be determined by the security established on the current Client screen.

The Spawn element can be used within Actions and Events to trigger the spawning of activities when specific fields are updated on the Client screen. [View the prototype for configuration details.](#)



There are some **configuration considerations** to keep in mind when working with the Client screen. In version 9 of OIPA, the ClientGUID is not an inherent GUID available to the fields section of the Client screen. This means that when a new client is created, a ClientGUID does not yet exist until after the Save button is clicked. If there are fields that require a ClientGUID, they can cause the screen to crash (stack trace) since the ClientGUID cannot be found. To avoid this situation, move any SQL in the field section that needs to resolve the ClientGUID into the Action/Event section for OnChange or OnSubmit events. This will insure proper screen results for a new client. This is not an issue for saved clients in the database since the ClientGUID

already exists.

The Client screen controls several aspects of the Group Customer screen in OIPA. Within the <Client> element, several important attributes are:

- **TYPECODE**: one of two attributes used to identify the type of client. Values for this attribute are stored in AsCodeOrganizationType.
- **CLIENTTYPE**: the second attribute that is used to identify the type of client. Values for this attribute are stored in AsCodeClientType.
- **ACTIVITYPLAN**: used to tell OIPA which plan to use when the Group Customer screen is loaded. All activities configured on this plan will be available as Group Customer level activities in OIPA.
- **RELATIONSHIPACTIVITYPLAN**: used to tell OIPA which plan to use when the Relationship link is selected in OIPA. All activities configured on this plan will be available as ClientRelationship activities in OIPA. These activities must be configured as Client-Relationship transaction types in order to appear in the Client-Relationship activity drop down list in OIPA.

If phone number fields are required on the Client screen and Group CustomerScreen, then the **<DisplayPhoneScreen>Yes</DisplayPhoneScreen>** configuration must be present in the ClientScreen business rule.

ClientSearchScreen: This business rule defines the configuration for search criteria and fields for the Client Search screen. The screen allows the user to search on various client types such as Individual, Corporate or Producer. When the user enters this screen, the default search criterion is Individual.

External clients can be searched from this screen in OIPA if the supporting configuration is added to this rule. The attribute EXTERNAL="Yes" can be added to the <Client> section to indicate that the client type is external. The element, <ExternalClientSearchRetriever>, can also be added to the <Client> section of the ClientSearchScreen. This element defines how the client specific information is to be populated in the Client Search results when

the user hits the **Find** button. This element contains a class name that implements an interface. The data retrieved is defined in the ExternalClientDetailScreen business rule

If this screen is called as a result of the use of a [client field](#) in an activity, then it will display in a popup window. Security for fields and buttons displayed in the popup window tabs will be determined by the security established on the current Client Search screen.

This rule will also allow use of policy and segment context variables providing POLICYGUID and SEGMENTGUID values for any search parameters. The PolicyGUID and SegmentGUID context variables will return appropriate values when there is a Policy and Segment context. In case there is no Policy context , the values will be returned as "Null" without any system error.

ClientSearchScreen BusinessRule works in three modes:

- Client Search under Client Context
- Find Client in RoleScreen (Policy and Segment Roles)
- Find Client in any Activities

In the RoleScreen FindClient context, the ClientSearchScreen will be able to access the POLICYGUID (both Policy and Segment RoleScreen)and SEGMENTGUID (only in Segment RoleScreen) fields using which other policy and screen field values can be calculated for search parameters. In other contexts, these two values will return "Null" in the ClientSearchScreen BusinessRule.

CommentsScreen: This business rule is used to configure OIPA's various Comment screens, which can be implemented for policies, segments, clients, activities and suspense records. Comments on these screens can use preset comment templates, which are configured with

the **Comment Templates** node in the Admin Explorer's Administration folder, or can be completely user-entered. Comment templates can be implemented at the global, company, Product or plan level.

CommentsSearchScreen: This business rule is used to define the search criteria for comments and to configure the display of comment search results. It can be configured at the global, policy, segment, activity, and client levels, as well as for suspense records. In addition to containing fixed fields, the Comments Search screen can use dynamic fields to filter out specific types or categories of comments.

CompanyScreen: The CompanyScreen business rule must be configured before accessing the Company Data node in the Main Explorer. This business rule defines the fields that hold constant values related to a company. It should only be overridden at the company level. The XML Source pane can be used to configure the screen using XML.

A field defined as <DataType>**Money**</DataType> in the CompanyScreen business rule will display a currency field for entry in the Rules Palette [CompanyData](#) node.

[DisbursementApprovalScreen](#): This business rule allows for the configuration of dynamic fields on this screen. These fields will be used to search for specific disbursements. The table section defines how the results are displayed to the user. The Disbursement Amount column can be totaled using optional configuration. Currencies must be of the same type. Mixed currencies will not total.

DisbursementScreen: This business rule contains the fields that display when the Disbursement link is selected in the Activity Results window.

[DisbursementSearchScreen](#): This business rule is used to configure the dynamic fields in the DisbursementSearchScreen to allow the user to search for disbursement records that match the specified criteria. If the DisbursementSearchScreen business rule is not used the fixed fields will be displayed by default and used for searches. For example Company, Plan, Start Date and End Date will be displayed.



The DisbursementScreen and DisbursementSearchScreen business rules together constitute the Disbursement screen. Configuration for the Disbursement Search section is done in the DisbursementSearchScreen business rule; whereas the configuration for the Disbursement details section is done in the DisbursementScreen business rule.

Enrollment Screen: This business rule defines the enrollment processing for an eligible member to participate in coverage. The selections made within the Enrollment screen are for coverage, beneficiaries and dependents. The screen provides the ability to enroll the member into multiple coverages simultaneously as well as perform partial enrollment while saving the in-progress data for completion at a later time. Enrollment can be active and complete or pending and partially complete.

ExternalClientDetailScreen: This business rule holds the associated configurable fields for an external client. The fields defined in this business rule will be accessible through SQL. Specific external client information stored in the external database will not be accessible in the OIPA database.

Client details for a role may be configured in this rule. Role fields are configurable fields on the Role Screen business rule based on the role code. The Role Code for the External Client will be labeled as 'External'. The field data is stored in OIPA's AsRole and AsRoleField tables.

FundScreen: The FundScreen business rule can be configured to provide additional information for fund records. This rule defines whether there will be child funds and/or benefit funds. Parent and child funds are used when the same fund may be offered but there are different classes of the fund (versions, bands, groups, etc.). Extra fields can be stored at the parent and child level of the fund. Additionally this is where funds applicable to benefit split are determined as well.

This rule is overridden at the Primary Company Level and copybooks are not supported in this rule.



This rule must be configured in the XML Source pane.

If child funds or benefit funds are needed, the following attributes must be present in the <ChildFunds> element.

- **ALLOWED**: this is a required attribute that will accept a literal **Yes** or **No** with the default value being No. This attribute will indicate if child funds should be created from parent funds.
- **BENEFITFUNDS**: this is an optional attribute that will accept a literal **Yes** or **No** with the default value being No. This attribute will indicate if benefit funds should be created from child funds.

Group Customer Screen: This business rule is similar to the Client Screen, but pertains specifically to Group Customers. The screen allows for the addition or editing of Group Customer clients. The business rule can be overridden at the Primary Company level.

A complete explanation of the elements available for this rule is included in the XML Configuration Guide. An overview of the major elements is provided below.

- The TYPECODE attribute in the <PrimaryRelationship> element defines the primary relationship type from AsCodePrimaryRelationshipType.
- The secondary relationships that can be associated to the primary relationship are defined in the <SecondaryRelationships> section.
 - The <TypeCode> element has a VALUE attribute, which identifies the secondary relationship type from AsCodeSecondaryRelationshipType.
 - The <ClientTypeIdentifier> element is used in conjunction with the main level <ClientTypeIdentifier> element at the bottom of the rule. This element identifies the ID that should be referenced when determining the client types that can be assigned a secondary relationship.
- The <ClientTypeIdentifier> element at the bottom of the rule has an ID attribute, that corresponds to the <ClientTypeIdentifier> element above in

the <SecondaryRelationship> section. Match the value from the above <ClientTypeIdentifier> element to the ID to find the client type.

- The <ClientType> element in the <ClientTypeIdentifiers> section holds the code that comes from either AsCodeOrganizationType or AsCodePersonType. Only client types defined in this element will be available to assign a secondary relationship.

Group Customer Search Screen: This business rule allows for the configurable search criteria in order to find the applicable Group Customer. The search can be filtered by selecting various criteria and selecting values for those criteria.

IntakeFileSearchScreen: This business rule allows the user to search for specific Intake Files. Search fields can be configured to be able to search on fixed and dynamic fields in order to narrow the results based on the search criteria. The rule also allows for configuration of a search results tables. The IntakeFileSearchScreen can be overridden at the global or primary company levels only. This screen contains several tabs that display various details pertaining to a selected Intake File:

- **File Details:** This tab displays an Intake File's status, received date, processed date and other basic information.
- **Activity Math:** This tab displays all of the MathVariables for the Intake File's Intake File transaction once the activity has moved to "Active" status.
- **System Error:** This tab appears when a system error is incurred during Intake File-level transaction processing, and displays an error table displaying the details of each error.
- **Business Error:** This tab appears when a business error is incurred during Intake File-level transaction processing, and displays an error table displaying the details of each error.

IntakeProfileScreen: The IntakeProfileScreen business rule allows for configuration of the Intake Profiles table via an optional table section, which has access to any combination of available columns from the AsIntakeProfile and AsIntakeProfileDefinition tables. The Intake Profiles table displays Data Intake Profiles for the current Group Customer. This screen also allows for the creation of Data Intake Profiles, although this

functionality involves no configuration.

IntakeRecordSearchScreen: This business rule allows the user to search for specific Intake Records. Search fields can be configured to be able to search on fixed and dynamic fields in order to narrow the results based on the search criteria. The rule also allows for configuration of a search results tables. The IntakeRecordSearchScreen can be overridden at the global or primary company levels only. This screen contains several tabs that display various details pertaining to a selected Intake Record:

- **Record Details:** This tab, open by default, displays an Intake Record's status, record type, processed date, pre-processed date and other basic information.
- **Activity Math:** This tab displays all of the MathVariables for the Intake Record's Intake Record transaction once the activity has moved to "Active" status.
- **Activity Sequence:** This tab displays a hierarchical "tree" of activities generated by the Intake Record transaction for the selected Intake Record. Details pertaining to each selected activity on the tree of activities are displayed to the right of the tree.
- **Record XML:** This tab displays the received Intake Record XML configuration for the selected Intake Record.
- **System Error:** This tab appears when a system error is incurred during Intake Record-level transaction processing, and displays an error table displaying the details of each error.
- **Business Error:** This tab appears when a business error is incurred during Intake Record-level transaction processing, and displays an error table displaying the details of each error.

PhoneScreen: This business rule controls the configuration and display of phone numbers on the Phone screen when the PhoneNumbers link is click from the Address screen, Client screen and Group Customer screen. The **<DisplayPhoneScreen>Yes</DisplayPhoneScreen>** configuration must be added to the Address screen and Client screen to invoke this business rule.

The phone format is driven by the phone type and country selected in

OIPA from the Phone screen. The PhoneScreen business rule defines all the possible phone type and country combinations using <Phone> sections. A default country and phone type can also be specified in the rule using the <DefaultCountry> and <DefaultPhoneType> elements. Use values from the AsCountry table for the default country and values from AsCodePhoneType for the phone type.

Each phone type is configured in a <Phone> section. The following elements apply to each <Phone> section.

- The <PhoneType> element has two attributes. VALUE is the code value and NAME is the short description of the phone type. These can be found in AsCodePhoneType. The <CountryCode> element should have a value from the CountryCode column of AsCountry.
- The <FixedFields> section contains the fields that are used to capture the phone information. [Masks](#) are supported as a field datatype. If a mask is used, it will control how the phone information displays as the user types it in. Masks are defined in the [Mask editor](#) of the Rules Palette.
- The <DisplayFormat> section controls how the phone number displays in search results and on screens in OIPA.

To view all the configuration steps required in formatting phone numbers, refer to the [Group Customer Address and Phone Number](#) page.

PlanActivityScreen: This Business Rule controls the Plan-Level Activity screen. The configuration will determine the number of activities that will be shown on the Plan-Level Activity screen, set the date from which to display activities and provide warnings when using activity icons. This rule may be defined at the Global level or as a Plan level override.

PlanCoverageScreen: The PlanCoverageScreen business rule allows users to attach sub-plans to a class. This screen is accessed by clicking on the Sub-Plan Details tab of the Class Sub-Plans screen. The only configurable aspect of this screen is the table in which the sub-plan records display. Refer to [Classes and Class Groups](#) summary section and the [Classes and Class Groups prototype](#) example for understanding this rule in more detail.

PlanSegmentNameClassParticipantsScreen: The PlanSegmentNameClassParticipantsScreen business rule controls the data that displays on the Class Sub-Plan Participants tab of the Class Sub-Plan screen. This tab displays data related to participants enrolled in a particular Sub-Plan. The participants that display on this tab are determined based on the policies that have a Class Sub-Plan with a segment role code indicating that the member is a sponsor. Refer to [Classes and Class Groups](#) summary section and the [Classes and Class Groups prototype](#) example for understanding this rule in more detail.

PlanSegmentNameClassScreen: The PlanSegmentNameClassScreen business rule controls the configuration of the Class Sub-Plans screen, which displays the Class Sub-Plan associations for a selected class. The Class Sub-Plans screen is a tab of the Class screen, which is configured with the ClassScreen business rule. Refer to [Classes and Class Groups](#) summary section and the [Classes and Class Groups prototype](#) example for understanding this rule in more detail.

PolicyOverviewScreen: This business rule is used to configure the PolicyOverviewScreen. This screen provides a read only summary of all policy details. It is the first option on the Left Navigation menu and is the default screen view when a policy is loaded in OIPA. Both fixed and dynamic fields from the PolicyScreen can be configured on this screen, as well as new fields, and CopyBooks are supported. All Data Types supported by the Field section of OIPA screen rules are supported in the PolicyOverviewScreen, with the exception of Client and Identifier types. Overrides of this screen are supported at the Global, Subsidiary Company, Product and Plan levels. Screen warning can be configured using Actions, Events and ScreenMath. On Load events for fixed and dynamic fields are also supported. Security is applied at the [Plan Page](#) level in the Admin Explorer.

The screen is divided into sections, the order of which is shown below and is set in base code. If a section is not present in configuration, then it will not display on the Policy Overview screen. If a user does not have access to the original page that corresponds to the section, then that

section will also not display.

- **Policy Details:** This is the first section of the rule. If this element is present in configuration, but no fields are defined, then the section will appear, but will be blank. Fields seeking data outside of Policy (for example, Client, Segment, Address, etc.) will require a query to populate. For all SQL access, the PolicyGUID must be a known value on the screen. If you are configuring a policy field for display, no query is needed. The field configuration will carry the same field name and data type and will pull the value from the Policy screen. If the field value changes on the Policy screen, the same field value will be reflected on the Policy Overview screen. Field level security and masking are defined in the PolicyOverviewScreen rule and are not inherited from the PolicyScreen rule. Field names determine whether fixed and dynamic fields belong to the Policy screen. Values are initiated from the corresponding Policy fixed or dynamic fields.
- **Policy Roles:** The SHOW="Yes" attribute tells OIPA to display the Policy Roles section. All active roles will display. The <Message> element allows a configured message to be presented to the user on the Policy Roles section in OIPA.
- **Segments:** The SHOW="Yes" attribute tells OIPA to display the Segments section. All segment fields will be displayed. The <Message> element allows a configured message to be presented to the user on the Segments section in OIPA. If Segments configuration is not present, Segment Roles will not be available for display.
- **Segment Roles:** The SHOW="Yes" attribute tells OIPA to display the Segment Roles section. This section displays all information for segment roles that is configured in the SegmentRoleScreen rule. The <Message> element allows a configured message to be presented to the user on the Segment Roles section in OIPA.
- **Values:** The SHOW="Yes" attribute tells OIPA to display the Values section. The <Message> element allows a configured message to be presented to the user on the Values section in OIPA.

PolicyRequirementScreen: This business rule is used to configure OIPA's requirement summary table, which is accessed by clicking the **Requirements** link in the menu on the left side of the screen when an

application or policy is open. If this rule is not configured, a default table will be used to display the requirement summary.

PolicySearchScreen: This business rule is used to configure the PolicySearchScreen. It defines the fields that are used to store the results of a search.

RequirementResultSearchScreen: This business rule is used to configure the Requirement Result Search screen, which is used to search for requirement results and, if needed, match them to existing requirements.

SegmentRoleScreen: This business rule defines the dynamic fields that can be displayed and updated on the specified Role Detail(s) windows. The segment selected during the policy entry process dictates which role options are visible and available on the Segment Role screen. This rule exists at Global and Plan levels. Configuration should only create company level overrides of this rule at the primary company level.

SuspenseScreen: This business rule is used to create and control suspense records. Suspense records are used to track money. This business rule identifies where the money came from and allows for the money to be used as payment to various polices. A suspense record is used as a holding account until the money is applied or refunded. A unique suspense number is generated with the suspense record for identification purposes.

SuspenseSearchScreen: This business rule is used to configure the Suspense Search Criteria section and Results section of the Suspense Search screen. Fields from AsSuspense and AsSuspenseField tables can be used as the suspense search criteria, based on specific suspense records in the database. The Results section can be configured as is the case with other search screens, to present on the UI as a grid using standard table definition syntax. [View prototype example...](#)

UnmatchedResultSearchScreen: This business rule allows a user to

search for unmatched requirement results and to manually match them to requirements.

WithholdingScreen: This business rule defines the layout of the Withholding screen, which signifies the amount or percentage of federal and state taxes to be withheld from taxable disbursements defined by the Policy Owner.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Group Information > Agreements > Agreement Screen_Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Agreement Screen_Rule

The Agreement screen rule allows the user to configure the Agreement Screen on the Group Customer navigation options to either show the summary level Group Customer information or skip it and also define the number of levels of agreement hierarchies to be displayed ONLOAD.

General Pane

The General Pane displays the name and properties of the AgreementScreen business rule.

XML Source Pane

Configuration can be done in XML via the XML Source pane. Configuration for the additional screen tags must still be done through the XML Source pane.

The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the XML Source Editing section of this document.

Please see the XML Configuration Guide topic in this help system for a list of all elements, attributes and values needed for configuration. View [Business Rules | Screen Rules | Agreement Screen](#).

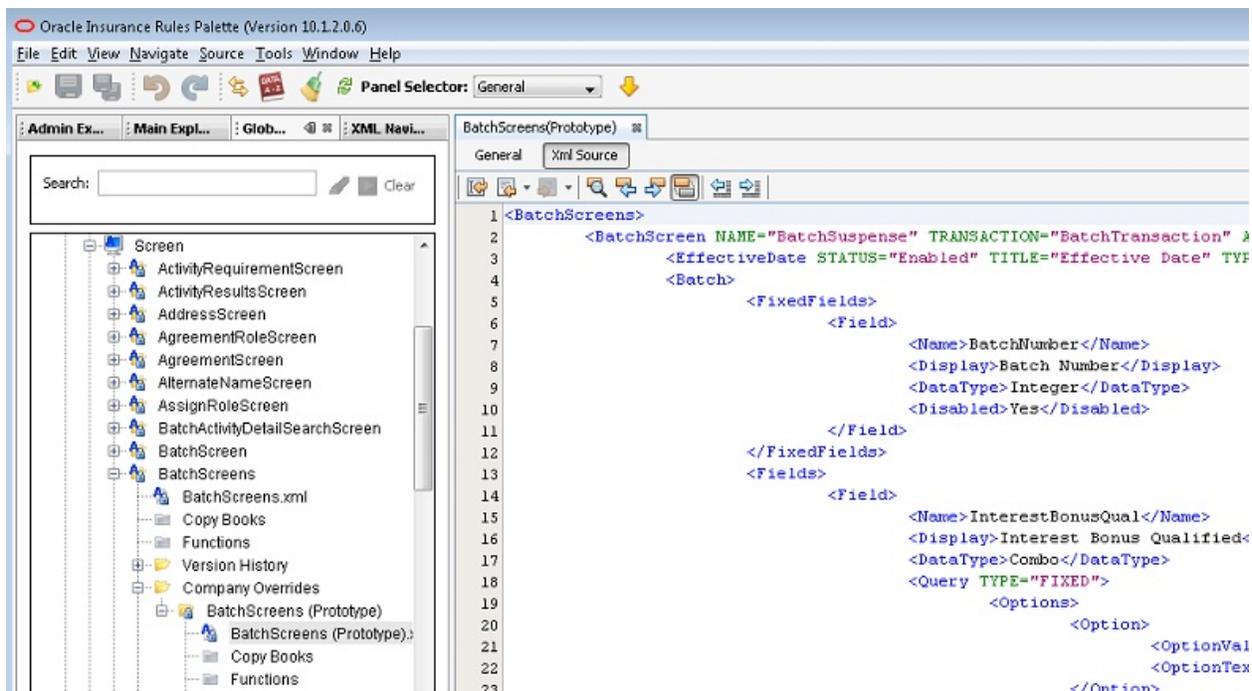
You are here: Global Explorer and Business Rules > Screen Rules > BatchScreen_Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

BatchScreen

Batch Processing allows a certain defined business process to be invoked upon a group or 'batch' of specified policies which can be extended to any entity. This feature helps user to invoke a defined business process for all specified policies from one screen (Batch Screen) and avoids the need to go to each individual policy.

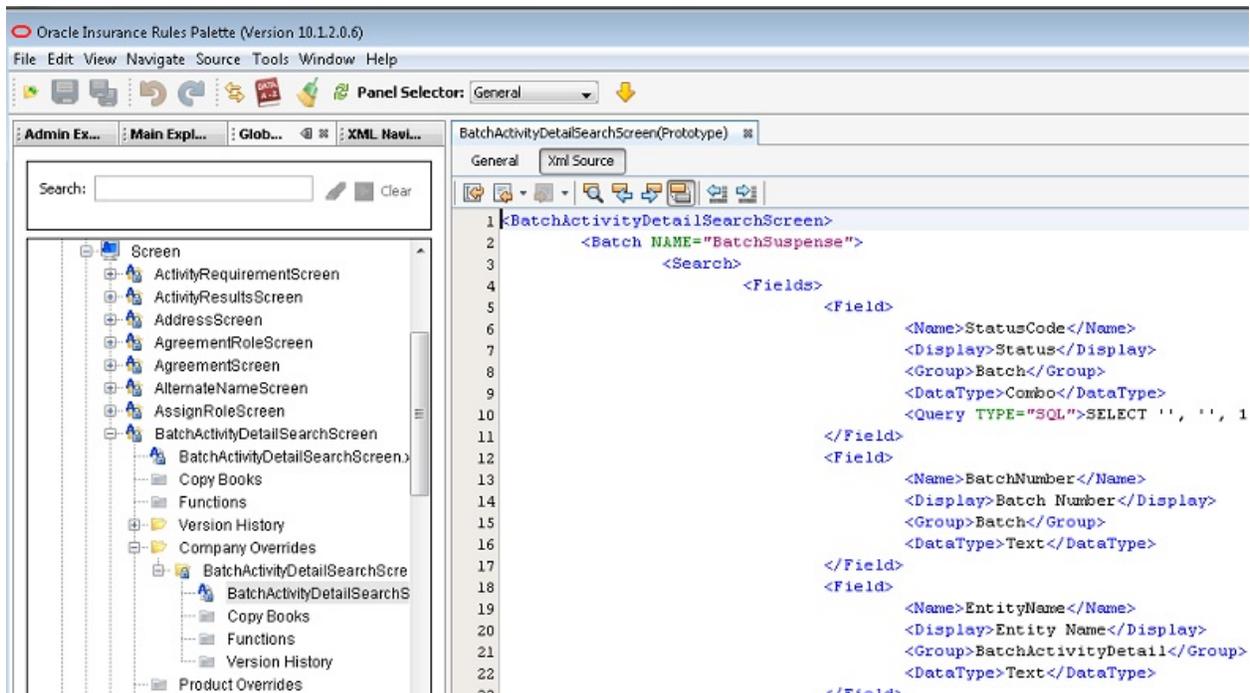
XML Source Pane



Batch Search Screen

This allows user to search the existing batch.

XML Source Pane



Miscellaneous Screen Details: Security

Page Name: Batch

Buttons:

- Save
- Release
- Delete
- Find
- New

Page Name: BatchActivityDetail

Buttons:

- a. Delete
- b. Select
- c. Detail
- d. Save
- e. Skip
- f. Back

Page Name: BatchActivityDetailField

Buttons:

- a. Ok
- b. Cancel

Page Name: BatchActivityDetailSearch

Button: Find

You are here: Group Information > Agreements > AgreementRelationship Business Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

AgreementRelationship Business Rule

The AgreementRelationship business rule is used to define the parent/child relationships between Agreement Types. Once the parent/child relationship options are defined using the specific AgreementType, the Add Child option will display only agreements which are allowed as child depending on the Agreement Type of the selected parent will be displayed by the OIPA application.

The AgreementRelationship business rule is available under the System rules on the Global Explorer pane. The AgreementRelationship business rule can be overridden at the Primary Company level.

You are here: Group Information > Agreements > Agreement Role Screen_Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Agreement Role Screen_Rule

The [AgreementRoleScreen](#) business rule tells OIPA what to display on the Roles tab of the Agreement screen. The fields that will be used to capture the role data are defined in this business rule.

The [AddAgreementRoles](#) business rule can be attached to a transaction to add roles to agreements via activity processing.

General Pane

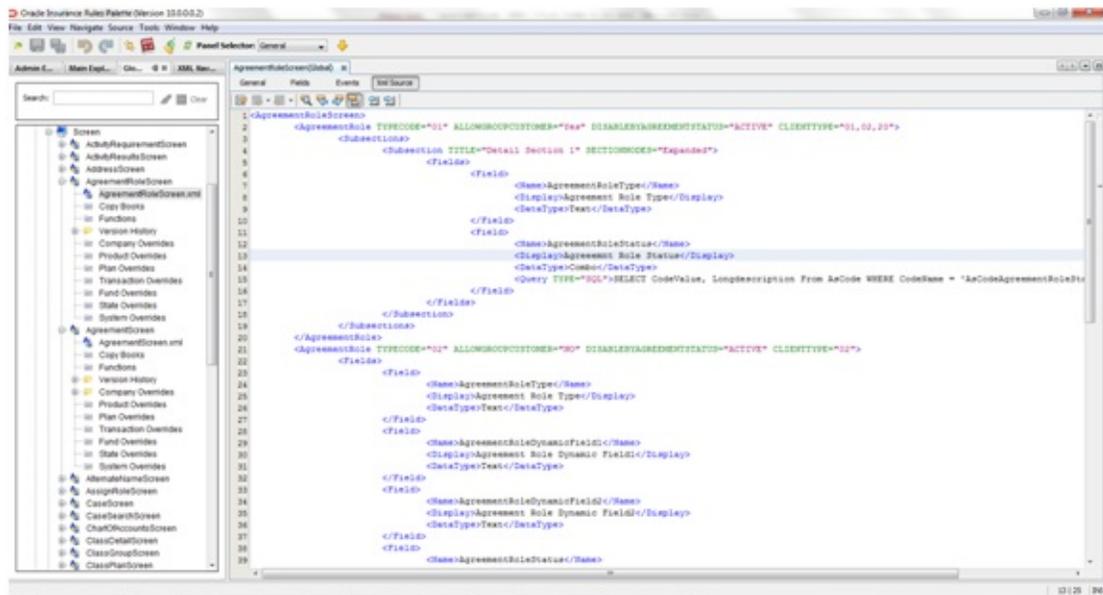
The General Pane displays the name and properties of the AgreementRoleScreen business rule.

XML Source Pane

Configuration can be done in XML via the XML Source pane. Configuration for the additional screen tags must still be done through the XML Source pane.

The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the XML Source Editing section of this document.

Please see the XML Configuration Guide topic in this help system for a list of all elements, attributes and values needed for configuration. View [Business Rules | Screen Rules | AgreementRoleScreen](#).



Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Group Information > EnrollmentScreen_Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

EnrollmentScreen_Rule

The Enrollment Screen allows you to handle the manual additions of participants and eligible individuals as enrollees for the Group Customer.

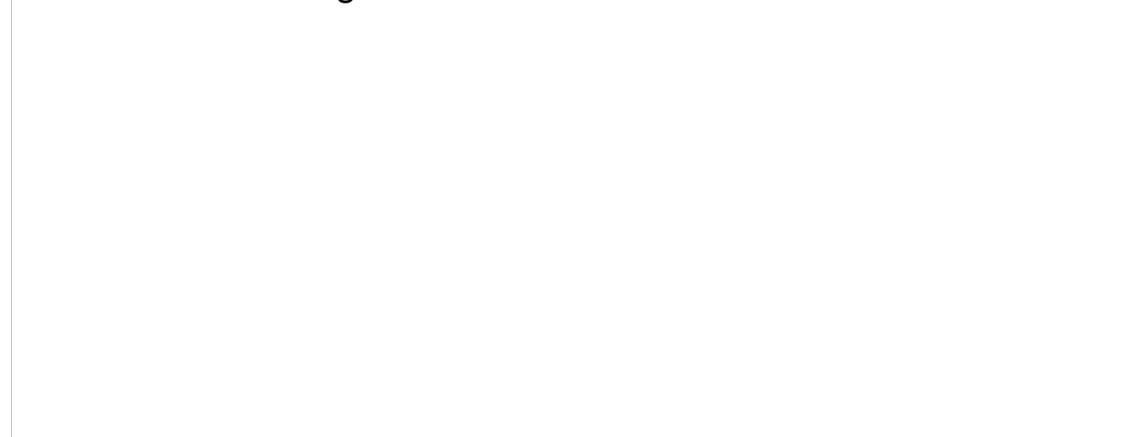
The Enrollment Screen allows you to make Product, Plan, Coverage, and Role selections. You can enroll an eligible participant into multiple products without leaving the screen. In addition, you can revisit an enrollment that is in progress / pending and make changes.

Validation is included within the screen when a user attempts to access the screen for a client who already has an active enrollment record in existence on the system, all enrollment fields and buttons become disabled.

The Enrollment Screen is override able at the Company level.

The selection of Product, Plan, and Coverage information is driven by drop down selection boxes. The selection of the product, plan and coverage information is being carried out using the appropriate qualifying record from the Plan Eligible Classes data table. Another section allows for the designation of Roles.

Enrollment Plan Eligible Classes

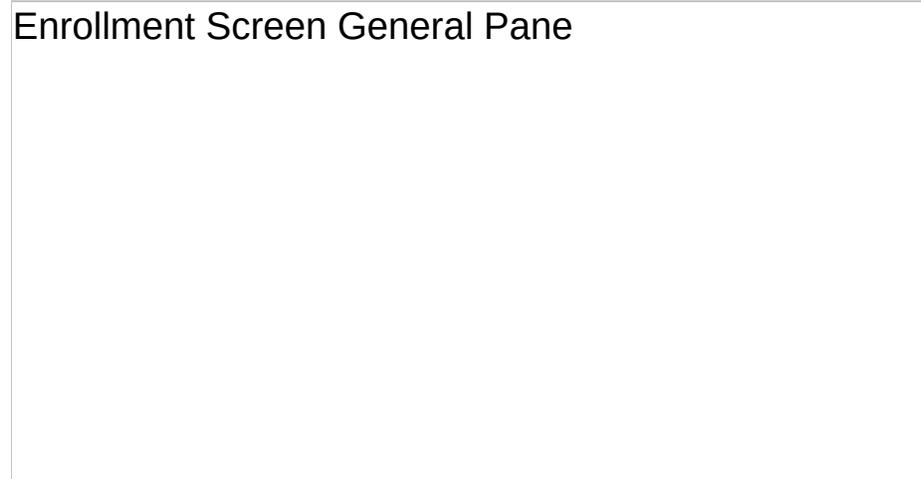


EnrollmentScreen Eligible Enrollment for Product/Plan/Coverage Per Class

General Pane

The General Pane displays the name and properties of the EnrollmentScreen business rule.

Enrollment Screen General Pane



EnrollmentScreen General Pane

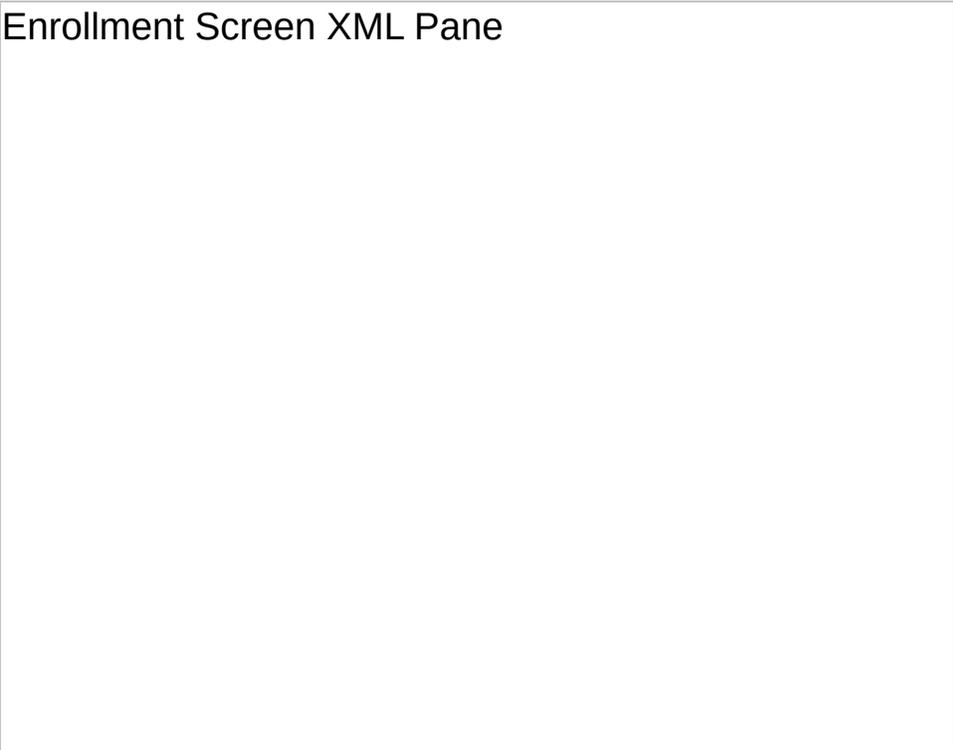
XML Source Pane

Configuration can be done in XML via the XML Source pane. Configuration for the additional screen tags must still be done through the XML Source pane.

The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the [XML Source Editing](#) section of this document.

Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Screen Rules | EnrollmentScreen**.

Enrollment Screen XML Pane



EnrollmentScreen XML Source Pane

Miscellaneous Screen Details: Security

Page Name: Enrollment

Buttons: Enroll, Activity, Save, Inquiry, Delete

Page Name: GroupCustomerRelationshipScreen (existing path to access Enrollment Screen)

Button: Enroll

Page Name: ClientSearch (existing path to access Enrollment Screen)

Button: Enroll

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Global Explorer and Business Rules > Screen Rules > FinalActionScreen

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

FinalActionScreen

A designated Final Action screen is available for underwriters to assess risk factors of an application and make a decision based on those risk factors. The underwriter decisions will be represented on the screen in the form of configurable activities. These activities will allow the underwriter to define applicable fields and make necessary changes to the Application if required based on the risk factors. This screen will house a history of the underwriter's final action decisions as well

FinalActionScreen business rule defines configured final action activities and views.

The Global Explorer contains the FinalActionScreen business rule, which can be overridden for each product and plan. Open Business Rules |Screen Rules to locate the FinalActionScreen business rule file. The following is the screenshot of the Final Action Screen in OIPA

Navigation Pane:

- CASE
- APPLICATION
- ROLES
- SEGMENTS
- COMMENTS
- REQUIREMENTS
- ASSESSMENTS
- FINAL ACTION

Application Details:

Application Name: AsFile Insert
Application Number: NB17214
Application Status: Pending
Plan Date: 2/15/2008
Company: Primary Company 1
Plan: NBU Term Life Plan

ADD FINAL ACTION

Final Actions: Select Final Action

VIEWS

Display: Shadows Reversals Delete
Refresh

View: Underwriter View

Auto-Process Process All

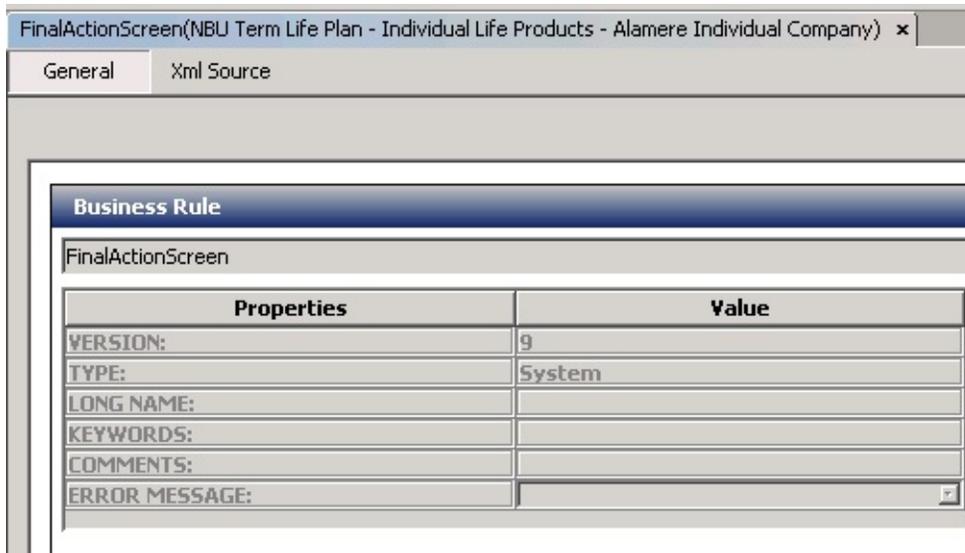
Activity	Status	Insured Name	Class	Base Face Amount	Rating	Action
Final Action	Pending	Don Draper	Standard	150000 USD	Table 2(B) 150%	
Add Offer	Active	Don Draper	Standard	150000 USD	Table 2(B) 150%	

Final Action

Effective Date: 1/1/2012
Submitted Date: 12/1/2011
Underwriter Decision: Approved
Decision Date:
Insured Name: Don Draper
Ok Cancel

General Pane

The General Pane displays the name and properties of the FinalActionScreen business rule.



FinalActionScreen General Pane

XML Source Pane

Configuration can be done in XML via the XML Source pane. The XML Source pane includes XML Editor Functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the XML Source Editing section.

Please see the XML Configuration Guide topic in this help system for a list of all elements, attributes and values needed for configuration. View Business Rules | Screen Rules | FinalActionScreen.

```
FinalActionScreen(NBU Term Life Plan - Individual Life Products - Alamere Individual Company)* x
General Xml Source
1 <FinalActionScreen>
2     <Activities>
3         <Activity>FinalAction</Activity>
4         <Activity>Offer</Activity>
5     </Activities>
6     <FinalActionViews>
7         <View NAME="ViewA">
8             <Table>
9                 <Column>
10                    <Name>Activity</Name>
11                    <Display>Activity</Display>
12                    <DataType>Text</DataType>
13                    <Group>Activity</Group>
14                </Column>
15            </Table>
16        </View>
17        <View NAME="ViewB">
18            <Table>
19                <Column>
20                    <Name>Activity</Name>
21                    <Display>Activity</Display>
22                    <DataType>Text</DataType>
23                    <Group>Activity</Group>
24                </Column>
25            </Table>
26        </View>
27    </FinalActionViews>
28 </FinalActionScreen>
```

FinalActionScreen
XML

You are here: Global Explorer and Business Rules > System Rules > Overview of System Rules

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Overview of System Rules

System rules are used to control processing in OIPA. System rules exist as global rules but can be overridden to meet the needs of individual plans. Plan overrides are on both the Global and Main Explorer. Company overrides are only accessible via the Global Explorer pane.

System rules are configured using the XML Source pane. Visual editing is not available for these rules at this time. An explanation of each rule is provided below.



Refer to the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for System rule configuration. View **Business Rules | System Rules**.

System Rules

AutomaticPolicyNumber : This business rule is a system rule that is located in the Global Explorer tab in the **Business Rules | System** folder. This business rule generates the policy number when a new policy is created.

AutoProcess: This business rule determines whether activities should process as soon as they are added or when specifically executed. If set to Yes, the AutoProcess button is checked on the Activity screen.

ChartofAccountsSpecifications: This business rule is used to filter or pinpoint the account/account entry details where the accounting should be performed, since a transaction can have more than one account and account entry details. Multiple suspense accounts can also exist, which in turn can have multiple account entry details. Using this business rule, criteria information for each (dynamic) field on the ChartOfAccountsScreen business rule can be defined. For additional information on the configuration of this rule, refer to [Steps to Set-up CoA Criteria](#).

ChartofAccountsResults: This business rule is used to write results to the AsAccountingDetailField table. This additional information is client specific and is used for various accounting purposes. For additional information on the configuration of this rule, refer to [Steps to Set-up CoA Results](#).

ClientAddressTypes: This business rule allows the user to specify which address roles appear in the Address screen's **Address Type** drop-down box when adding a new address for a client or customer. The address roles in the drop-down selection are driven by the rule's configuration to specify which address is available based on client or customer type. There are several configuration steps that must be accomplished before configuring this rule:

- Define the client types in AsCode using the code name AsCodeClientType.

Use the [Code Names](#) editor in the Admin Explorer of the Rules Palette to edit codes.

- Define the address roles in AsCode using the code name AsCodeAddressRole. Use the [Code Names](#) editor in the Admin Explorer of the Rules Palette to edit codes.
- Configure the [Address Screen](#) business rule.
- Add security for **Client Address** and **Group Customer Address** under [Company Pages](#).

When configuring the ClientAddressTypes business rule, note the following:

- The CLIENTTYPECODE attribute allows the user to specify the type code for client. The attribute value should be a code from AsCodeClientType.
- The value entered for the <ClientAddressType> element should be an address role code from AsCodeAddressRole. If a client can be assigned multiple address roles, then each role must be defined within its own <ClientAddressType> element.

[CompanyCosmetics](#): This business rule identifies several signification details that apply across a particular company.

- Defines the background to display on all screens except the login screen.
- Identifies the status code(s) that will represent the [shadow status code](#)
- Enables or disables the [ActivityShading feature](#).
- Enables or disables the Requirement Shading feature.
- Indicates that a Money data type screen or transaction field will automatically round the entered amount to the specified significant digits of the currency (DisplayRoundPlaces) using the CurrencyRoundMethod on the [AsCurrency](#) table. The <**RoundCurrencyEntry**> element controls this and accepts values of Yes and No.
 - **Yes**: value entered is rounded to the significant digits set by the DisplayRoundPlaces column on the AsCurrency table for the given currency code based on that currency's CurrencyRoundMethod.
 - **No**: value entered is truncated according to the value set in the DisplayRoundPlaces column of the AsCurrency record.

Computation: This business rule supports scheduled computation. It is executed for each policy or segment defined in the [ScheduleValuation](#) rule's Query element array when scheduled computation must be performed. The RULENAME attribute allows a user to give each computation rule a distinguishing name. There are two main sections included in this rule: Input and Output.

The Input section defines a MathVariables section containing standard math variable syntax and operations. The MathStatement element is not supported. The context of the Policy or Segment defined in the ScheduledValuation rule Query element carries over to the named computation rule. Policies will be able to execute loops of their segment records and segments will have access to their related policy records. The Query element can be used to specify the SegmentGUIDs that the Computation rule is executed against in lieu of using segment loops. Copybooks and functions with contextual overrides are supported within the Math syntax, however, the context cannot go below that of the originating transaction (i.e. Plan Activity level).

The Output section defines a Mappings section that contains the named computation fields and their data types and then maps each to a math variable from the Input section. The mapped fields are created on the AsScheduledComputationField table and associated to the record on AsScheduledComputation.

[DataDictionaryEnforcement:](#) This business rule can force users to use the [Data Dictionary](#).

DisbursementNumber : This business rule is used to configure the disbursement number format and sequence.

DuplicateClient: This business rule defines the criteria identifying a duplicate client. If it does not exist the system will not perform a check for duplicated clients. This rule has a Primary Company context and is extraordinary in that it does not resolve to the global level. The rule supports the resolution of copybooks.

DuplicatePolicy: This business rule can be configured to prevent users from entering policies with the same policy name and/or policy number as an existing policy record in the database. There are two types of messages available: warning messages and error messages. Either of these messages can be assigned to the policy name field or the policy number field to alert a user of an existing policy with the same information (warning) or prevent a user from adding a policy with the same information (error). DuplicatePolicy is currently only configurable via the XML Source Pane. For detailed information on configuring the DuplicatePolicy rule, navigate to the XML Configuration Guide in the Rules Palette help menu and search for **Configuration Overview | Business Rules | System Rules | DuplicatePolicy**.

EligibleSegmentNamesByPolicyStatus: This business rule is used to make segments selectively available to policies based on the policy status. If a particular policy status is included in the rule only the segment names referenced are available for entry for that policy status. If a policy status is not included in this business rule then all of the plan's segments are available.

GainLossCalculation: Gain/Loss is the gain or loss incurred by a company when a financial activity buys and/or sells variable fund units. This business rule determines whether or not the system should calculate Gain/Loss for the back-dated transactions. For example, if a transaction processed on a date that is a few days after the effective date of the transaction, the difference in price would be calculated as a gain or loss to the company.

There are two columns in [AsValuation table](#) that are used to store Gain/Loss information.

- ValuationGainLoss: ValuationGainLoss is populated during forward activity processing.
- GainLossOnShadow: GainLossOnShadow is populated when a financial activity is reversed.

InterestRateCalculation: This business rule defines the type of interest

calculation as well as the specific details of the calculation for fixed funds. Methods of interest rate calculation can be assigned to fixed funds using this rule. In most cases, this rule is overridden on a specific fund level.

[View interest rate prototype example.](#)

Multifield: This business rule defines the main combo box where the number of multifields is determined. It is like a copybook, where it can be called from various transactions or screens and there may be many MultiField business rules. It works in conjunction with a Multifield transaction, which calls the Multifield business rule.

PlanOrder: PlanOrder configuration will define the sort order for all screens that have a system defined Plan and Plan Group drop down box. The required element <OrderBy> will be used to define the ordering basis.

For plans, this basis will be AsPlan columns or dynamic plan fields. If a fieldname is used that is not applicable to all plans, then the plans where the field is not applicable will appear alphabetically at the end of the drop down list.

For PlanGroups, the ordering basis will be the PlanName column of AsPlan. An alphabetical sort order will be applied to PlanName.

ORDER is an optional attribute of the <OrderBy> element. The valid values of this attribute are ASCENDING or DESCENDING. An ascending sort order is the default order if the ORDER attribute is absent from the configuration.

The following screens contain a Plan or Plan Group drop down box:

- StateApproval
- OIPA Table Views (Plan Withholding, Plan Fields, Plan Allocations and Unit Value Table)
- PolicySearch
- PolicyScreen
- DisbursementSearch

- Plan Activity
- Company Activity
- DisbursementApproval

PoinInTimeValuation : This business rule is configured to indicate how often valuation records may be saved to the database. Options include writing valuation records at the beginning of an activity and conditions to write valuation records at the end of an activity. It also directs the display of a beginning valuation area on the Valuation tab of the Activity Results screen.

PrecisionValues : The PrecisionValues business rule provides rounding precision convention (i.e. number of digits allowed after the decimal point) for the display of Unit Values and for the calculation and display of Number of Units. Unit Values and Number of Units are displayed on the following OIPA screens:

- Policy Screen's Values link
- Activity Result Screen's Allocation link
- Activity Result Screen's Valuation link

This rule can be overridden at the company and plan levels.



As of the 9.5.0.0 release, the PrecisionValues rule performs the functionality of the UnitValuePrecision rule, which is deprecated.

RedemptionAmountFormula: This rule defines the calculations that must be performed to determine the redemption fee associated with a withdrawal activity. Redemption fees can be applied to a withdrawal if the deposit has not yet matured at the time of withdrawal. This rule must reference the redemption dynamic fields in the FundScreen rule, using the exact name.

TransactionTimes : The TransactionTimes business rule controls how

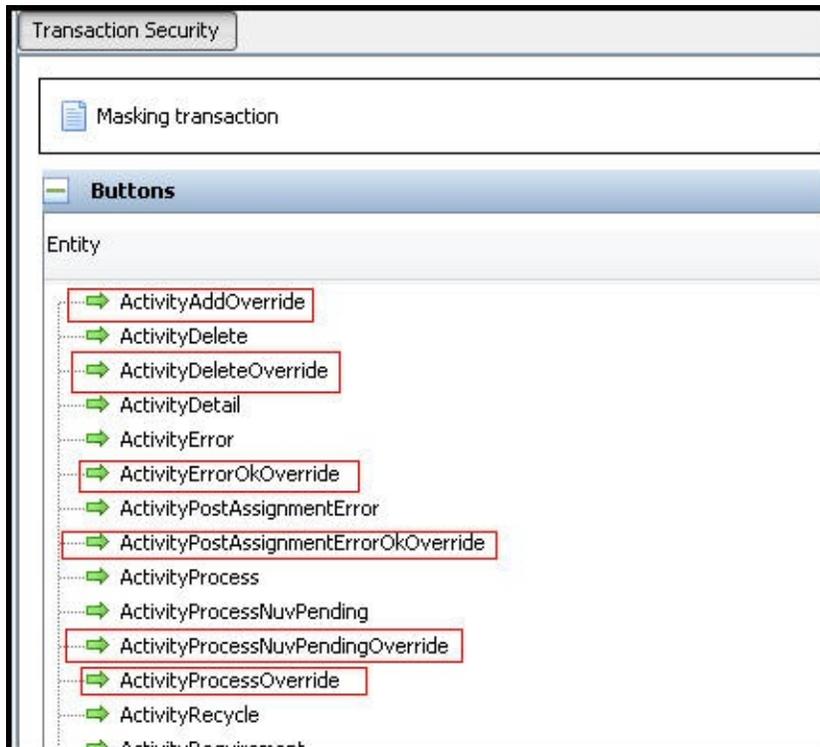
an activity behaves based on the time of day it is processed. The rule is attached to transactions and any portion of the rule may be configured as a CopyBook overridden at the Company or Plan levels. TransactionTimes also exists as a System Rule at the Global, Company, and Plan level.

TransactionTimes provides two primary elements for controlling whether or not a transaction may process when added or updated: **UpdateDate** and **Allow**. Both elements have a common optional attribute called MESSAGE that displays a configured message if any of the validation criteria fail for the primary element. If the value of the message attribute is equal to an entry found on the translation table, the localized (translated) version of the message is displayed to the user; otherwise, the literal value of the message attribute is displayed. If no message attribute exists, then a system message displays a default error message: *Screen alert message is not defined*. Unless otherwise indicated, the message appears at the top of the Activity Detail screen. To control the messages that display, the system processes in the order UpdateDate and Allow. Failure alert messages are displayed in the order that the associated validations are processed.

The TYPE attribute for the **Allow** and **UpdateDate** elements determines when the associated validation is applied based on the screen action being performed.

- Add: invokes the validation when the activity is first added to the system.
 - Delete: invokes the validation when the activity is deleted.
 - Process: invokes the validation when the activity is processed.
 - Recycle: invokes the validation when the activity is recycled.
 - Reverse: invokes the validation when the activity is reversed.
 - Update: invokes the validation when the activity is re-opened from a pending status or during the re-display of the activity when recycling.
- Only one Allow element and one UpdateDate may be added for each action type attribute.

[Security group Override buttons](#) for the **Allow** and **UpdateDate** elements determine which user security groups are exempt from processing the validation. Override buttons are available for Add, Delete, Process, Reverse, and Update. The system checks permissions for the current user's security group and the list. If the group has the proper override permission checked, then the timing validation is not executed.



Override Buttons in Transaction Security

The **UpdateDate** element always targets the EffectiveDate field on the activity. The VALUE attribute identifies the source date value. The source date is always derived from the system date record for the Plan's MarketMaker calendar. Custom values are not supported. The available date values are:

- NextSystemDate
- NextBusinessDate
- NextMonthEndDate
- NextQuarterEndDate
- NextYearEndDate.

Another attribute on **UpdateDate** is AUTO, which allows a value of **Yes** or **No** (the default being No). If the attribute is set to Yes and the validation fails, the failure alert message is displayed and the effective date is changed to the VALUE attribute's date reference. Clicking **OK** again on the activity's detail screen saves the activity and closes the detail screen if there are no alert messages from other timing validations. If the AUTO attribute value is set to No and the validation fails, the failure alert message displays as a confirmation with an OK and Cancel option. Selecting the confirmation's OK sets the effective date to the VALUE attribute's date and closes the alert message. If there are no more alert messages, the system saves the activity and closes the activity detail screen. Selecting the confirmation's Cancel closes the alert message and if there are no more alert messages, returns to the Activity Detail screen for more editing without changing the effective date.

The Allow element controls whether or not the activity can be saved based on the time date validation criteria. If all the criteria are true, the activity is saved.

ValuationDetails: This rule allows the configurator to set the proper rounding in valuation calculations for deposits.

WriteValuationElements: This business rule limits the amount of valuation information that is written. Existing valuation fields can be written to the system and are divided into sections: Policy, Fund, Deposit, and PolicyValues. Inside of these sections the available elements are from a set list. The only exception is the Policy Values section, which can be configured to access any variables from the Policy Values business rule. CopyBooks may be used in this rule. The rule can be overridden by Plan.

You are here: Data Dictionary > Data Dictionary Enforcement

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Data Dictionary Enforcement

The Rules Palette can be configured to enforce the use of the [Data Dictionary](#) through the DataDictionaryEnforcement business rule. Configuration is done via XML. If enforcement is needed, make sure Compliance="Yes" is placed in the opening <DataDictionaryEnforcement> tag.



Data Dictionary Compliance XML

Compliance Tag Options:

- **No:** Data Dictionary enforcement is turned off for the environment.
- **Yes:** Data Dictionary enforcement is turned on for the environment. Additional compliance is set through the <Math> tag.

Compliance Math Values

- **Ignore:** All fields and logged math variables must be defined in the Data Dictionary in order to check in a rule. Non-logged math variables are not validated against the Data Dictionary and are not required to be in the dictionary for check in of the rule.
- **Warning:** All fields and logged math variables must be defined in the Data Dictionary in order to check in a rule. Non-logged math variables will be presented if they are not defined in the Data Dictionary, but are not required to be in the Data Dictionary for check in of the rule.
- **Denied:** All fields, logged and non-logged math variables need to be defined in the Data Dictionary in order to check in a rule.

Steps to Set Data Dictionary Enforcement

1. Locate the DataDictionaryEnforcement business rule in the Global Rules Explorer tab. It can be found in the **Business Rules | System** folder.
2. Configure the DataDictionaryEnforcement business rule.
3. Select **Check-In** from the right-click menu to save your changes to the database.

Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | System Rules | DataDictionaryEnforcement**.

You are here: Global Explorer and Business Rules > System Validations > CheckEFTInformation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

CheckEFTInformation

This business rule is used by Disbursement processing to verify an EFT address before disbursing money to it.

Note: You must have the syntax **<Disbursement CheckEFTInformation="Yes" ...>** in the XML of the disbursement transaction in order for the system to check the EFT information.

Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | System Rules | CheckEFTInformation**.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Group Information > Group Customer Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Group Customer Overview

Group Customer

Group Insurance provides a single insurance policy that covers groups of employees and their dependents on behalf of the company or organization to which they belong. The company or organization is referred to as a **Group Customer** in the Oracle Insurance Group Policy Administration system.

The display of the Group Customer screen in OIPA is controlled through the configuration in the GroupCustomerScreen business rule.

Agreements

Based on the organization's needs, the terms of the Group insurance [product](#) are negotiated and Group Customer enters into an **agreement** with the Insurance Carrier. The agreement provides information on the contract between the Carrier and the Group Customer detailing the plan offerings available to members. Agreements may be funding, financial, or service related and will not have plan offerings or coverage attached. New plans can be created within the Plans tab under a specific Master Agreement. See [Agreements](#) for information on configuring agreements.

Relationships

The Group Customer has **relationships** with clients and non-group customer entities. The ability to connect or link the Group Customer to employees, contacts, Human Resource Administrators, Account Managers and other affiliations is key in establishing relationships within OIPA. Relationships can be designated as primary or secondary. A primary relationship type describes the general point of contact. The secondary relationship further defines the specific title and characteristics of the relationship. For example, the primary relationship may be an employee. The secondary relationship further defines the employee as Full-Time, Part-Time or Retired. Users can add and maintain relationships in OIPA. The type of relationship drives what activities (Client-Relationship activities) are eligible for processing for the Group Customer relationship. See [Relationships](#) for information on configuring relationships.

Classes and Class Groups

The Group Customer selects the product and plan offerings to offer various subsets, or **Classes**, of its overall employee population. Classes provide a way for the employee population to be divided into groups based on similar characteristics the employees share.

Examples of classes include:

- Full-Time
- Part-Time
- Union
- Non-Union
- Management
- Non-Management
- Location (Payroll location)
- Active Employees
- Retired Employees

Classes may be arranged into a collection called a **Class Group** based on the Group Customer's needs. See [Classes](#) for information on configuring classes and class groups.

Plan Eligibility and Enrollment

Eligibility provides the stipulated requirements that mark a class member as qualified to enroll in a plan. Based on parameters and rules set by the Group Customer, plans are filtered and made available to the members at a particular point in time, via classes. As a member's characteristics change over time (salary change, job title or status qualifying life event) the member may become eligible for different plans.

Enrollment is the act of selecting coverage for members. The enrollment period can be any of the following types:

- **Open:** specified number of months to enroll and guaranteed issue/no underwriting.
- **Relaxed:** set number of questions, relaxed underwriting
- **Evergreen:** no closed period, can enroll at any time, no relaxed underwriting rules
- **Qualified Life Event:** Open 30 day window for the member
- **Auto Enrollment:** Noncontributory benefits that do not require enrollment by the member.

Data Intake

Carriers must be able to receive, process and apply Group Customer Data Intake files, which provide employee and policy coverage data. The files include additions, changes and deletions pertaining to both employee and coverage data. The employee information as well as the coverage information will need to be updated. This is achieved through the use of a transaction at the Group Customer level where the receipt of the intake file triggers and produces individual transactions at the policy/certificate level. The process is housed on the Data Intake screen. The transactions may be viewed, filtered or reversed. Users can also drill down into member record level details.

You are here: Group Information > Group Customer Activity Processing

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Activities_Screen_Group Customer

The Group Customer Activity screen will give the ability to initiate Group Customer level activities. Since a Group Customer is essentially a client, the Customer Activity Screen functionality will mirror the current Client Activity Screen capabilities.

After a Group Customer is selected via Group Customer Search screen ,the navigation list will be displayed on the left side of the Group Customer Screen. The list currently includes the following links, each of which navigates to a separate screen – CUSTOMER (main), CUSTOMER ADDRESSES, RELATIONSHIPS, AGREEMENTS, CUSTOMER ACTIVITIES, CLASS GROUPS

GroupCustomerScreen

The Group Customer screen business rule will be configured to specify the appropriate Customer Plan (i.e. system convention to differentiate from actual benefits plan) for the Group Customer Client Type using the ActivityPlan attribute.

The ActivityPlan attribute is defined in the GroupCustomerScreen rule to specify the name of the Customer Plan where the activities have been defined.

As stated above, the functionality of the Customer Activity Screen will mirror the existing Client Activity Screen.

All transactions configured for the Customer Plan is available for selection and processing (subject to user security role/access rights) on all Group Customers Existing (i.e. previously submitted pending and processed) Group Customer level activities can be viewed on the Customer Activity Screen by clicking the Customer Activities navigation link.

Only those pending/processed Activities which belong to a specific Group Customer will appear in the Activity History for that Group Customer

New Group Customer level activities for the current Group Customer can be initiated by clicking the **Add Activity** button below the Main Menu bar.

The **New Activity** Pop-Up window appears for user to select, input and submit appropriate Group Customer activity (transaction) for processing

The following Group Customer fixed columns/field should be available for activity processing

GroupCustomer : CustomerNumber

All dynamic fields configured for Group Customer Client are available. To view the Group Customer, there is a left side navigation menu for **Customer Activities** as well as the **Add Activity** tab at the top of the screen.

Policy activities can be accessed by clicking the Activities link on the Policy screen. Client/Group Customer activities can be accessed by clicking the Activities/Customer Activities link on the Client/Group Customer screen. Plan activities can be accessed by opening the Plan screen from the Plan menu.

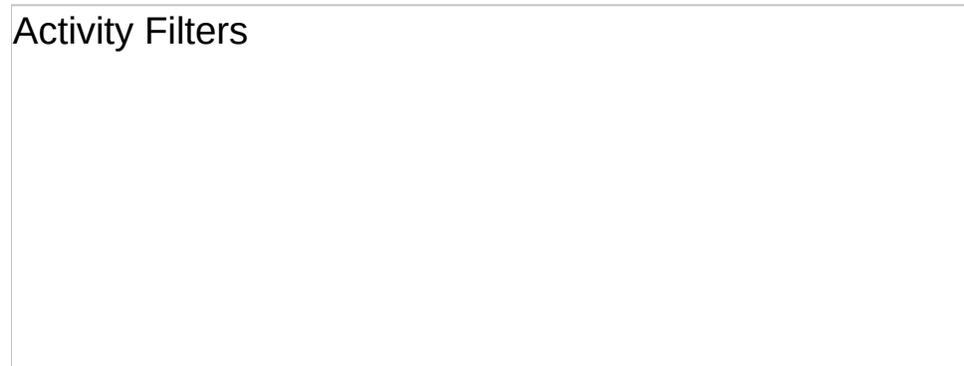
Group Customer Activities Link



Links to Activity Screen

Activities can be filtered on the Activity screen by selecting an activity status filter option. Activities will only display if their activity status filter is checked. The system default is to show all document and financial

activities. If an activity is a shadow, reversal, or future activity it will not display unless the filter is checked to show these items. An explanation of filter options is given below.



Filters on Activity Screen

- **Shadows:** This filter will only display activities that have been removed from the Activity screen.
- **Reversals:** This filter will only display activities that have been reversed.
- **Futures:** This filter will only display activities that are set to process on a future date.
- **Documents:** This filter will only display activities that are defined as documents, such as a confirmation letter, an annual statement. or a lapse letter. This is a default option.
- **Financials:** This filter will only display activities that are defined as financial. This is a default option.

Activities are processed either manually using the action buttons

 to the right of the activity or automatically using the

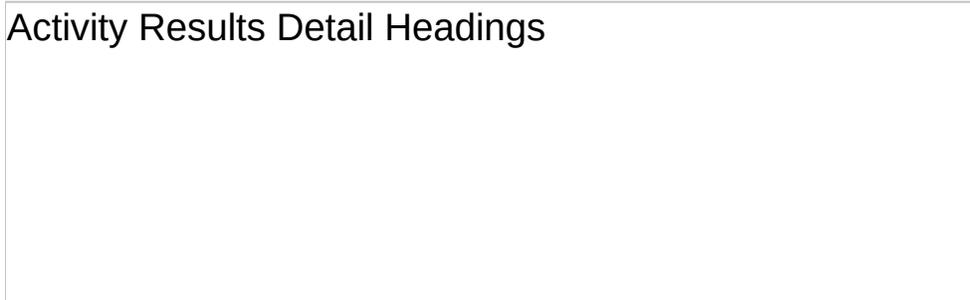
Auto-Process **Auto-Process** checkbox. If that box is checked, then the system will attempt to process all pending activities up to the current date, or activities that will be updated due to a change in a dependent activity.

View the details of an activity by clicking the **Activity Detail icon** to the left of the activity. A window will open with links to all details that apply to that activity.



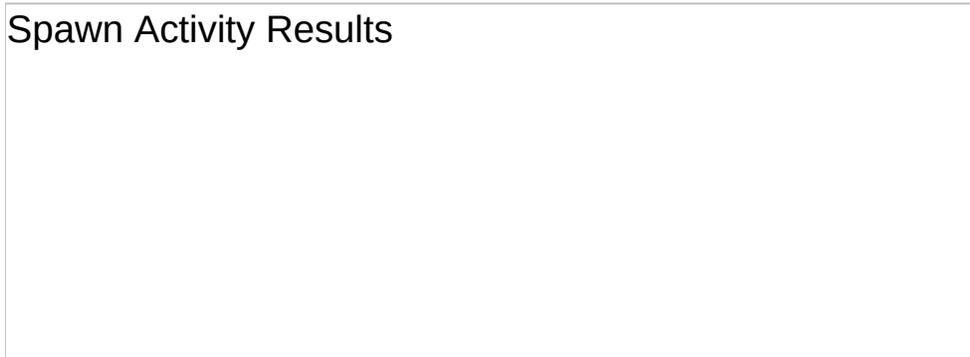
Activity Detail Icon Next to Activity on Activity Screen

When the Activity Detail window opens, click a link along the top of the window to see the details for that particular activity. There will be links for details such as Spawns, Accounting, Valuation, Disbursement, Suspense, Allocations, Math and Entry Fields.



Links Along the Top of the Activity Detail Window

The image below shows the detail that appears in the Activity Detail window when the Spawn link is clicked.



Spawn Details

Miscellaneous Screen Details: Security

Page Name: Group Customer Activity

Buttons: GroupCustomerActivity, AddActivity, Inquiry, AutoProcess,

ProcessAll

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Group Information > Alternate Names

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Alternate Names

Customers can have multiple alternate names in OIPA. A user can add, update and run inquiries on customer alternate names. The status of alternate names is also recorded in OIPA.



Alternate Name Link in OIPA

Configuration Requirements for Alternate Names

The Rules Palette establishes the configuration of alternate names. There are several steps involved in preparing alternate name functionality for use in OIPA. They are outlined below.

- Enter all alternate name types and available statuses in the AsCode table.
 - **AsCodeClientAltIdStatus:** contains a list of all available statuses to assign to alternate names.
 - **AsCodeAltIdTypeCode:** contains a list of all available types of alternate names.
- Configure the [Alternate Name](#) screen rule. If a canceled alternate name should automatically capture the system date when canceled, then the following configuration should be added to the rule.
 - The field for status code must be configured within the Fields section of the rule.
 - An OnChange validation event must be configured to trigger the action.
 - Global Screen Math must be configured to pull the Status Code field value into a math variable for the retrieval of the Status Code option value. In addition, a global math variable must be configured for the date in which the Effective To date will be assigned.
 - An ActionSet for the OnChange validation must be configured to assign the EffectiveTo date to the current date (EffectiveDate) based on the Status Code field condition where the status code equals that of a canceled name entry.
- Configure the [CreateClientAltId](#) business rule, which can be attached to a transaction to create an instance of an alternate name for a client when the activity is processed in OIPA.
- Configure a [CopyTo](#) business rule using ClientAltId as the object name to update the details of an alternate name when an activity is processed.
- Configure the [ClientSearchScreen](#) and

[GroupCustomerSearchScreen](#) business rules to search by alternate names is alternate name search support is needed.

- Add security to the Alternate Name screen. This is a [Company Page](#) in the Admin Explorer Application Security section.

Database Tables

There is one database table used to store alternate name information.

AsClientAltID: relates the alternate name to the group customer.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Group Information > Group Customer Addresses and Phone Numbers

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Group Customer Addresses and Phone Numbers

Addresses and phone numbers can be captured at various levels in OIPA. At the Group Customer level, addresses and phone numbers are captured from the **Customer Address** link on the Group Customer screen in OIPA. Through configuration, specific roles can be assigned to group customer addresses to distinguish them by type. Once an address is assigned a role, it can then also be restricted to specific customer types.

Phone number formats can also be controlled through configuration to display differently based on the country and type of phone number.



Phone number displays can also be configured for the [Address screen](#).

Address Roles

The [ClientAddressTypes](#) business rule allows the user to define the address roles that are available for the client and customer address based on the client type.

If this rule is not configured, then all address role types will be present in the Address Type drop down box on the Address screen.

If this rule is configured, but a given client type is not referenced in the configuration, then all address types that are applicable for the CountryCode specified in the corresponding AddressScreen business rule will be displayed for that client type.

The rule can be overridden at the Company and Group Customer level.

Configure Group Customer Addresses

There are several steps involved in configuring Group Customer addresses.

- Define the Client types in **AsCodeClientType**
- Define the address roles in **AsCodeAddressRole**.
- Configure the [AddressScreen](#) business rule.
- Configure the [ClientAddressType](#) business rule. This defines the address types that will be available in the Address drop down list when a new address is created.
- Add security for the **Client Address** and **Group Customer Address** pages under [Company Pages](#).

Phone Numbers

Phone number formats can be controlled through configuration to display differently based on the country and type of phone number. A separate Phone screen is available in OIPA to capture phone number information. Phone numbers can be associated to client addresses.

Configure Phone Number Formats

There are several steps involved in configuring the formats available for displaying phone numbers in OIPA.

- Define the phone types in **AsCodePhoneType**. The [Code Names](#) editor in the Rules Palette can be used to accomplish this.
- Define all applicable countries in the [Country editor](#) in the Rules Palette. These values are stored in the AsCountry table. Calling codes are also stored in the AsCountry table, but must be added directly in the database.
- Add **<DisplayPhoneScreen>Yes</DisplayPhoneScreen>** configuration to the AddressScreen and ClientScreen business rules to allow the Phone screen to be available on those screens. This element and value can be added at the beginning of the rule. If this configuration is added to the ClientScreen, then it will also be available on the Group Customer screen.
- Configure the [PhoneScreen](#) business rule at the appropriate company level.
- Add security to the [Phone](#) page in Company Pages.

You are here: Group Information > Relationships > Relationships

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Relationships

Client Relationship is a direct association of two Clients outside of any prerequisite context. Either Client in the Relationship may be a Group Customer or non-Group Customer (e.g. company, individual, trustee, etc.) Client.

A client (Group Customer or non-Group Customer) can have an unlimited number of primary and secondary relationships. The relationships of the given primary relationship type with the current customer (Group Customer or non-Group Customer) can have one or several secondary relationship types. The Relationship screen in OIPA provides the user with the capability to add and update relationships and their associated information during case implementation. Configuration for the Relationship screen can set a maximum number of relationship by using the <Maximum> attribute in the ClientRelationshipScreen business rule.

The ClientRelationship screen gets displayed in two modes from the one business rule configured under ClientRelationshipScreen BR - 1. Group Customer Relationship: when used in group customer context, the security set up for this screen would be done through the GroupCustomerRelationship node and 2. Client Relationship: When used in an non-Group Customer context, the security set up for this screen would be done through the ClientRelationship node.

Configuration Required for Relationships

There are three steps involved in configuring relationships. First, the AsCode table needs to contain the primary and secondary relationship types. Next, the ClientRelationshipScreen business rule must be configured to control the Relationship screen in OIPA. And finally, transactions must be configured to allow changes to be made to relationships in OIPA.

AsCode Table

Relationship types are defined in the AsCode table. Both primary and secondary relationships must be defined.

- **AsCodePrimaryRelationshipType:** The primary relationship types are defined here. These types will be available to the user when a new primary relationship is created in OIPA.
- **AsCodeSecondaryRelationshipType:** The secondary relationship types are defined here. These types will be available to the user when a secondary relationship is created in OIPA.
- **AsCodeChangeStatus:** The system codes that define the record status of the relationship.
- **AsCodeBusinessStatus:** The system codes that define the business status of the relationship. This is applicable only if the UseBusinessStatus element is set to Yes in the ClientRelationshipScreen business rule.

Client types are also referenced in the ClientRelationshipScreen business rule, so values must exist for the AsCodeClientType code name. The codes are referenced in the business rule to identify the client types that can be assigned a secondary relationship.

Business Rules

The Relationship screen is controlled by the [ClientRelationshipScreen](#) business rule. This business rule identifies the primary relationship types

that are allowed and the secondary relationships that can be created with each primary relationship type. It also defines the client types that can be assigned a secondary relationship. The <Maximum> element in this rule can set a maximum number of primary and secondary relationships. This rule also defines the activity that needs to be spawned when a client relationship time slice is submitted. The transaction can either be defined at each PrimaryRelationship level or at the ClientRelationship level. The ClientRelationship spawn configuration will be used, by default, if no specific spawn configuration exists at PrimaryRelationship level.

Transactions

The transaction(s) that will be spawned when a client relationship time slice will need to be defined under the Customer Activity Plan relating to the Group Customer and to the Client Activity Plan relating to the non-Group Customer Client. The type code of the activity should be 26 - Client-RelationshipScreenUpdate.

DataBase Tables

The following database tables hold the relationship information.

AsClientRelationship: Stores the data and values for the connection between two clients.

AsClientRelationshipField: Stores the data and values for the field names related to Client Relationship details and characteristics.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Group Information > Agreements > Agreement Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Agreement Overview

Agreements house the characteristics and information regarding the contract between a Carrier and the Group Customer. The Group Customer and Carrier define the agreement.

The agreement details may include plan offerings for members, coverage availability criteria and service and billing details for the Group Customer and the Carrier.

The Group Customer can have multiple agreements based on the business needs surrounding servicing, billing and plan administration.



Agreements may also be structured in a parent and child relationship. Please see Parent and Child Agreements for more information.

The agreement will have roles attached for the purpose of designating a primary point of contact, a service contact and an administrator. An agreement **does not** require that a plan(s) be associated to the agreement. There are agreements specific to funding, financial, data intake and performance guarantee arrangements that are exclusive of a plan. Please see [Agreement Roles](#) for more details.

Configuring Agreements

There are several steps involved in setting up agreements.

- Define all agreement types in AsCodeAgreementType and all agreement type statuses in AsCodeAgreementStatus. This can be done in the Rules Palette using the [Code Names](#) editor in **Admin Explorer | Administration**.
- Define all agreement roles in AsCodeAgreementRoleTypes and all role statuses in AsCodeAgreementRoleStatus. This can be done in the Rules Palette using the [Code Names](#) editor in **Admin Explorer | Administration**
- Configure the Agreement Definition for each type of Agreement.

- Configure the [AgreementScreen](#) business rule and define the parent/child relationship between agreements using the [AgreementRelationship](#) business rule.
- Configure the [AgreementRoleScreen](#) business rule.
- Add security to all Agreement pages through [Company Pages](#) security.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Group Information > Agreements > Agreement Roles

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Agreement Roles

Agreement roles are defined in AsCode within the code name AsCodeAgreementRoleType. The AgreementDefinition business rule provides the method for associating roles to agreement types. Each agreement type listed in the business rule has an element called EligibleRoleTypes. The value listed for this attribute tells OIPA which role(s) to allow on the agreement.

The [AgreementRoleScreen](#) business rule tells OIPA what to display on the Roles tab of the Agreement screen. The fields that will be used to capture the role data are defined in this business rule.

The [AddAgreementRoles](#) business rule can be attached to a transaction to add roles to agreements via activity processing.

You are here: Group Information > Group Customer Plans > Plan Screen Overrides

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Plan Screen Overrides

The Plan screen can be overridden at several levels. An explanation of the various levels are provided below:

- **Product Overrides:** This override is selected in OIPA when a user is on the Plan screen and selects a Product from the Product drop down list. If an override exists for that product, then the Plan screen will load the override configuration.
 - Configuration Requirements:
 - Create a Product at Subsidiary Company level.
 - Create a PlanScreen Override. Select a Subsidiary Company and Sub-Product.
- **Child Product Overrides:** This override is selected in OIPA when a user is on the Plan screen and selects a Child Product from the Child Product drop down list. If an override exists for that sub-product, then the Plan screen will load the override configuration
 - Configuration Requirements:
 - Create a Child Product at the Subsidiary Company level.
 - Create a PlanScreen Override. Select a Subsidiary Company and Sub-Product.
- **Group Customer Overrides:** This override is used in OIPA when the ClientScreen business rule defines the plan associated with the Group Customer client type and that plan has a Plan screen override.
 - Configuration Requirements
 - Create the Plan screen override at Group Customer level.
 - Modify the ClientScreen rule <Client> element with a CLIENTTYPECODE of 20 (Group Customer). The ACTIVITYPLAN attribute should be the name of the plan to override.

Configuration that is split between different override levels of the PlanScreen business rule can be shared, which eliminates the need to

duplicate rules.

- If a field is defined at only the Product level, then that field will be displayed for all plans related to that product.
- If a field is defined at the Product and Child Product level, then the Child Product level override of the field will be displayed for all plans that are related to that Child Product.
- If a field is defined at the Product, Child Product and Plan levels, then the Plan level override of the field will be displayed for that plan only.

Plan is the lowest override level available. Fields will be displayed at the lowest level configured. When a relationship has been established linking a Product, Child Product and Plan, all of the unique fields defined in all three rules will be displayed on the Plan screen for a selected plan.

When the Plan screen is loaded through the UI the system will gather all of the fields defined at the Product, Child Product, and Plan level and display the highest overridden instance of each.

Please see the **XML Configuration Guide** topic in the **Help** menu for a list of all elements, attributes and values needed for configuration. View **Business Rules | Screen Rules | PlanScreen**.

You are here: Group Information > Classes and Class Groups

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Classes and Class Groups

Classes represent a distinct subset of employees and their dependents. The Group Customer can divide members into different classes for reporting, eligibility or billing purposes. Members can also be divided into classes by associations such as Management, Non-Management, Union, Non-Union, Full-Time, Part-Time and so forth.

Multiple classes can be grouped together into a larger selection called a Class Group. Classes are defined with a parent or child relationship to the class group. In addition, a new time slice for the existing class group can also be added and can use the "Show Diff" option to select and compare two timeslices of same class group. Proper security must be provided for the security group in order to perform class group specific operations

Configuration Requirements for Class and Class Group

There are several codes that must be defined in the AsCode table before configuring classes and class groups.

- **AsCodeClassType:** this defines the class types that can be assigned to a class.
- **AsCodeClassGroupType:** this defines the class group types that can be assigned to a class group.
- **AsCodeClassPlanType:** this defines the class plan types that can be assigned to a class Plan.
- **AsCodeClassSegmentNameType:** this defines the class segment types that can be assigned to a class Plan.

There are several business rules that control class information in OIPA. These rules can be configured to control the display of class and class group information. For more information on the XML configuration of these rules, see the relevant pages in the XML Configuration Guide by navigating to **Configuration Overview | Business Rules | Screen Rules**.

- **ClassScreen:** This business rule defines the fields that are used to capture class information when a new class is entered in OIPA. This screen is accessed from the Group Customer screen when the user selects the **Class Group** left navigation link, clicks the **Classes** tab at the bottom of the screen and selects a class from the tree navigation structure.
- **ClassGroupScreen:** This business rule defines the fields that are used to capture class group information. This also defines the various tabs that will be displayed on the screen and whether business status will be used for Class Group records. Further, this business rule will also identify the screen update activity that need to be spawned when a class group time slice is submitted. The tabs available below within the Class Group section are: Class Group Detail, Classes and ClassRuleVariables. The tabs available below within the Class Group section are: Class Group Detail and Classes.

- **PlanCoverageScreen:** This business rule defines the fields that are used to capture details of a new class plan. This screen is accessed by clicking on the Sub-Plan Details tab of the Class Sub-Plans screen. The only configurable aspect of this screen is the table in which the sub-plan records display.

- **PlanSegmentNameClassScreen:** The PlanSegmentNameClassScreen business rule controls the configuration of the Class Sub-Plans screen, which displays the Class Sub-Plan associations for a selected class. This screen will have three tabs available: Class Segment Details, Segment Details, Class Segment Participants.
 - The Segment details houses the details of the association. The fields are TypeCode, Effective Date and Expiry Date.
 - Segment Details houses the details of the Plan Coverage/Segment (Sub-Plan). The tab information is presented in read-only format.
 - The Class Segment Participants contains the participants enrolled in the Sub-Plan. The **PlanSegmentNameClassParticipantsScreen** business rule controls the data that displays on this tab. The Sub-Plan participants are determined based on policies that have a Plan Coverage Segment with the role of Sponsor (Primary Member). Sponsor is configurable within the Company Cosmetics business rule. The tab information is presented in read-only format.

Code-Generated Class Screens

In addition to the configurable business rule explained above, there are also two class-related screens that, while not configurable in the Rules Palette, can be used to configure classes themselves.

Class Rule Screen

The Class Rule screen allows users to add membership criteria (i.e. rules) to classes within a class group. These rules are like the global rules that are valid across all classes within the class group.

On the Class Rule screen, users will be able to view the rule content of the class, as well as that of the class' parent classes. Each child class has its own independent membership rule(s); however, the members in each child class must first satisfy the membership criteria for each parent class above it within the class group hierarchy in order to be subject to the child class' criteria.

The Class Rule screen is accessed in OIPA by selecting a specific class in the class hierarchy on the Class Group screen, then selecting the **Class Rules** tab from the Class Details pop-up screen.

Class Rule Variable Screen

The Class Rule Variable screen allows a user to configure variables for referencing employee/employment related attributes required to determine class membership. Although this screen is not configurable in the Rules Palette, the level at which it is accessed will create different override levels for the screen and the variables it is used to define, similar to how different override levels can be created in the Rules Palette. The screen will also display differently depending on the level at which it is accessed. The Class Rule Variable screen is accessible—and therefore overrides levels can be created—at the following levels:

- Global—for definition of variables. The global level Class Rule Variable

screen will be the default screen displayed if lower-level overrides are not defined.

- Group Customer—for definition of additional variables and overrides of variables defined at the Global level
- Class Group—for definition of additional variables and overrides of variables defined at the Global and Group Customer levels
- Class—for use of all variables defined or overridden at any of the above levels.

Each level of the Class Rule Variable screen is accessible as follows:

- Global—by selecting **Rule** from OIPA's main menu, and then selecting **Class Rule Variables**. As long as the user has appropriate security access, this menu option will be available regardless of what screen the user is viewing.
- Group Customer—by selecting the **Class Rule Variables** link in OIPA's secondary menu. This link will be available when the user is viewing a screen within the context of an existing Group Customer.
- Class Group—by selecting a specific class in the class hierarchy on the Class Group screen, then selecting the **Class Rule Variables** tab from the Class Details pop-up screen.
- Class—by following the same navigation used for Class Group, since the process involves selecting an individual class.

Class Rule Math Operators

The following mathematical operators can be used to construct class rules on both the Class Rule screen and Class Rule Variable screen.

Operator	Description
And	Used for logical conjunctions returning true or false. (Can also be used as a bitwise operator ‐ bit by bit 1 and 0=0, 1 and 1=1).
Mod	Divides two numbers and returns the remainder.
Not	Used to perform a logical negation of an expression.
Or	Used for logical disjunctions returning true or false. (Can also be used as a bitwise operator ‐ bit by

	bit 1 or 0=1, 0 or 0=0.)
()	Parentheses for ordering/nesting within expression.
=	Compares values for equivalence.
<	Compares values for less than. Use in Rules Palette.
>	Compares values for greater than. Use in Rules Palette.
<=	Compares values for less than or equal to. Use in Rules Palette.
>=	Compares values for more than or equal to. Use in Rules Palette.
<>	Compares values for non-equivalence. Use in Rules Palette.
+	Add two numeric values.
-	Subtracts a number from another.
/	Divides a number by another.
\	Integer divide.
*	Multiplies two numbers.
^	Exponential.

Class Groups and Agreements

Under the Group Customer context, the Agreement links a Class Group to the Group Customer. Creation of new class groups are done through an Agreement linkage with effective from and effective to dates. Under the Agreement Definition, the CLASSGROUP attribute defines whether one or more Class Groups can be linked to the Group Customer through the specific Agreement Type.

You are here: Group Information > Enrollment

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Enrollment Screen Business Rule

Employee / members of a particular [class](#) will have the opportunity to enroll in benefits offered by the Group Customer. The determination of eligibility for the employee/member is often based on employment status (full-time, part-time, salaried, or hourly), other rules that may be defined for eligibility and is specific to a class based on or created for the purpose of Eligibility.

Employees are individuals employed by the Group Customer (who may be subject to further qualification such as employment status, as stated above).

Members are those individuals who are not employees, but are individuals belonging to a group, association, or organization such as an alumni, church, or trade association.

The Group Customer divides the population into Classes and an employee can become a Member of a Class.

Membership is defined and executed using Rules (Class Rules and Class Rule Variables) within OIPA. Membership Rules are defined by the Group Customer. The rules define the logic and criteria for which an individual becomes a member of a Class.



A Member may belong to a Class outside of the realm for Enrollment. The Class may be specific to billing, reporting, etc... and not specific to Eligibility and/or Enrollment.

Enrollment is the act performed by an eligible member / employee of the Group Customer where benefits are selected and dependents / beneficiaries are designated. The Enrollment Screen allows for the

manual entry of enrollments for a Group Customer.

Logic and rules defined by the Group Customer are enforced during the enrollment process. An enrollment period designates the timeframe in which the member has to register for coverage(s).

A policy certificate will be issued to the participant upon completion of the enrollment and approval of the coverage(s). The participant that has enrolled in a group plan has the right to elect dependents and beneficiaries.

The policy owner for group insurance is the Group Customer.

How it Works

A member or employee will meet the stipulated requirements set forth by the Group Customer in order to participate and/or qualify for coverage/benefits. Based on the parameters defined by the Group Customer, particular plans are offered to the member/employee at a particular point in time (enrollment period).

Enrollment periods:

- Open: specific number of months for enrollment
- Qualified Life Event: open window based on marriage, divorce, birth of a child
- Auto: Noncontributory benefits that do not require enrollment by the member

You are here: Group Information > Membership Processing

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Membership Processing

Class Membership rules are created in OIPA using math variable-like syntax within Class Rules and Class Rule Variables. The Membership rules define the characteristics and requirements of each class. Class Membership rules are used to evaluate employee/sponsor clients to determine if they can be included for membership into a particular class. Establishing membership in classes within a Class Group can be used for any grouping purpose and application of common values, such as for plan eligibility, billing or reporting. See [Classes for Class Rules and Class Rule Variables](#).

How it Works

The membership process was implemented to invoke Membership rules that evaluate whether a provided data set for a client record and the associated data satisfy the requirements to allow class membership. The evaluation process for determining membership is executed during activity processing at the client and/or policy level. The transaction pulls client, address and Group Customer relationship data to use during the evaluation process.

This process typically occurs via a client level transaction that is processed during the data intake process. The transaction determines if Membership rules exist for the client and returns an error if class membership cannot be determined. Client level transactions associated with the data Intake process determine plan eligibility, class membership for eligibility and Population files.



At the **Client** Level, the evaluation occurs on the client the transaction is processed on. At the **Policy** Level, Class Membership is evaluated for the client designated as the Primary Member role on the policy.

In addition, multiple Policy level transactions can execute Class Membership processing. At the Policy level, the evaluation occurs on the client assigned to the Primary Member role.

Transaction and Activity Level Membership Rule Processing

Both Client and Policy level transactions make use of the Rule variables and Class Rules during the execution of the Class Membership process. Class Rule variables can be defined at the Global, Group Customer and Class Group levels. They are used in the XML then created, stored, and executed at the Class level for each specific class. The Class Rule variables house the specific demographic and employment type data for defining the Class Membership rule being executed.

For Activity Processing at both the Client and Policy levels, the system retrieves the Group Customer data for Policy and Client level activities through the relationship association. The Primary Relationship will indicate **Employment**. Execution of Membership rules automatically assigns any employees evaluated against the Class Group that do not resolve to any of the defined classes to a default (Orphan) class.



An Orphan class is set up as a default class when a new Class Group is set up in OIPA.

Membership Activity Processing

A transaction is configured in the Rules Palette at the Client or Policy level that will evaluate membership to a class.

Note: The Client must have a Relationship to the Group Customer as an Employee to imply Membership to become a Member of a Class.

Employee Relationship with Group Customer

The Data Intake File containing a Member Record to Add will imply that membership needs to be assessed and a Membership related activity will be spawned for processing at the Client or Policy Level.

The system code pulls in the Class Rule Variables and Class Rules, not the configuration. The configuration within the Transaction will be comprised of the Membership Element and its components for the Class Groups, Write Membership, Effective From/To.

We have developed Membership syntax within the configuration specific to writing the Membership records to the database. For more information, please consult the **XML Configuration Guide > Transactions > Membership Element**.

The Activity is added to the Client (in this example) and processed. The Membership results are displayed within the Class Membership tab.



Activity Results > Class Membership

The system views the Class Rule Variables in order to facilitate that funneling through to get to the last possible terminal leaf node or orphan Class to where the Member will be matched.



Activity Results > Class Rule Variables

The Class Rules facilitate the filtering to the next Class or Orphan Class using syntax that is formed using the predefined Class Rule Variables to form tests and conditions. The tests and conditions evaluate whether or not to move on to the next Class or to stop because a suitable Class was found to which the Client will become a Member.



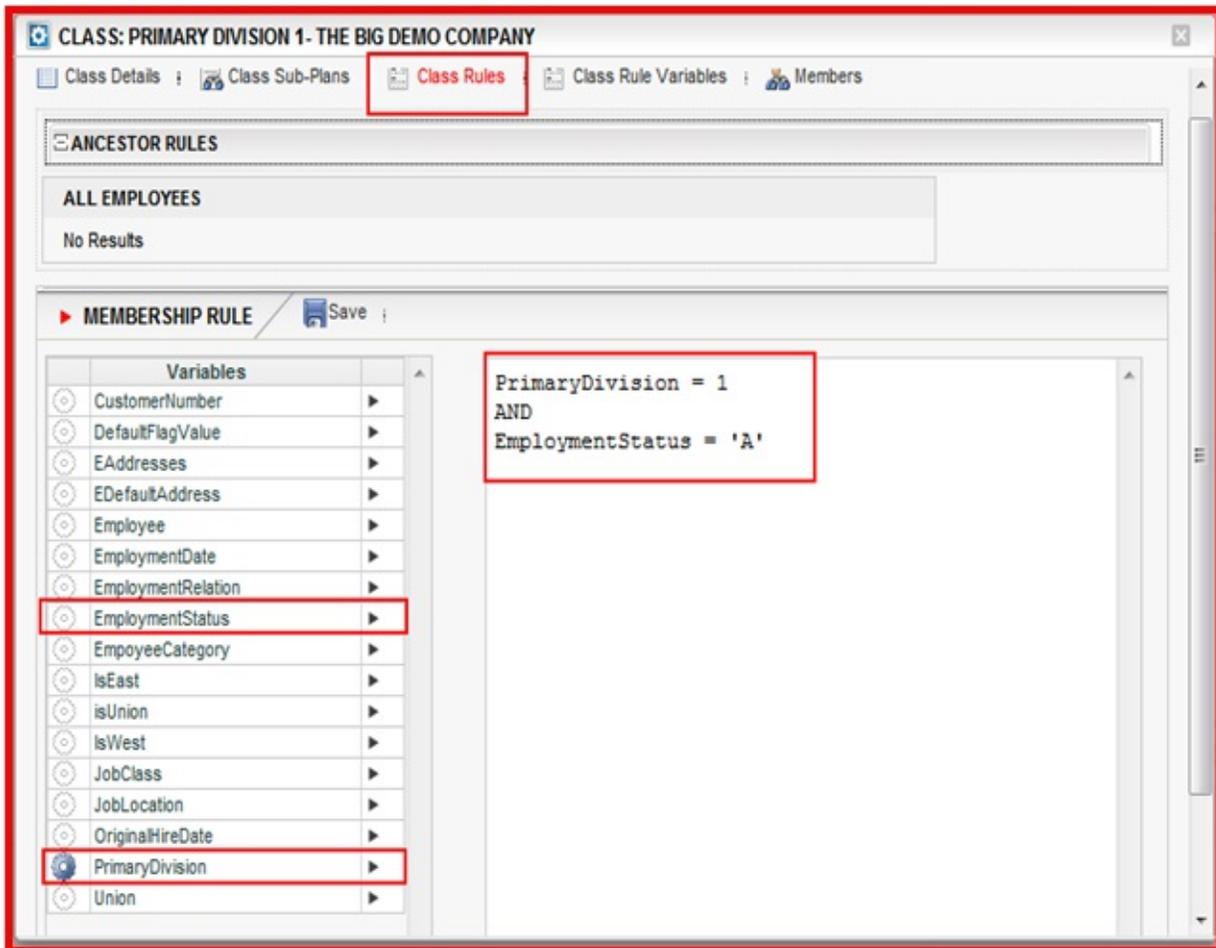
Activity Results > Class Rules

The Class tree structure can be accessed under Class Groups and the user can select the applicable Class to view Members.



Class Tree Structure

The Class Rules contain a condition or test that if True, places the Client into Membership with the specific Class. If the condition is false, the system goes to look at the next Parent Class and Child, and if still the condition ends in a False response, the Client will go into Membership within the Orphan Class.



Class Rules

The Members tab allows the user to view the Clients who are now Members of the Class.



Members Tab > Class > Part Time Union Employees

Navigation > Membership Process > Class Membership Activity Results

A **Class Membership** tab will be housed on the Activity Results screen of the Activities determining Class Membership. The Class Membership tab will contain a table listing all available Class Names and Class Group Names.

A Class Membership tab will be housed on the Activity Results screen of any Activities that include determination of Class Membership. The Class Membership tab will contain a table listing all available Class Names and Class Group Names.

If the Class Membership syntax was not used within the Transaction, a message will display indicating *"No Membership Processing Performed"*.

Membership records will be committed to the database (optional) unless the process is to provide a quote. We allow the override of writing the records to the database. The default action is to write the records, but the quote capabilities in the application provides an instance / circumstances where the need is not necessary to write a record to the database. Membership records will exist for the "leaf" records only - the leaf Class Membership only.



Activity Results > Class Membership

You are here: Configuration > Access External Database > Access Client Information from an External Database

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Access Client Information from an External Database

OIPA may be integrated with external systems to retrieve client information that can be displayed on the OIPA Role screen. External client information can also be attached to a policy. Information that is pulled in from an external database cannot be edited in OIPA.

Steps to Configure Access to an External Database

1. Make sure that [External Client](#) in Plan Pages security is enabled (checked) in the Rules Palette.
2. From **Administration | Code Names**, add codes for new external roles to the AsCodeRole code set in the AsCode table. If using the OIPA client search option, then add codes for external client types to the AsCodeClientType code set.
3. If needed, insert authorizations for the ExternalClient page to be able to see the ExternalClientDetail screen that is used to view the details of the external client.
4. Create a company level override of the [ExternalClientDetailsScreen](#) rule to configure the external client role codes and fields that are associated with each defined external role. The information in the rule is used to validate a set of keys identifying a client in the external system. The keys are returned when a new external role is created. The fields that must be configured include the following:
 - Key fields that identify a client in an external system. They are retrieved from an external system when a new external role is created, and stored in the AsExternalClient/AsExternalClientField tables in the OIPA database. Later, they are used as keys in requests to an external system whenever OIPA needs to obtain external client data. The key fields are indicated by a “No” value in the ExternalSource tag, which has its KEY attribute set to “Yes”. When a new external role is created, OIPA validates returned key values for a selected external client against this rule to make sure the keys are not empty.
 - OIPA-specific fields that are configured and exist in the OIPA application only. They are stored in the OIPA database. Values for these types of fields can be entered and modified on the External Client Detail screen. The OIPA-specific external client fields are indicated by a “No” value in the ExternalSource tag, which has its KEY attribute set to “No” (default value) in the ExternalClientDetailsScreen rule.
 - External fields that are always returned from an external system and never stored in the OIPA database. They are indicated by a “Yes” value in the ExternalSource tag in the ExternalClientDetailsScreen rule.

Fields of this type can be displayed on the External Client Detail screen but cannot be modified.

5. Configure a plan level override of the PolicyScreen rule or Segment rule (for segment roles). External client roles must be added to the configuration. The class that implements the IExternalClientRowRetriever interface must be called to retrieve a row of external client data from an external system.
6. If the Client Screen Search menu is needed in OIPA to search for external clients, then the ClientSearchScreen rule needs to include configuration for the client search screen for the defined external client types. The name of a class that implements the IExternalClientSearch interface should be specified using the ExternalClientSearchRetriever tag.

Implement Java Classes

After configuration is complete, integration with an external client system requires implementation of several OIPA interfaces by a Java developer. The external classes that implement the interfaces should be deployed in a shared library similar to any other extension classes.

Refer to the OIPA Extensibility document on [OTN](#) in the 10.1.2.0 library for a complete explanation of the process for implementing java classes to support external client retrieval.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > General Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transaction General Pane

Update required and optional transaction processing features from this pane. There are two required sections: Transaction and Effective Date, which are automatically added. The following sections are optional and can also be added.

- Values Block
- Withholding
- Valuation
- Suspense
- Membership
- Multi-suspense
- Transitions
- ValueFinancialEntry

The optional sections can be configured via the wizard when a [new transaction](#) is created or edited via **Edit Transaction General** from the right-click menu.

Expand or Collapse Sections on General Pane

Sections of the General Pane can be expanded or collapsed by clicking the box on the left side of the section. The box contains a plus sign  when the section is collapsed. Click the plus sign to expand the section. The box contains a minus sign  when the section is expanded. Click the minus sign to collapse the section.

Transaction Section

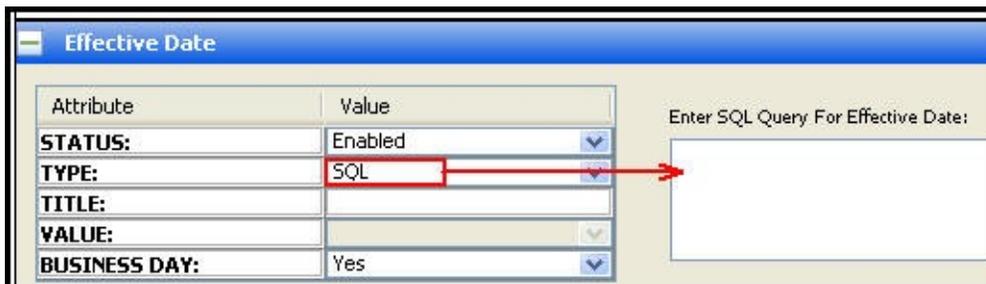
There are four fields that can be updated in the Transaction section on the General Pane.

- Type: The transaction type can be updated from this drop down box.
- Status: The transaction status can be updated from this drop down box.
- Processing order: Update the processing order by entering a new value in the processing order field. The Time Line button will open a window, which will display a list of all transactions and their processing order.
- NAV Correction: This should be set to Yes for all financial transactions.

Effective Date Section

All of the effective date attribute values can be updated from the Effective Date section on the General Pane.

- **Status:** Select Enabled or Disabled. If blank, no status attribute will appear in the XML source.
- **Type:** Select SQL, System or blank.
 - If SQL, then a box will appear to the right of the Properties information with a space for the user to enter a SQL query.
 - If System, then Value field should be disabled. Effective date should default to system date.
 - If blank, use the Value attribute field to enter the value for the effective date.
- **Title:** Enter a title for the Effective Date.
- **Value:** Only used when Type=blank. This field allows user to select a specific value from the Market Maker calendar for the activity effective date.
- **Business Day:** Select Yes or No. If Yes, the system checks to see if the current system date was a business day and if not, bumps it to the next business day for the Market Maker's calendar. If No, the system leaves the date unchanged. If blank, the BUSINESSDATE attribute will not appear in the XML.



Attribute	Value
STATUS:	Enabled
TYPE:	SQL
TITLE:	
VALUE:	
BUSINESS DAY:	Yes

Enter SQL Query For Effective Date:

Effective Date Section with SQL Selected

Optional Sections

The optional sections will have a green plus sign next to the name if the section contains information. Click the plus sign to open the section and update information. If the plus signs are greyed out, then that indicates that the sections are empty. Please see the **XML Configuration Guide** for a list of all elements, attributes and values needed for configuration.

- Values Block: Creates a section on the Activity Detail screen that displays the current policy fund values. The section is display only.
- Withholding: Allows for the configuration of the Withholding functionality.
- Valuation: Calls the valuation engine to run valuation processing.
 - **Effective Date NUV Must Exist:** If this box is checked, the Unit Values (NUVs) as of the activity's Valuation Date must be present in order for the activity to process at all. If there are no Unit Values, the system displays an error message stating **NUV's Missing (Effective Date)**.
 - **System Date NUV Must Exist:** If the box is checked, Unit Values as of the System Date must be present in order for the activity to process at all. If no Unit Values are present, the system displays an error message stating **NUV's Missing (System Date)**. If the box is unchecked for a variable product and NUV's do not exist for the System Date, but Unit Values are present for the Valuation Date, the activity will process in its entirety, except for the calculation of Gain/Loss. The Activity List screen will display a lightning bolt icon next to the activity with a status of Gain/Loss Pending until Unit Values are added for the System Date. After Unit Values have been entered for the System Date and either the lightening bolt is clicked again or Cycle is run, the Activity Status will change to Active and Gain/Loss will have been calculated.



For Fixed products, this should never be checked.

-
- **Suspense:** Displays and controls the suspense field inside the activity.
 - **MultiSuspense:** Defines the multi-suspense attribute values. The currencies of each suspense item must match the premium currency selected on the activity in OIPA.
 - **Membership:** Allows for the Membership section to be expanded and enabled. This section allows for the details relevant to the class membership calculation process to be entered on the Membership table.
 - **ClassGroupType:** this is used to determine the Class Group type of membership calculation. The drop down box will be populated with the CodeValue defined on the AsCode table for CodeName AsCodeClassGroupType.
 - **DisplayMembershipClass:** this is used to specify the Class Membership types to display. The default value for this field will be blank. When blank is selected the DISPLAYMEMBERSHIPCLASS element is not added.
 - **WriteMembership:** this indicates if membership records will be written to the database. Users can enter Yes, No or have the option to enter field or math variable names equal to Yes or No. When a value is entered a **WriteMembership** element will be added to the transaction XML. If blank, then no element will be added.
 - **EffectiveFromDate:** define the Effective From date of the membership calculation. Users can enter the string literal, **SystemDate**. Users also have the option to enter field or math variable names equal to a valid date. When a value is entered, an EffectiveFromDate element will be added to the transaction XML. If blank, no element will be added.
 - **EffectiveToDate:** define the Effective To date of the membership calculation. Users can enter the string literal, **SystemDate**. Users also have the option to enter field or math variable names equal to a valid date. When a value is entered, an EffectiveFromDate element will be added to the transaction XML. If blank, no element will be added.
 -
 - **Transitions:** Provides the ability to automatically update the effective

date of an activity as a way to transfer Gain/Loss to the client. If this checkbox is selected, two fields are enabled: Method and AdvanceToSystemDate. Two tags are also added to the XML Source with default values: <Transitions> and <Queue>.

- **Method:** Valuation is the only option for this drop down box. This is a required attribute that tells what validation is driving the Queue determination.
- **AdvanceToSystemDate:** This is an auto-complete text field that accepts a literal value of **Yes** or **No**. This auto-complete text box will load all field names and math variable names that are text datatype. This field will also resolve copybooks contained within the transaction. When not entering literal Yes or No values the user must select a field name or math variable name that holds a valid value of Yes or No. Entry in this field is optional.

Steps to Add or Remove Sections in the General Pane

1. Check-out the transaction's XML file.
2. Right-click on the transaction's XML file.
3. Select **Edit Transaction General**.
4. Check or uncheck XML section checkboxes.
5. Save and check-in the transaction.

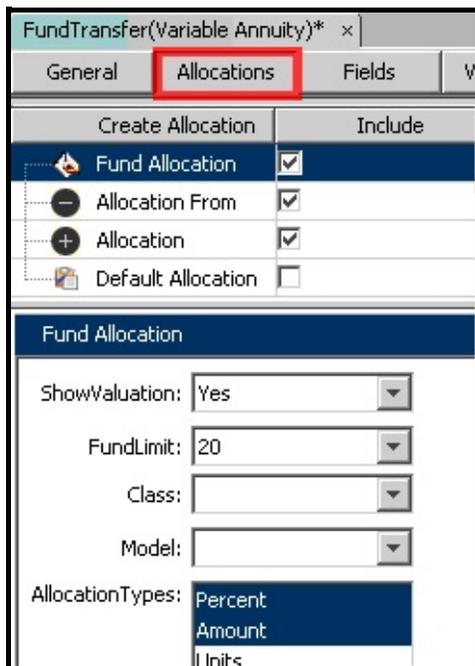
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Allocations Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Allocations Pane

This pane is used for the configuration of allocations. The allocation structure and default values for entries are configured here. When allocations are configured, the Activity Details screen in OIPA will have an Allocation link. If an activity in OIPA uses allocations, then a record will be written to AsAllocation when the activity is saved.



The screenshot shows a window titled "FundTransfer(Variable Annuity)*" with a tabbed interface. The "Allocations" tab is selected and highlighted with a red box. Below the tabs is a table with two columns: "Create Allocation" and "Include". The table contains four rows: "Fund Allocation" (checked), "Allocation From" (checked), "Allocation" (checked), and "Default Allocation" (unchecked). Below the table is a section titled "Fund Allocation" with several fields: "ShowValuation" (Yes), "FundLimit" (20), "Class" (empty), "Model" (empty), and "AllocationTypes" (Percent, Amount, Units). The "AllocationTypes" dropdown menu is open, showing "Percent", "Amount", and "Units" options.

Create Allocation	Include
Fund Allocation	<input checked="" type="checkbox"/>
Allocation From	<input checked="" type="checkbox"/>
Allocation	<input checked="" type="checkbox"/>
Default Allocation	<input type="checkbox"/>

Fund Allocation

ShowValuation: Yes

FundLimit: 20

Class:

Model:

AllocationTypes: Percent
Amount
Units

Allocations Pane of a Transaction

About Allocations

Allocations are the amount or percent of money that is applied to selected investment options for a policy. An example of an investment option could be mutual funds or money market accounts, which are called funds in the system. Policy holders or other requirements can direct how money is allocated to funds. Money is applied to allocation(s) by either assigning a percentage of the total amount or an exact value amount. Allocations can be individually selected funds or part of a model.

POLICY CLIENT DISBURSEMENT SUSPENSE TABLES

POLICY: Add A Withdrawal (01/29/2031) -

Details | Withholding | Allocations

ALLOCATIONS

Fund Name	Cash Value
Balanced (Band125)	141,400.00 USD

ACTIVITY

Number Of Funds: 2 Allocation Method: Percent

Equal Percent

Fund	Percent
Balanced	50.00
Mid Cap	50.00

Total: 100.00

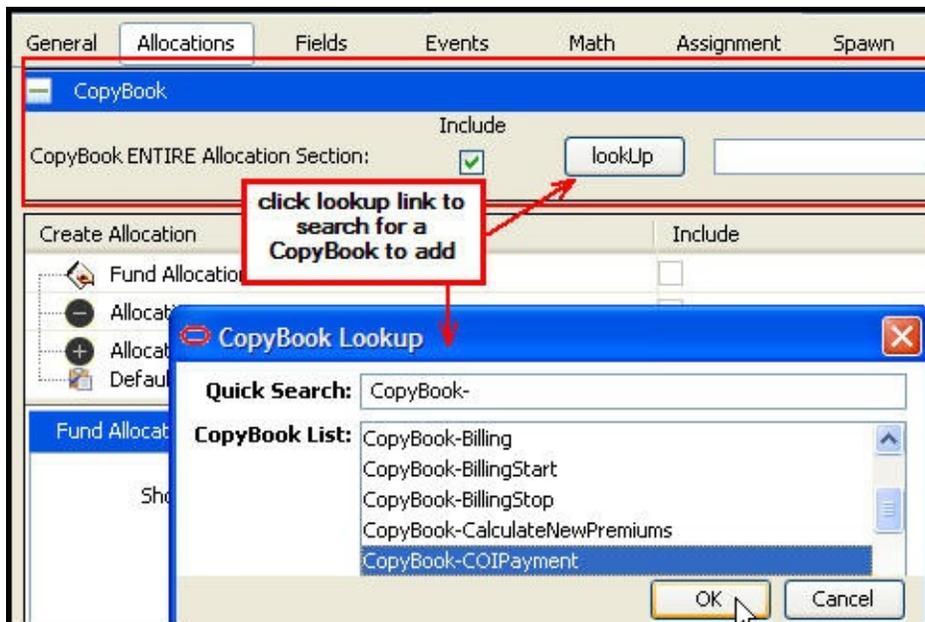
Ok Cancel

Allocations for an Activity in OIPA

When setting-up allocations there are two ways to configure: insert an entire Allocation section from a CopyBook or configure the [Allocation section through the Allocation pane](#).

Insert Entire Allocation Section from CopyBook

1. Navigate to the transaction and click the Allocation pane.
2. Expand the CopyBook section at the top of the pane.
3. Click the Include box.
4. Click **lookUp** and select the CopyBook from the CopyBook Lookup window.
5. Click **OK**. The CopyBook allocation information will be added to this pane.



Allocation from a CopyBook

Steps to Configure Allocations

1. Navigate to a transaction or create a new transaction.
2. **Check-out** the transaction's XML file.
3. Open the **Allocations** Pane.
4. Check the **Fund Allocation** checkbox. This is **required** and displays the Allocation section on the Activity Detail screen.
5. Select the Fund Allocation options needed from the Fund Allocation section that is displayed on the bottom portion of the screen. Data does not need to be entered for every field.
6. Check any additional allocation options from the top of the screen. These are all optional. A description of each option is listed below.
 - **AllocationFrom:** This option should be used when configuring for allocations that move money out of funds.
 - **Allocation:** This option should be used when configuring for allocations that move money into funds.
 - **DefaultAllocation:** This option defines a default allocation that will be displayed on the allocation section of the Activity Details screen and whether or not it can be modified. Only use this option if **FundAllocation** has been configured.
7. Configure each additional allocation that was checked.
8. **Check-In** when finished.



Hold down the CTRL key to select multiple items in a combo box.

Create Allocation		Include
Fund Allocation		<input checked="" type="checkbox"/>
Allocation From		<input checked="" type="checkbox"/>
Allocation	Select allocation	<input checked="" type="checkbox"/>
Default Allocation		<input type="checkbox"/>

Fund Allocation	
ShowValuation:	Yes
FundLimit:	20
PolicyField:	
Class:	
Model:	AIF
AllocationTypes:	<ul style="list-style-type: none"> Amount Percent Units
Assignment:	
AllowMixedMethods:	
Repeat Fund:	No
Exclude Fund Status:	<ul style="list-style-type: none"> Active Closed Closed to Additional Money
ExcludeType:	DCA+ Fund

Allocation Fields for Configuration

You are here: Configuration > Configuration Panes > Fields Pane > Fields Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Fields Pane

The Fields pane is used to create fields that will display in the OIPA application. Drag and drop fields from the [DataDictionary](#) or use the **SearchPalette** window to add new fields. The DataDictionary can also be used to search for existing fields that may be used.

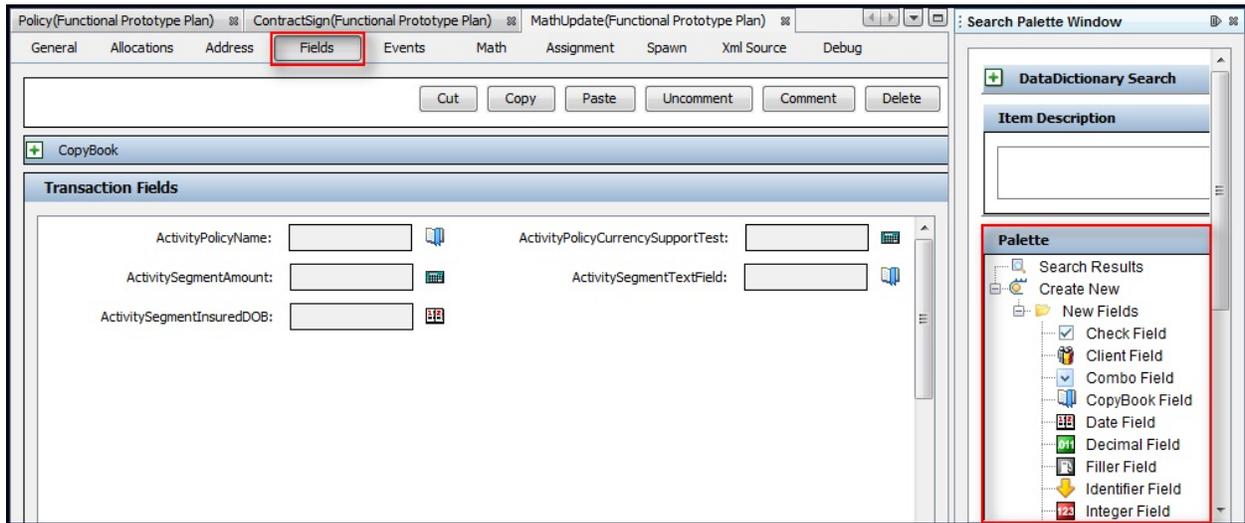
If a new field needs to be created, use the **Palette**, as it contains all the available fields for screen configuration. The Palette is included in the SearchPalette window. The Palette will automatically open when a transaction is checked out.

Field Placement

The Fields pane has a left and right column where fields can be placed. Drag a field to the left side of the pane to occupy the left column, or drag the field to the right side of the pane to occupy the right column. Fields can be expanded across both columns when the Expanded property is added to the field through the [Field Properties](#) window. [Filler fields](#) can also be added to provide a blank space to the layout. Fields will display in OIPA as they appear on the Fields pane.

Commenting Fields

Fields can be commented out using the **Comment** button at the top of the Fields pane. Once a field has been commented out, it can be uncommented using the **Uncomment** button.



Fields Pane with Palette Window Open

Steps to Add a New Field to a Transaction

1. Check out a transaction.
2. Click the **Fields** pane in the Configuration Area. The SearchPalette window should open on the side of the screen. If it does not, click **Window** on the Main Menu and select **Open Search Palette Window**. This window holds all the available field types.
3. Click one of the field types and drag it into the Configuration Area. The new field will appear with a generic field name, such as Field 1.
4. Click on the name of the new field, which will highlight it in blue. This will open the FieldProperties Window so the field can be configured. If the FieldProperties Window does not open, click **Window** on the Main Menu and select **Open FieldProperties Window**.
5. Click inside a field property in the FieldProperties Window and type the field information.



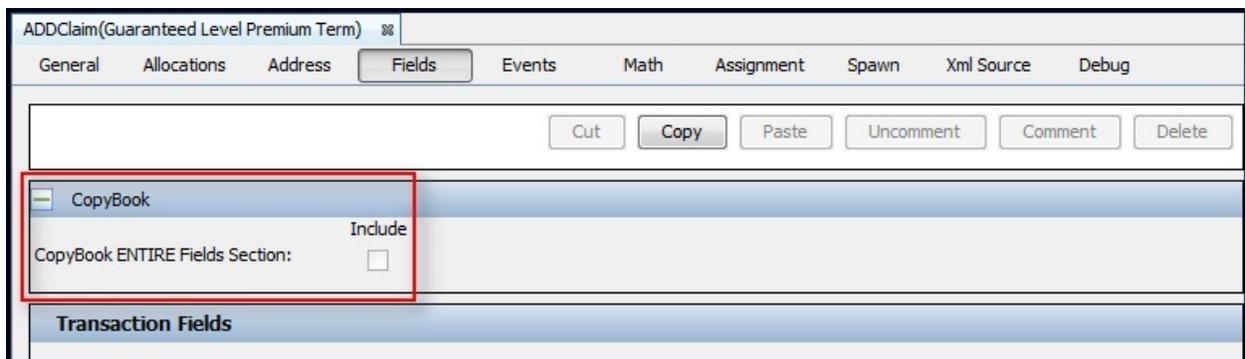
Any field with the [Required](#) property set to **Yes**, will appear in OIPA with an asterisk. This lets the user know the field must contain a value. Make sure to add any error message text to the translation table when configuring required fields.

6. Check in the transaction when finished.

Fields can also be configured using CopyBooks instead of the Fields pane. Expand the CopyBook section on the Fields pane and then follow the steps listed below.

Insert Entire Field Section from CopyBook

1. Navigate to the transaction and click the Fields pane.
2. Expand the CopyBook section at the top of the pane.
3. Click the Include box.
4. Click **lookUp** and select the CopyBook from the CopyBook Lookup window.
5. Click **OK**. The CopyBook field information will be added to this pane.



CopyBook section on Fields Pane

Cut, Copy and Paste Fields

The Fields pane supports the ability to cut, copy and paste fields. Fields can be pasted into either the same business rule from which they were cut/copied, or into the Fields pane of a different rule.

To cut fields:

1. Check out the rule containing the field(s) to be cut.
2. Select the field(s) to cut. Use the **Ctrl** key to select multiple fields at once.
3. Click the **Cut** button or press **Ctrl-X** on the keyboard. The selected field(s) will be removed and copied to the system's clipboard. The field immediately preceding those just cut will be highlighted.

To copy fields:

1. Select the field(s) to copy. Use the **Ctrl** key to select multiple fields at once. The rule does not need to be checked out to copy fields.
2. Click the **Copy** button or press **Ctrl-C** on the keyboard. The selected field(s) will be copied to the system's clipboard.

To paste fields:

1. Check out the rule to which the field(s) will be pasted.
2. Click the **Paste** button or press **Ctrl-V** on the keyboard. Fields can only be pasted into other Fields panes—an error message will be displayed if the user attempts to paste fields into a different configuration section. Fields will be highlighted after being pasted.



Actions within the Fields pane can be undone by using the keyboard shortcut **Ctrl-Z**.

Delete Fields

If a field needs to be deleted, select the field and click the **Delete** button at the top of the Fields pane. This will remove the field from the Fields pane and also remove the XML from the XML Source pane. Multiple fields can be deleted by using the Ctrl key to make multiple selections before clicking Remove. Alternately, the XML can be removed directly from the XML Source pane.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Fields Pane > Field Properties

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

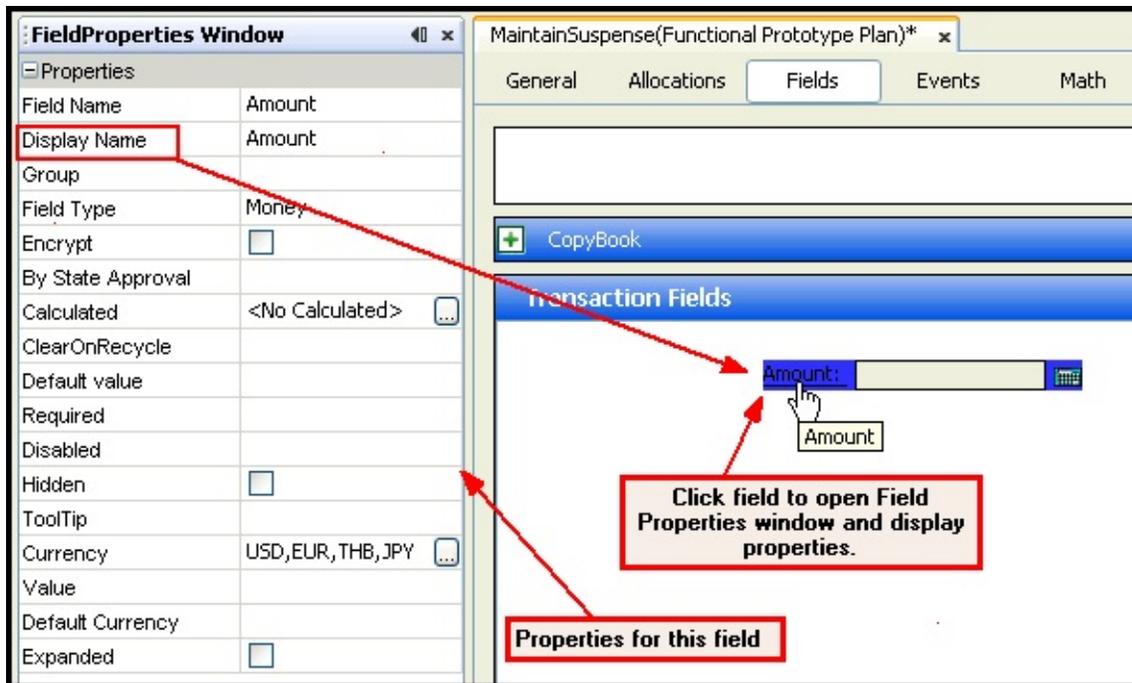
Field Properties Window

When a transaction or applicable business rule is checked out and the Fields pane is open in the [Configuration Area](#), then a user can click on a field to open the FieldsProperties window. The properties of the field will display and can be updated from this window. The name of the field shown on the Fields pane corresponds to the Display Name property in the FieldProperties Window.

There are several ways to update a property. Examples of each method are provided below.

- Some properties have a blank row. Click in the row next to the property name and type the property information.
- Some properties have a checkbox. Click the checkbox to apply the property to the field.
- Some properties have a  button next to the property. Click the button to open a window and add the field information. After the information is added, select **Close**.

Regardless of the method, the **Save** button should be clicked after updating the property information. When the transaction is checked in, the new property information will be saved to the database.



Field Properties Window for a Transaction Field

Most Commonly Used Field Properties

Field	Definition
Field Name	The name of the field.
Display Name	The name that displays on the screen. This is the label and it can have spaces in the label name.
Group	
Field Type	The type of field displayed on the screen. Since Version 8, configuration Field Type is now required. Field Type replaces DataType in pre-version 8 configuration. Please see the XML Configuration Guide topic in this help system for a list of all elements, attributes and values needed for configuration. Field Type also defines a set of values and the allowable operations on those values.
Encrypt	Allows the value entered to be encrypted. This is used for credit card information or other highly sensitive information.
By State Approval	Field will hide or display based on State Approval (used for rider indicator or program fields). Only applies to segment fields.
Query	Can be used to populate a combo box using a SQL statement or Fixed values defined through Field Options .
Calculated	Can be used to default the field to a value derived through an SQL statement.
ClearOnRecycle	A value of Yes means that the field is cleared when an activity is recycled by the user. If the activity is reversed as the result of a system undo-redo cycle, the field is not cleared. A value of No means that the field is not cleared when the activity is manually recycled.
Default Value	The initial value displayed when the user brings up the screen or transaction.
Disabled	Displays the field and the fields information, but the user cannot enter the data.

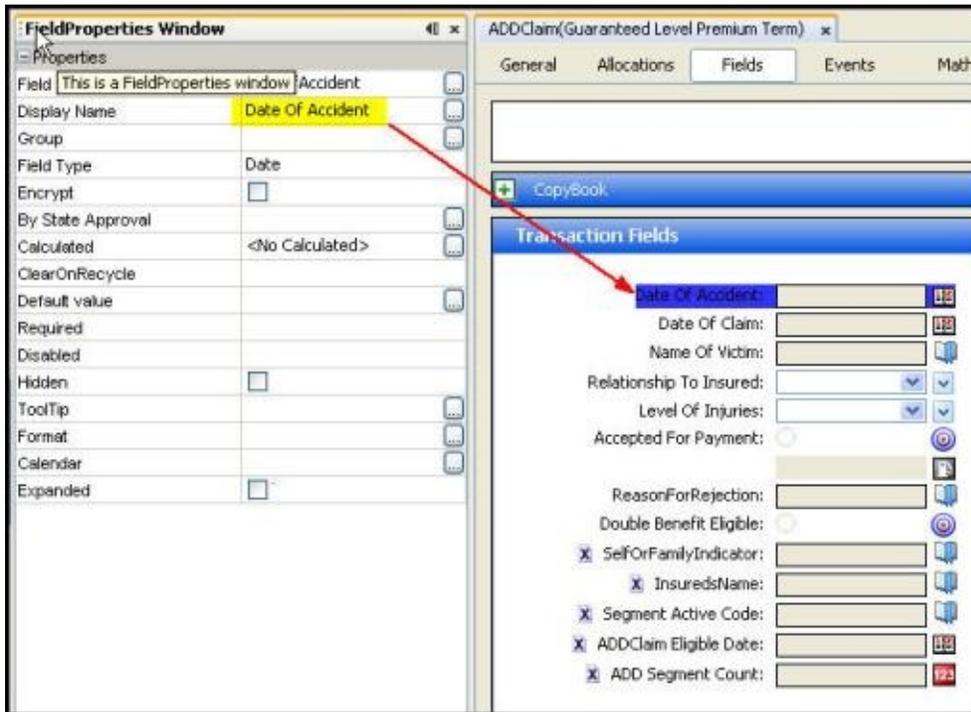
Hidden	Whether or not the field is displayed on the screen.
Required	<p>Yes or No value. This tells OIPA if the field is required. An asterisk will display next to all required fields in both the Rules Palette and OIPA. Note: If this field property is left blank, then no XML is added.</p> <p>The following field types support the required element: fixed fields, dynamic fields and multifields.</p> <p>The following datatypes support required element: combo, text, date, decimal, integer, money, percent, client, radio.</p> <p>If a field is required, then the hidden and disabled field properties will not be available.</p>
Length	Determines how many characters are permitted for a field value.
Tool Tip	Allows a tool tip to be entered that will be visible at mouse-over of the field.
Currency	Allows a currency to be selected, which will define how money is displayed.
Value	Allows a value to be entered.
Default Currency	Allows a default currency to be selected.
Parts	Provides information to create a next identifier. Used commonly with Identifier Field Types.
Expanded	If this box is checked, then the field will extend across both columns and display as one row in OIPA. If this box is not checked, then the field will display in only one column.
Mask	Allows a mask to be put on a field.
Final	If this box is checked, all lower-level overrides will be unable to edit the field's configuration. This option will only be available if the use of Product structure is enabled, and only applies to fields utilizing Product-Plan data inheritance .

You are here: Configuration > Configuration Panes > Fields Pane > Field Properties_Group

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Field Properties Window

Clicking on a field in the [Configuration Area](#) opens the FieldsProperties window. Make changes to a field's configuration from this window. To update a property, click the select  button next to the property. A window will open with the available options for that property. Make the selections and then select **Close**.



Field Properties Window for a Transaction Field

Most Commonly Used Field Properties

Field	Definition
Field Name	The name of the field.
Display Name	The name that displays on the screen. This is the label and it can have spaces in the label name.
Group	
Field Type	The type of field displayed on the screen. Since Version 8, configuration Field Type is now required. Field Type replaces DataType in pre-version 8 configuration. Please see the XML Configuration Guide topic in this help system for a list of all elements, attributes and values needed for configuration. Field Type also defines a set of values and the allowable operations on those values. DECIMAL, TEXT, DATE, INTEGER.
Encrypt	Allows the value entered to be encrypted. This is used for credit card information or other highly sensitive information.
By State	Field will hide or display based on State Approval (used for rider indicator or

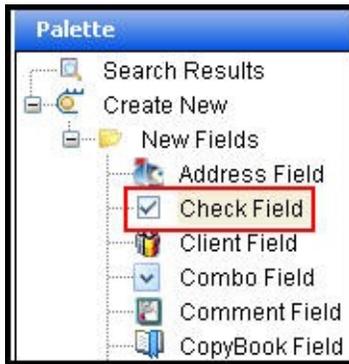
Approval	program fields). Only applies to segment fields.
Query	
Calculated	Can be used to default the field to a value derived through a SQL statement.
ClearOnRecycle	Used to enable clearing the field when the user manually recycles the transaction.
Default Value	The initial value displayed when the user brings up the screen or transaction.
Disabled	Displays the field and the fields information, but the user cannot enter the data.
Hidden	Whether or not the field is displayed on the screen.
Required	The field must be filled in by the user when screen or transaction information is submitted.
Element	Allows a mask to be put on a field.
Length	Determines how many characters are permitted for a field value.
Tool Tip	Allows a tool tip to be entered that will be visible at mouse-over of the field.
Parts	Provides information to create a next identifier. Used commonly with Identifier Field Types.
Currency	Allows a currency to be selected, which will define how money is displayed.
Value	Allows a value to be entered.
Default Currency	
Expanded	

You are here: Configuration > Configuration Panes > Fields Pane > Check

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Check Field

The Check field is used to place a checkbox on the screen. The user will have the option to check the box or leave it unchecked.



Check Field in Palette

Define the Check Field

Drag and drop the Check field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information such as field name, display name, etc., by clicking in the field and typing the information.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Fields Pane > Client

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Client Field

The Client field allows a client to be added to a policy through activity processing. In OIPA, this field does not support text entry, but instead provides a client icon, which when clicked by a user opens the ClientSearchScreen in a popup window. An existing client in the AsClient database can be selected using the **Find Client** link or a new client can be added using the **New Client** link at the top of the window. The New Client link opens the Client screen in a popup window.

When a new client is added using the client field, the new client information is added to the AsClient database table. The client name will also display in the Client field. The display characteristics of the client name are handled in the code and parts definition in the configuration and are formatted by the system based on client type. The field will hold the ClientGUID and the system will do the necessary formatting based on that selection so that the option text is displayed properly for the selected client.

Multiple clients can be added using the Client field. If multiple clients are added, only the last ClientGUID will be stored and the name displayed in the Client field on the transaction.

The [AddRoles](#) business rule may be attached to the transaction to allow a role to be assigned to a client when the activity processes. Creating the client and adding the role can occur all within one transaction but as two separate events within the same transaction. The adding of the client will take place prior to the transaction processing. The adding of the role will occur once the transaction is processed. If the activity is reversed, any assigned roles will be removed, but the client information will remain in the database.

All ClientScreen validations for required fields and duplicate clients are available.



Client Field in Palette



The client datatype will not be available on the Client screen. This is due to multiple instances of the Client screen generating when clicking the people icon from the Client screen and its use being redundant on that screen. OIPA will simply ignore a field with the Client datatype if used on the Client Screen.

Define the Client Field

Drag and drop the Client field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information such as field name, display name, etc., by clicking in the field and typing the information.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Fields Pane > Combo

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Combo Fields

The Combo field is used to provide the user with a specific set of options that can be selected from a drop down box. These options can be configured directly in the XML Source pane using the <Query> and <Options> tags or they can be configured using Rules Palette visual configuration.



Refer to the **XML Configuration Guide** in the Help menu for additional information on Field configuration if configuring in the XML Source pane.



Combo Field in Palette

Define the Combo Field Using Visual Configuration

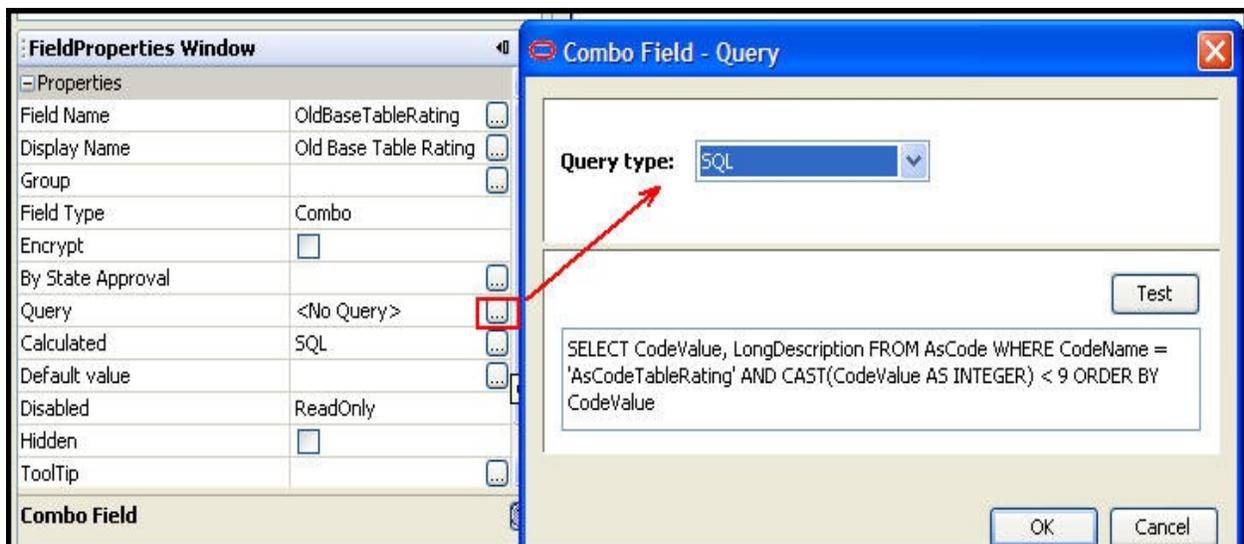
Drag and drop the Combo field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information such as field name, display name, etc., by clicking in the field and typing the information.

Determine the Contents of the Combo Box

The Query and Calculated properties can be used to populate the combo box. Click the button to the right of the Query property to open the Combo Field Query window. Select the Query type.

- **SQL:** enter the statement in the box provided. The **Test** button can be used to make sure the statement runs without errors. Click **OK** when the query is complete.
- **Fixed:** click **Add** in the window provided and then enter an [option value](#) and [option text](#) for each option that should appear in the Combo box. The option text is what will display to the user in the combo box.

Click the button to the right of the Calculated property to open the Combo Field Calculated window. Select the type of calculation, the method and then enter any necessary statements in the box provided. The Test button is an additional tool that can be used to run through the statement to see if any errors occur. Select **OK** when the calculated information is complete.



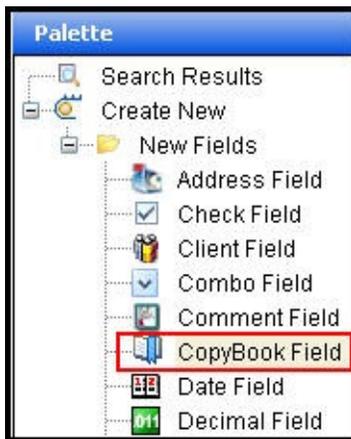
Query Property and Combo Field Query Window

You are here: Configuration > Configuration Panes > Fields Pane > Copybook

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

CopyBook Field

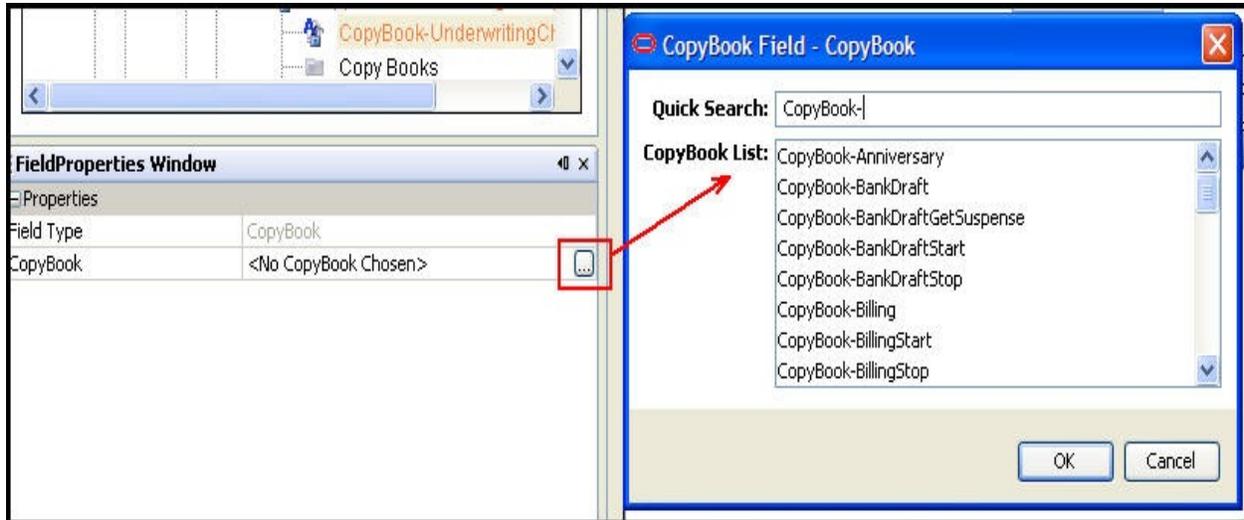
The Copybook field is used to attach a copybook, which is a way to cut and paste configuration directly into your transaction. Make sure you use a copybook that is compatible with the type of transaction you are configuring. For example, don't use the copybook of a Client transaction on a Policy screen.



CopyBook Field in Palette

Define the CopyBook Field

Drag and drop the CopyBook field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Click the button to the right of the Copybook property. A window will open with a list of CopyBook options. Select one from the list and click **OK**.



CopyBook Property with Window to Select CopyBook

You are here: Configuration > Configuration Panes > Fields Pane > Date

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Date Fields

The Date field is used to provide the user with a way to capture date information.



Date Field in Palette Window

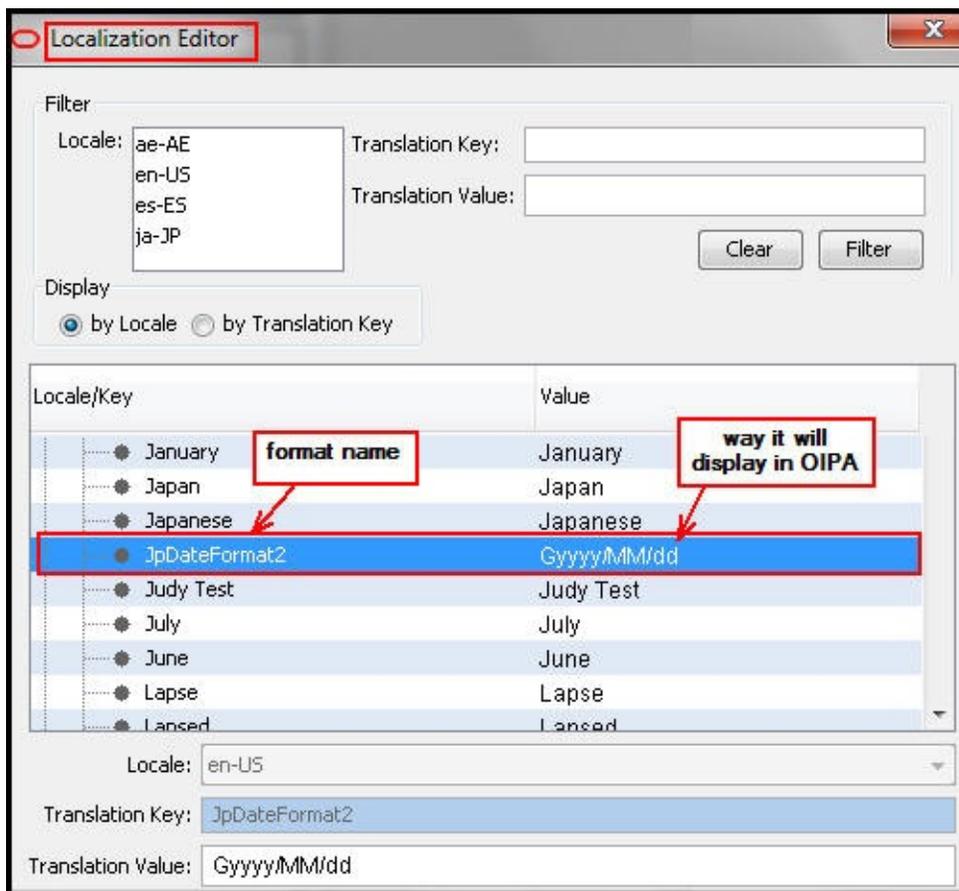
Define the Date Field

Drag and drop the Date field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information such as field name, display name, etc., by clicking in the field and typing the information.

Determine Date and Calendar Format

The Format property in the FieldProperties window allows a user to specify a date display format. The format must first exist in the AsTranslation table so it can be referenced in the Format field. If the format does not already exist, a record can be added to the AsTranslation table through the [Localization Editor](#).

The Calendar property allows a user to specify a calendar to use for the date field. The Imperial and Gregorian calendars are the only calendars which are supported by the system so the translation value on the AsTranslation table must be either Japanese or Gregorian stored in English only.



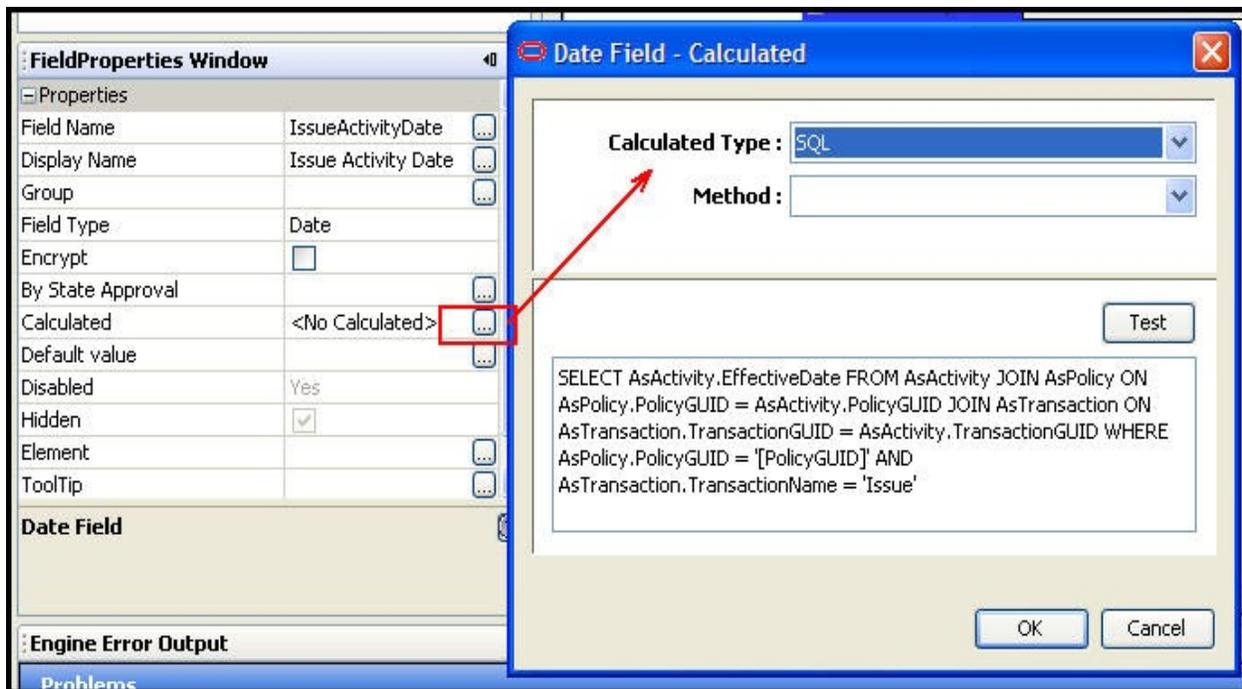
Date Format in Localization Editor

Steps to Define a Date and Calendar Format

1. Drag the Date field onto the Field pane of a transaction.
2. Click on the new field in the Configuration Area. The Field Properties window will open.
3. Type a name for the field and a display name.
4. Type any other properties needed.
5. Click the ellipses button next to the Format field and type the name of the translation key that describes the date format. This must exist in the AsTranslation table. A sample format translation key is shown in the image above. JpDateFormat2 would be the value entered for the Format property.
6. Click the ellipses button next to the Calendar field and type the name of the translation key for the calendar. This must exist in the AsTranslation table. For example, the translation key with a translation value of Gregorian would be entered to call the Gregorian calendar and the translation key with a value of Japanese would be entered to call the Imperial calendar.
7. Check-in the translation to save the information.

Determine How To Calculate Dates

Specific dates can be calculated using the **Calculated** property in the FieldProperties Window. Click the button to the right of that property and the Date Field-Calculated window will open. Select the type of calculation, the method and then enter any necessary statements in the box provided. The Test button is an additional tool that can be used to run through a SQL statement to see if any errors occur. Select **OK** when the calculated information is complete.



Calculated Option in Field Properties and Date Field-Calculated Window

You are here: Configuration > Configuration Panes > Fields Pane > Decimal

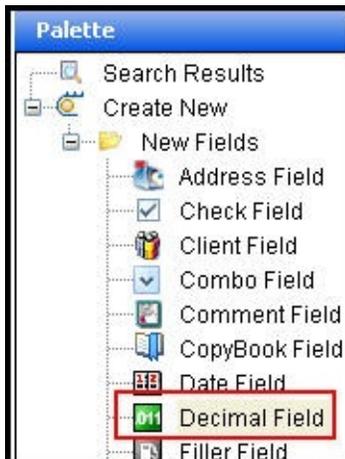
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Decimal Fields

A Decimal field is used to capture values that require decimal places. When a field is identified as a decimal field, then OIPA knows that the value entered into the field should be handled during processing as a decimal value. A decimal field should not be used to hold currency values.



When dealing with integers larger than 2,147,483,648 in math configuration, DATATYPE="DECIMAL" must be used.



Decimal Field in Palette

Define the Decimal Field

Drag and drop the Decimal field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information such as field name, display name, etc., by clicking in the field and typing the information.

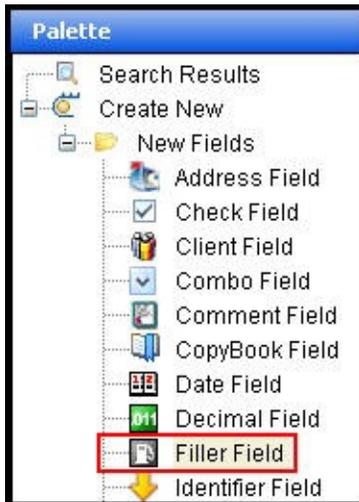
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Fields Pane > Filler

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Filler Fields

A Filler field is used to add empty space between fields. This allows the configurator to control spacing between fields.



Filler Field in Palette

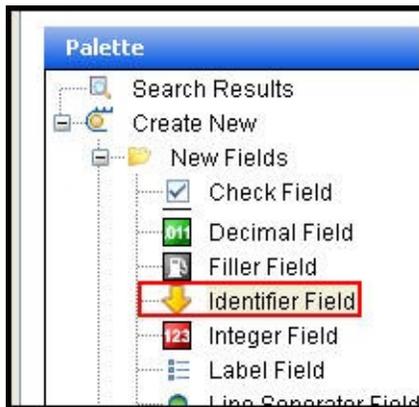
You are here: Configuration > Configuration Panes > Fields Pane > Identifier

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Identifier Fields

The Identifier field data type is typically used in the following situations:

- to generate identification numbers using alphabetic and/or numeric parts. The numeric parts may be incremented in order to provide a unique ID constructed with the same elements. An example would be the generation of a claim number for a death claim process that is unique among other claim numbers.
- to generate a value built from variable data values. An example would be the generation of a classification of payments by its source (Payment-Cash, Payment-EFT, Payment-Rollover).



Identifier Option in Palette Window

Configuration Considerations

Keep in mind the following points when configuring Identifier field types.

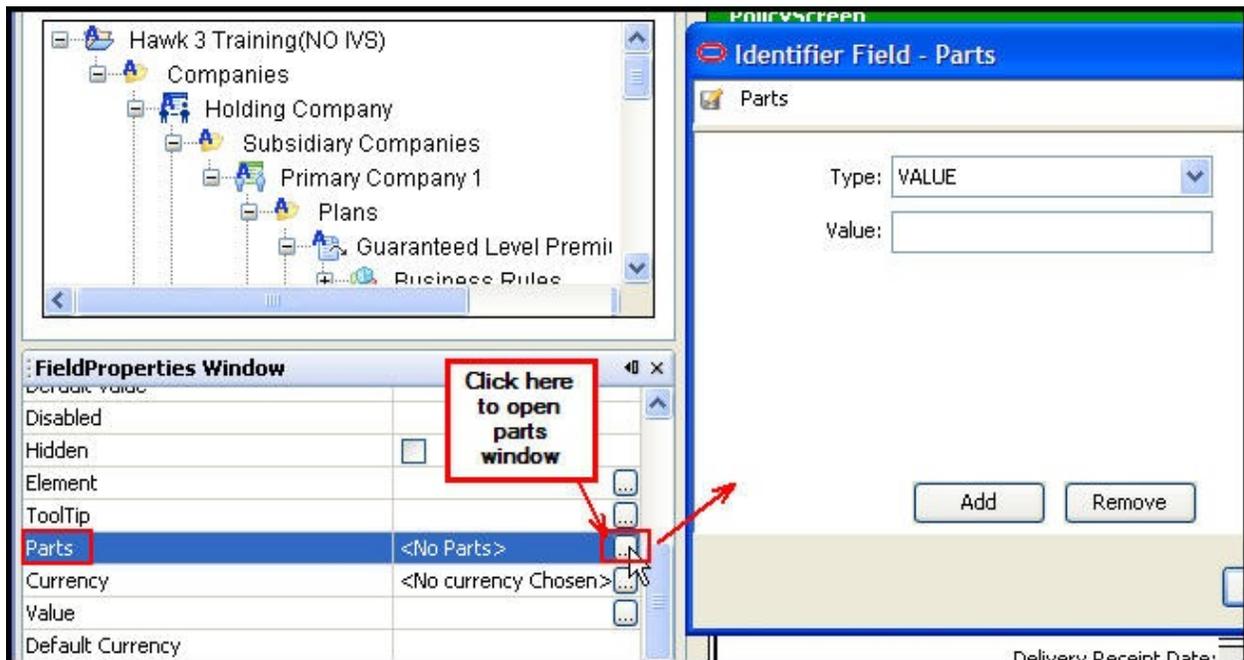
1. The Identifier field data type is applicable in any component that may contain the common entry Fields definition (screens, activity fields section). It does **NOT** apply where data must pass from one component to another as in spawns and updating business rules (CopyTo...) or in Math and ScreenMath.
2. An Identifier field cannot be used in ScreenMath. The field's value is generated when the component is saved to the data store and is therefore not always available for ScreenMath.
3. Once an Identifier field has been generated using its unique set of attributes, it is treated as a text value everywhere else.
4. An identifier field may be passed to another activity through a spawn or updated to a component in an APE rule. The receiving field is defined as a text field and the data is passed as if it were text.
5. An identifier field may be source data for a math variable. It is treated as a text field with the math variable defined as a text data type.
6. The total character length of the identifier field is limited to the same size as a field of data type TEXT.
7. Each type is added to the Parts node in the Rules Palette Identifier Field - Parts window. Each type translates to a <Part> element in the XML.
8. When generating confirmation numbers used by the [Confirmation screen](#), the Identifier field should be used to generate the original, or parent, activity confirmation number. The [Identifier math variable](#) should NOT be used. Use the Identifier math variable to generate confirmation numbers for spawned activities.
9. Identifier Field configuration supports the use of the <[ClearOnRecycle](#)> sub-element. When <ClearOnRecycle> is set to **Yes**, the value of the Identifier Field will be on-load cleared if the activity is manually recycled. If the activity is re-done due to system-generated undo/redo, then the original Identifier Field value is retained.
10. Numbers generated by the Identifier field element cannot be used as

search criteria.

Define Identifier Field and Parts

Drag and drop the Identifier field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information such as field name, display name, etc., by clicking in the field and typing the information.

Scroll down to the Parts line and click the box to the right of that line. This will open the Parts window, where the individual components of the unique identifier can be created. An explanation of each type is given in the section below.



Field Properties Window for Identifier Field

Identifier Parts

The following list explains the various Identifier Parts options. Each type is listed, along with the configuration options that accompany the type. Each type translates to a <Part> element in the XML.

- **When Type=Value is selected:**

- The value of the <Part> element is the literal text that is concatenated to the identifier field.

- **When Type=SystemDate is selected:**

- Format: Portions of the system date or all of the system date may be concatenated to the identifier field. The required attribute FORMAT defines the portions that will make up the resulting value to concatenate. The following is a list of the attribute's values and what they accomplish.

- YY and YYYY results in 2 and 4 digit years, respectively.
 - MM results in a 2 digit month.
 - DD results in a 2 digit day.
 - YYMM and YYYYMM results in a 2 digit year, 2 digit month value or a 4 digit year, 2 digit month value, respectively. Both create a value in year-month order.
 - YYYYMMDD results is a full date value with a 4 digit year in year-month-day order.
 - YYMMDD results is a full date value with a 2 digit year in year-month-day order.

- **When Type=Sequence is selected:**

- A sequence number is generated from a sequence set identified by the sequence name referenced in the <Part> element's value. Once a sequence number becomes part of the Identifier field's value, it cannot be duplicated by any subsequent Identifier field using the same sequence set. A sequence number becomes part of the Identifier field's value as the screen component or Activity Detail

screen is persisted in the data store. Concurrent accesses to the same sequence will yield different sequence values.

A required FORMAT attribute specifies the padding characters and length of the output that is concatenated to the identifier field. The padding character may be any visible character or mix of characters that define the padding character(s). The length of the output is defined by the number of characters in the FORMAT attribute's value. For example, if FORMAT="000", the complete length of the <Part> value is 3. If the sequence number is 1, then the output is 001. If the sequence number is 100, then the output is 100.

An optional SEQUENCEDATE attribute filters the sequence records for a specific sequence set. It may be used to restart the sequence numbering as of a specific date. The value of the attribute indicates the data source for the date. One data source for the attribute's value is a field name. The field must be defined prior to the identifier field and must be a date data type. Other data sources for the attribute's value may be the system date and an activity's effective date. The latter is applicable to transaction configuration only.

The following image shows data that resides in the sequence data store for a specific sequence set.

	Sequence date	Sequence number
1	1/1/2008	501
2	1/1/2009	25
3	1/1/2010	1

Sequence Data Store for a Sequence Set

- For any date value provided to SEQUENCEDATE that falls between 1/1/2008 and 12/31/2008, inclusively, record #1 is used to supply sequence numbers. The next number is 501.
- For any date value provided to SEQUENCEDATE that falls between 1/1/2009 to 12/31/2009, inclusively, record #2 is used to supply sequence numbers. The next number is 25.
- For any date value provided to SEQUENCEDATE that is equal to or greater than 1/1/2010, record #3 is used. The next number

is 1.

- Any positive integer value may be used as a starting value. Maximum value for a sequence number is about 2.1 billion.

■ Field

- Parse: select **Left, Right, Mid** and enter a Parse Value. These indicators are used to tell the system what part of the value to concatenate to the identifier's field value. A sample of each option is given below:
 - Left & Parse Value: specifies the number of characters extracted from the start of the field's value. Ex: If the Field value is **12345**, then LEFT with Parse Value=2 would mean that **12** is extracted and concatenated to the identifier's value.
 - Right & Parse Value: specifies the number of characters extracted from the end of the field's value. Ex: If the Field value is **12345**, then RIGHT with Parse Value=2 would mean that **45** is concatenated to the identifier's value.
 - Mid & Parse Value: specifies the starting and ending location in the field's value from which to extract the value. Ex: If the Field value is **12345**, then MID with Parse Value 2 and 5 would mean that **2345** is concatenated to the identifier's value.
- Format: indicates the padding character and the length of the resulting value. Any character or mixture of characters define the padding character(s). The length of the output is defined by the number of characters in the FORMAT attribute's value. Padding will always occur to the left of the parsed value to create a result that is the length of FORMAT attribute. For example, FieldA contains the value of **Mon**. FieldB is an identifier data type field with a Part type of Field defined with a FORMAT=**ddddd**. The resulting value is **ddMon**, which is then concatenated to the identifier field's value. The padding character is **d** and the resulting value's length is 5 because **d** was repeated 5 times in the FORMAT attribute's value.

- Value: The value of the Part element of TYPE="FIELD" is the name of the field whose value may be parsed and padded then concatenated to the identifier field's value.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Fields Pane > Integer

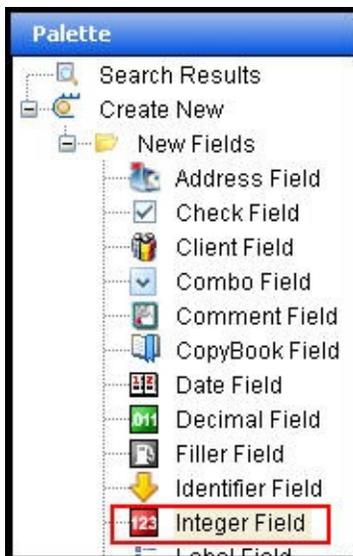
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Integer Field

The Integer field is used to store non-decimal numeric values. Do not use this type of field for currency values.



When dealing with integers larger than 2,147,483,648 in math configuration, DATATYPE="DECIMAL" must be used.



Integer Field in Palette

Define the Integer Field

Drag and drop the Integer field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information such as field name, display name, etc., by clicking in the field and typing the information.

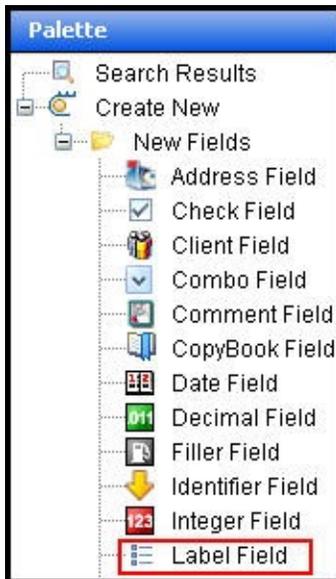
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Fields Pane > Label

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Label Field

The Label field is used to create a low-level heading for a section of fields (as opposed to the higher-level heading created by the Title field), and can be used in both transaction and business rule configuration.



Label Field in Palette

Define the Label Field

Drag and drop the Label field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information, such as field name, display name, etc., by clicking in each field and entering the information. The text entered in the Display Name field will be displayed in OIPA as regular (i.e. not bold) text, left-aligned with respect to the column in which it is placed.

You are here: Configuration > Configuration Panes > Fields Pane > Line Separator

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Line Separator Field

The Line Separator field is used to divide section of the screen with a horizontal line. No property information is necessary for this field.

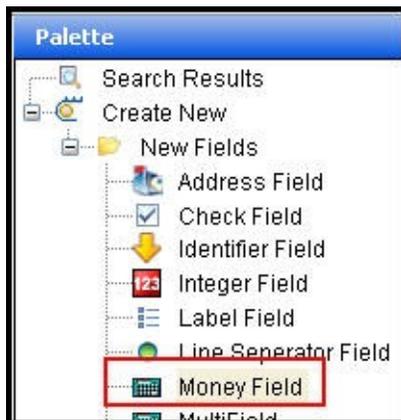
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Fields Pane > Money

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Money Fields

The Money field data type is used to indicate that the field contains a monetary value.



Money Field in Palette Window

Define the Money Field

Drag and drop the Money field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information such as field name, display name, etc., by clicking in the field and typing the information.

Use Currency Properties

There are two currency properties that can be set to define how the value displays in the money field. The **Currency** property defines how the value entered into the money field will display. The use of separators and decimals is governed by the [locale](#) of the user. The currency defines the currency precision, meaning the number of places to display after the decimal. The AsCurrency table contains the precision information for each currency and can be edited through the Currency editor in the Admin Explorer.

Click the ellipses button on the Currency row. Click the currency for the field and then click **OK**. If multiple currencies should be available to the user in OIPA, then hold CTRL and click the currencies from the currency list. They will be available to the user in a drop down box in OIPA.

Note: If the Currency is not selected for a Money field, then the DefaultCurrency for the Plan will be used.

The **Default Currency** property defines the currency that will display in the drop down box if multiple currencies are available for the money field. It also defines the default format for the value in the money field.



Currency Options in Field Properties Window

You are here: Configuration > Configuration Panes > Fields Pane > Multifields

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Multifields Field

The Multifields field is used to turn on multifold functionality and specify a particular multifold rule. When a multifold is added to the Fields pane, it will be added as the first field on the pane in a section that spans across both rows.



Multifield Field in Palette

Define the Multifield Field

Drag and drop the Multifield field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Click the button to the right of the Multifield property and select a multifield rule. Only one multifield field is allowed for each transaction.

Refer to the [Multifields prototype](#) for specific configuration information on working with multifields.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Fields Pane > Multifield Events

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Multifield Events

Transaction multifield events are very similar to general use events with some notable relationship and functionality variances. This section explains those variances.

General Structure

Multifield events retain most of the events building blocks, and are configured between <MultiField> tags. Each multifield is identified by a <Name> element, which is how the multifield is identified in event processing. Each multifield may have its own set of Events, however it will not have its own Math section. The multifield can access Global ScreenMath from the transaction.

The general structure for Events is:

Events>Event>Math>ActionSet>and/or QuerySet.

The general structure for Actions is: Actions>ActionSet and/or QuerySet>Condition>Action>Else.

Multifield Syntax

There are three types of events that are supported in multifield processing: OnLoad, OnChange and OnSubmit.

Event syntax remains the same, with the addition of two new <Action> attributes: MULTIFIELD and INDEX.

The MULTIFIELD attribute is used to identify the multifield name where the action is to occur. If only the MultiField is referenced then the event occurs on the entire section.

The INDEX attribute is used to indicate the position/row of the field the action is to occur on (starts at 0). If INDEX is used then only the field for that index will be impacted.

The FIELD attribute is an existing attribute that identifies the field name where the action is to occur. When used with MULTIFIELD this indicates which field within the multifield will be impacted. If INDEX is not indicated all instances of that field within the multifield will be updated.



Please see the **XML Configuration Guide**, which can be accessed from the Help menu, for a list of all elements, attributes and values needed for configuration. View **Transaction Rules | Transaction Elements | ActionEvents**.

Additional Multifield Event Considerations

- Events configured within the “calling” business rule will be able to designate whether the entire MultiField section, or any of the multifields (names) within the section are visible or editable (enabled or disabled).
- ActionSets and QuerySets configured within the MultiField business rule will be able to access (use) fields that are configured within the same MultiField section (name).
- Events configured within the MultiField business rule will be able to affect properties and values of only those fields that are configured within the same MultiField section (name), and only fields with the same index.
- Each MultiField business rule’s multifield (name) may have its own set of events.
- Events configured within the MultiField business rule will NOT be able to affect fields configured within the “calling” transaction rule.
- Events configured within the MultiField business rule will NOT be able to designate whether other multifields (names) are visible or editable (enabled or disabled).

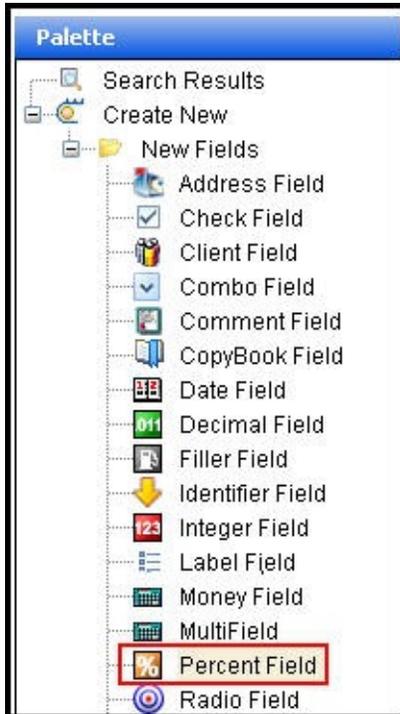
You are here: Configuration > Configuration Panes > Fields Pane > Percent

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Percent Field

This field is used to enter a percent.

Note: The value of 1 in the database will display on the screen as 100. Entering 100 on the screen will save to the database as 1.



Percent Field in Palette

Define the Percent Field

Drag and drop the Percent field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information such as field name, display name, etc., by clicking in the field and typing the information.

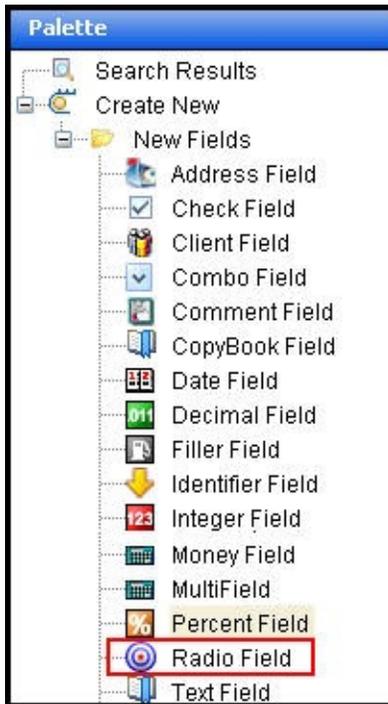
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Fields Pane > Radio

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Radio Field

The radio field is used to add a radio button and descriptive text.



Radio Field from Palette

Define the Radio Button

Drag and drop the Radio field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information such as field name, display name, etc., by clicking in the field and typing the information. You can have multiple buttons on a radio field but only one button can be selected at a time.

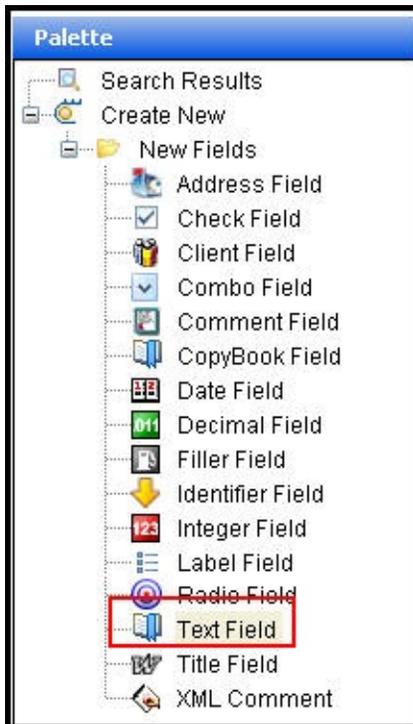
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Fields Pane > Text

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Text field

The Text field is used to capture text information. A [mask](#) can be applied to a text field to obscure the field value from view or to format the data when displayed in OIPA.



Text Field in Palette

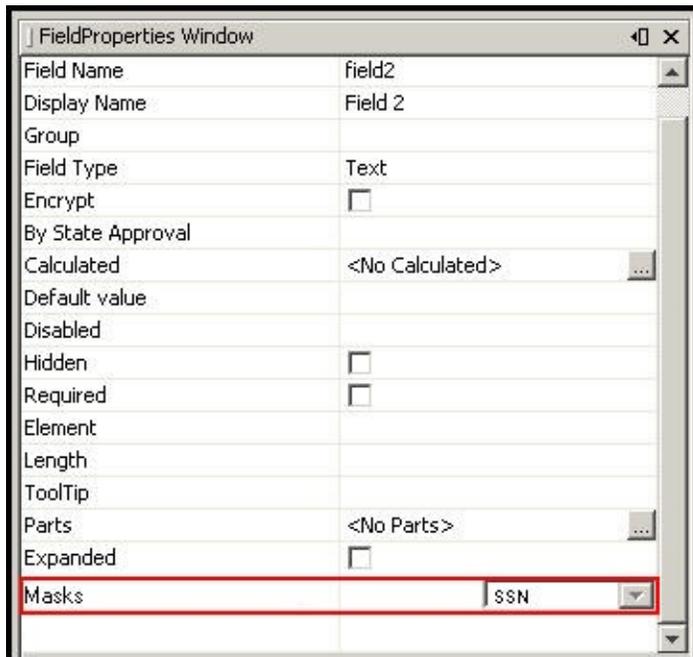
Define the Text Field

Drag and drop the Text field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information such as field name, display name, etc., by clicking in the field and typing the information.

Apply a Mask

A mask can be applied to a text field to hide the field value or format the data when a user views it in OIPA. In order to add a mask, open a transaction or an entry screen and click the Fields pane. Click on the text field to highlight it. This will open the FieldProperties Window. Scroll to the bottom of the window and located the Masks property. A drop down box will display to the right of the property. Select a mask option from the drop down box.

The mask options in the drop down box are pulled from the AsMaskDetail table. The [Mask Editor](#) is available in the Rules Palette to create and edit these mask options.



Masks Field Property

You are here: Configuration > Configuration Panes > Fields Pane > Text Area

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Text Area Field

The Text Area field allows a user to enter large blocks of text in a field. The data entered in the TextArea will display in OIPA screens and in Activity Results. A character limit can be specified using the <Length> element. There is no minimum or maximum validation for length.

There are some restrictions imposed on these fields. TextArea fields cannot be used as criteria fields in a search. They are also not supported within the <Table> element. The Inquiry screen requires special consideration as well. The field entry section of an Inquiry screen does not support the new data type. However, the new data type is supported in the results section when it displays fields as opposed to a data grid.

Multifield rules support the use of TextArea fields.



Text Area Field option in Palette

Define the TextArea Field

Drag and drop the TextArea field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information such as field name, display name, etc., by clicking in the field and typing the information. All field elements are supported, with the exception of masks. There is no restriction to the <Action> element or its attributes and events may be triggered from changes to a TextArea field.

Refer to the [Large Text Prototype](#) for a complete explanation of the configuration required to support the use of large amounts of text on screens in OIPA.



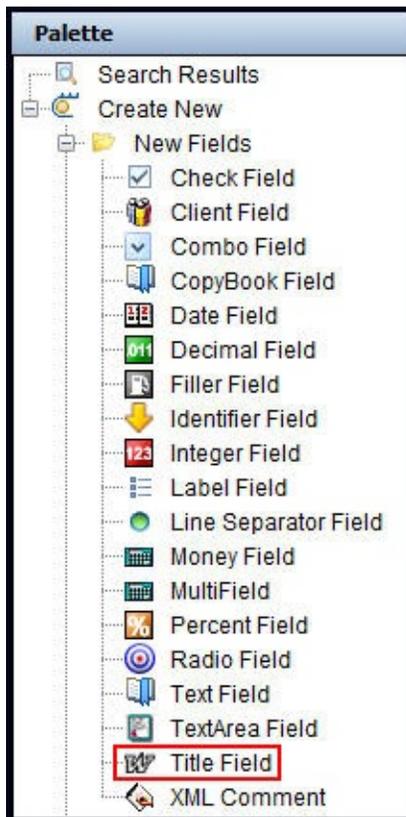
If [spawning](#) to a TextArea datatype in the To Spawn field, it is recommended that only TextArea and Big Text datatypes are used in the From Spawn field.

You are here: Configuration > Configuration Panes > Fields Pane > Title

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Title Field

The Title field is used to create a high-level heading for a section of fields (as opposed to the lower-level heading created by the Label field), and can be used in both transaction and business rule configuration. Title fields will display as centered text spanning both columns of fields on the screen.



Title Field in Palette

Define the Title Field

Drag and drop the Title field from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Enter the field information, such as field name, display name, etc., by clicking in each field and entering the information. The text entered in the Display Name field will be displayed as bold text centered across the entire screen.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Fields Pane > XML Comment

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

XML Comment

This allows you to add a comment from the Fields pane instead of scrolling through the XML in the XML Source pane.

Define the XML Comment Field

Drag and drop the XML Comment from the Palette window onto the Fields Pane. The Field Properties window will open, which provides the means for configuring the field. Click the button to the right of the XML Comment and enter the comment in the window that is provided.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Events Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Events Pane

Action Events provide the ability to perform actions on a field when an event occurs on the screen. Configuration is done by establishing user events and linking them to Action Sets via ID attributes. When the user selects something in OIPA an event invokes a set of Action Sets that change fields or display messages.

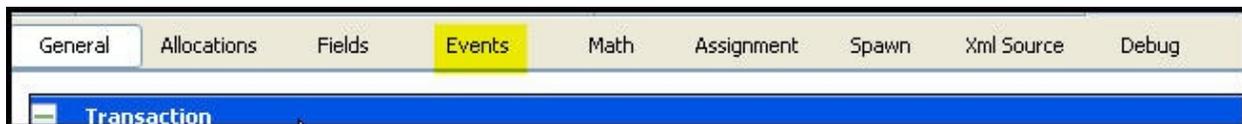
There are various types of Action Events that can be configured.

- create custom error and warning messages that appear to the end user after the system validates what was entered.
- change whether fields are enabled or disabled when an event occurs.
- change whether fields are hidden or displayed.
- change a field's data according to the value in a trigger field.

Configuration is performed through the Events pane located on the menu bar across the top of the [Configuration Area](#).

Please see the **XML Configuration Guide**, which can be accessed from the Help menu, for a list of all elements, attributes and values needed for configuration. View **Transaction Rules | Transaction Elements | ActionEvents**.

Events can also be configured in Multifields. Refer to the [Multifield Events](#) section of the help for information.



Events Pane

High Level Steps to Configure Action Events

1. [Configure the fields](#) that the actions can be applied to.
2. [Configure the event](#) that needs to occur to invoke the actions. This must be done in the XML Source pane as no visual configuration is available at this time. If events are configured within the <Multifield> tags, then some additional consideration is required. Refer to [Multifield Events](#) for additional information.
3. Configure the [ScreenMath](#) to define the math that is available for use when an event is invoked. Either global or local math can be used by fields.
4. [Identify the set of actions](#), ScreenMath, QuerySets and/or ActionSet, that are configured in the event.
 - If changing field data, configure a [QuerySet](#) that defines the data used to populate combo boxes according to a trigger field's value.
 - If creating errors, warnings or controlling a field, configure an [ActionSet](#) that defines the action to be performed on a field.

Configure an Event

The events section identifies the type of event that needs to occur to invoke a set of actions. It also identifies the field that is connected to the event.

The available event types that can invoke an action are:

- **ONLOAD:** runs the event on the initial load of the screen.
- **ONCHANGE:** runs the event when a trigger field is changed. OnChange will recognize a change to the specified field's value but not a change to its currency code.
- **ONSUBMIT:** processes the event when the final submission on the screen is selected.

Steps to Configure an Event

Events can be configured directly in the XML using the XML Source pane or they can be configured in the Events pane with the help of visual configuration tools. The Events pane also provides a way to see the individual parts of Events and Actions configuration without the additional transaction XML. Click an Action from the Event Navigator window to the left of the Configuration Area to add the opening tag to the configuration. If configuration already exists for the Action or Event, that configuration will display in the Event pane.

1. Check-out the transaction, segment or business rule's XML file.
2. Click the **Events** tab. The Event Navigator window will open.
3. Double-click the **Events** node. The opening tag for Events will display in the Configuration Area.
4. Configure the event. Identify the event TYPE and, if applicable, the FIELD that the event applies to.
5. Configure any combination of Math, ActionSets and/or QuerySets. Each action ID attribute must have a matching action.
6. **Check-in** the file to save the changes to the database.

The screenshot displays the 'Event Navigator Window' on the left and the 'XMLData' pane on the right. The 'Event Navigator' shows a tree view with 'Events' selected, indicated by a red box and the word 'Select'. The 'XMLData' pane shows the XML configuration for events, with several elements highlighted by red boxes and arrows:

- Type of Event**: Points to the 'TYPE' attribute in the XML, such as 'ONCHANGE'.
- Field Name if applicable**: Points to the 'FIELD' attribute in the XML, such as 'RelationshipToInsured'.
- Action IDs to call if Event occurs**: Points to the 'ID' attribute in the 'ActionSet' elements, such as 'OnChangeRelationshipToInsured'.

```
<Events>
  <Event TYPE="ONCHANGE" FIELD="RelationshipToInsured">
    <ActionSet ID="OnChangeRelationshipToInsured"/>
  </Event>
  <Event TYPE="ONCHANGE" FIELD="AcceptedForPayment">
    <ActionSet ID="OnChangeAcceptedForPayment"/>
  </Event>
  <Event TYPE="ONSUBMIT">
    <ActionSet ID="FinalValidation"/>
  </Event>
</Events>
```

Events XML

```
<Events>
  <Event TYPE="ONCHANGE" FIELD="EffectiveDate">
    <Math ID="Math100"/>
    <QuerySet ID="Query100" FIELD="MailCode"/>
    <ActionSet ID="Action101"/>
  </Event>
</Events>
```

Event TYPE, FIELD and IDs

ScreenMath

Configure the <ScreenMath> section in order to perform field math to support actions. In <ScreenMath> configure <Math> sub-elements that are each identified with an ID so that events may locate it for use. This is different from the <Math> section, because it provides the ability to perform calculations for the Fields section. ScreenMath is not available for use in the main transaction <Math> section.

The <ScreenMath> is identified through a configurable ID and can include two types of <Math>:

- **Global ScreenMath:** All fields can use math in this section. It is only loaded on the initial load of the screen and cannot use field values.
- **Local ScreenMath:** This is invoked when indicated by an action. Fields may be used; however, the configurator must be aware that the ScreenMath logic must be configured in such a way as to ensure that all fields being used take into consideration all of the calls to the use of the ScreenMath for that ID.

There can only be one GLOBAL ScreenMath section. The GLOBAL attribute with a value of **Yes** indicates to the system that the math is global. Without the GLOBAL attribute the system identifies the <Math> as local. The global ScreenMath is available in the local ScreenMath section. Local ScreenMath is not available in global ScreenMath. When naming ScreenMath do not name global and local math the same. If the names are the same, the local math will take precedence.



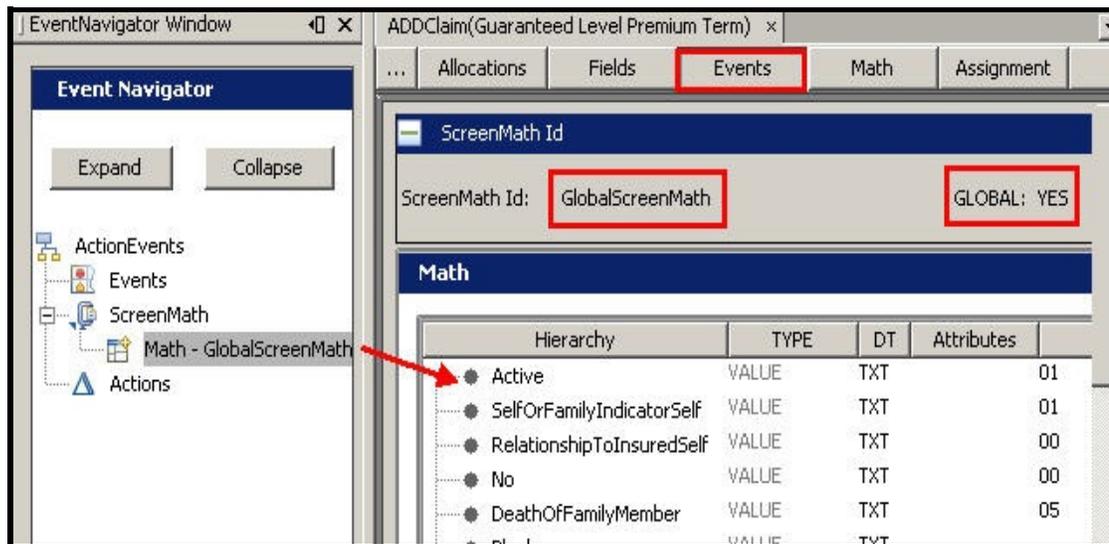
Refer to the [Screen Math Considerations](#) page for additional configuration guidelines.

```
<ScreenMath>
  <Math ID="GlobalScreenMath" GLOBAL="No">
    <MathVariables>
      <MathVariable VARIABLENAME="Currency1MV" TYPE="VALUE" DATATYPE="DECIMAL">10000.00</MathVariable>
      <MathVariable VARIABLENAME="Currency1MVUSD" TYPE="FUNCTION" DATATYPE="CURRENCY">ToCurrency(Currency1MV,'USD')</MathVariable>
      <MathVariable VARIABLENAME="Currency2MV" TYPE="VALUE" DATATYPE="DECIMAL">20000.00</MathVariable>
      <MathVariable VARIABLENAME="Currency2MVUSD" TYPE="FUNCTION" DATATYPE="CURRENCY">ToCurrency(Currency2MV,'USD')</MathVariable>
      <MathVariable VARIABLENAME="Currency3MV" TYPE="VALUE" DATATYPE="DECIMAL">30000.00</MathVariable>
      <MathVariable VARIABLENAME="Currency3MVTHB" TYPE="FUNCTION" DATATYPE="CURRENCY">ToCurrency(Currency3MV,'THB')</MathVariable>
      <MathVariable VARIABLENAME="Money1MV" TYPE="EXPRESSION" DATATYPE="CURRENCY">Money1</MathVariable>
      <MathVariable VARIABLENAME="Money1CurrencyCode" TYPE="FUNCTION" DATATYPE="TEXT">GetCurrencyCode(Money1)</MathVariable>
    </MathVariables>
  </Math>
</ScreenMath>
```

Screen Math XML

Steps to Configure Screen Math

1. Check-out the transaction, segment or business rule's XML file.
2. Click the **Events** tab. The Event Navigator window will open.
3. Right-click the **Screen Math** node and select **New Math Section**.
4. Enter the **Section ID**, which is the name of the ScreenMath.
5. Check the **Global** box if the math being configuring is global math.
6. Select **Finish**.
7. [Configure the math](#) section by dragging and dropping math variables from the Palette window.
8. Check-in the XML file to save the changes to the database.



Global Screen Math on Events Pane



View the ScreenMath XML from the XML Source pane. See image below.

```
Allocations Fields Events Math Assignment Spawn XML Source Debug
</Event>
</Events>
<ScreenMath>
  <Math ID="GlobalScreenMath" GLOBAL="Yes">
    <MathVariables>
      <MathVariable VARIABLENAME="NA" TYPE="VALUE" DATATYPE="TEXT">00</MathVariable>
      <MathVariable VARIABLENAME="NonQualified" TYPE="VALUE" DATATYPE="TEXT">01</MathVariable>
      <MathVariable VARIABLENAME="Replacement" TYPE="VALUE" DATATYPE="TEXT">98</MathVariable>
    </MathVariables>
  </Math>
  <Math ID="ValidationMath" GLOBAL="No">
    <MathVariables>
      <MathVariable VARIABLENAME="MinimumSubPremiumSMV" TYPE="PLANFIELD" DATATYPE="DECIMAL">MinimumSubP
    </MathVariables>
  </Math>
</ScreenMath>
```

Screen Math XML from XML Source Pane

Define Actions

There are two types of actions: (Use any combination of the options listed below.)

- [QuerySets](#) identifies a set of action configuration to be used for the event.
- [ActionSets](#) identifies a set of query configuration to be used for the event.

Actions and Currency Considerations

Actions can be configured to compare currency values and assign new ones to fields. Some common operations include comparing money to money fields using the less than expression, comparing currency math variables, assigning a currency code to a math variable or field and assigning a value from a money field to another with like currency codes.

Keep in mind the following points when using currency in Actions.

- `IsEmpty()` just checks for the presence of a numeric value, the currency code is ignored.
- Comparisons of named fields or math variables use the attached currency codes in addition to their numeric values; the associated currency code values must be the same or the system will throw a stack trace indicating the currency codes are incompatible.
- The Currency Code of a field can be found using the syntax of "FieldName:CurrencyCode".
- The Currency Code of a math variable can be found using the function "GetCurrencyCode ()".
- Action type **ASSIGN** will carry the Currency Code of the math variable or field when assigning a value provided the source currency is permitted for the target field. It cannot target changes to the currency code independent of the Money field.
- Action type **ASSIGN** should validate the source currency code against that of the target. If the currency code of the target does not allow the currency of the source, the system will throw a stack trace indicating the currency codes are incompatible. The Currency code of the target is defined in the <Currency> element of the field or in its absence, the plan or company default currency using standard best match logic.
- Action type **ASSIGN** will only accept Money fields or CURRENCY math variables as assignment values for Money fields.

QuerySets

QuerySets allow combo boxes to load with a SQL statement, as well as define a set of hard coded combo box options. <QuerySet> is a sub-element of <Actions> and each <QuerySet> is identified with an ID attribute that the <Event> section uses to locate the correct configuration.

When configuring the change of combo box data, make sure to use the FIELD attribute in the corresponding <Event> section for the trigger field. In the <QuerySet> sub-element use the FIELD attribute for the field that will have its data changed.

Conditional logic can also be written in this section.

- Compare single values by writing expressions. Please see the **V9 XML Configuration** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Common Elements | Operations Available for Expression Writing**.
- Use available [functions](#).
- Use Global and Local math (Must use MathID prefix).
- Compare dates.

```
<QuerySet ID="Query100">
  <Condition IF="Field != '01'">
    <Action ACTIONTYPE="SQLQUERY">SELECT CodeValue ...</Action>
    <Else>
      <Action ACTIONTYPE="OPTIONS">
        <Option>
          <OptionValue>01</OptionValue>
          <OptionText>Mail</OptionText>
        </Option>
        <Option>
          <OptionValue>02</OptionValue>
          <OptionText>Hold</OptionText>
        </Option>
      </Action>
    </Else>
  </Condition>
</QuerySet>
```

QuerySet XML

Steps to Configure Action with QuerySets

1. Check-out the transaction, segment or business rule's XML file.
2. Click the **Events** tab. The Event Navigator window will open.
3. Double-click the **Actions** node. The XML data will open on the Configuration Area.
4. Configure the action by defining the QuerySets needed for the event. A sample is shown above.
5. **Check-in** the file to save the changes to the database.

ActionSets

ActionSets allow for the configuration of errors and warnings, enabling fields, disabling fields, hiding fields, displaying fields and assigning a value to a field. `<ActionSet>` is a sub-element of the `<Actions>` and each `<ActionSet>` is identified with an ID attribute that the `<Event>` section uses to locate the correct configuration.

There is also the ability to write conditional logic in the section. Available logic includes:

- Compare single values by writing expressions.
- Use available [functions](#).
- Use global and local math (must use MathID prefix).
- Compare dates.

Available actions are:

- **Error**: Provides an error message to the user. The user will not be able to proceed until all errors are fixed.
- **Warning**: Provides a warning message to the user. The user will be able to proceed even with warnings on the screen.
- **Enable**: The user can key in a field value.
- **Disable**: The user cannot key in a field value and the fields background color changes.
- **ReadOnly**: The user cannot key in a field value but the fields background color does not change.
- **Show**: The user will see the field.
- **Hide**: The user will not see the field.
- **Assign**: Allows a value to be set based on the change of a different field. Assign does not have an option to set a value and then re-execute the OnLoad or OnChange of the field being set.

Within the Actions tags for `<Action ACTIONTYPE="ERROR">` or

<Action ACTIONTYPE="WARNING"> , a validation message can be added. Within this element, a screen math variable or activity field name may be substituted in the validation message to display an actual value from the activity. In the example below, a field or screen math variable is named "Amount". The validation verifies that the amount is greater than 100.00. In the Error message \$\$\$Amount\$\$\$ would be used in the configuration validation message to display the value in the Amount field or screen math variable. If using an activity field name or math variable in the Actions/Events validation, a prefix is not required. The validation would then read, "Payment Amount of 40000.00 must be less than 100.00." The configuration in the <Actions> sections would be similar to the example below:

```
<Action ACTIONTYPE="ERROR">Payment Amount of $$$Amount$$$ must be less than 100.00.</Action>
```

Steps to Configure Action with ActionSet

1. Check-out the transaction, segment or business rule's XML file.
2. Click the **Events** tab. The Event Navigator window will open.
3. Double-click the **Actions** node. The XML data will open on the Configuration Area.
4. Configure the action by defining the ActionSets needed for the event.
5. **Check-in** the file to save the changes to the database.



Additionally, Group Elements such as Class, SegmentName, Agreement, Client Relationship, Group Customer can all utilize Action/Events (in conjunction with MultiFields Rule) for validation/warnings/error messaging.

You are here: Configuration > Configuration Panes > Math Pane > Math Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Math Pane

The Math pane is used to perform the bulk of business processes. Calculations can be performed using math variables and the values calculated can be used by supporting rules.

Examples of when to use math:

- add, subtract, multiply or divide values
- pull in policy data to be used within functions
- pick the next business day for an activity
- determine whether or not to spawn a transaction

The Math pane contains the definitions and calculations that process math variables and functions. Literal values may also be applied to math variables. Set-up conditions to evaluate if something is true or false and create an array of values. Fields may be referenced in the Math pane of a transaction in order to perform calculations.



Math variables are populated in the order in which they are configured. If a math variable name is repeated in the configuration, the final value will display in the Math tab of the Activity Details screen.

Steps to Create a New Math Variable

(Please see the **XML Configuration Guide** for a list of all elements, attributes and values needed for configuration. View **Transaction Rules | Transaction Elements | Math Elements**.)

1. Open a transaction in the Configuration Area. Check-out the transaction so that changes can be made and saved.
2. Click the Math pane at the top of the window.
3. Make sure the SearchPalette window is open. It contains a list of math variables that can be dragged and dropped onto the screen. If this window is not open, click **Window** on the Main Menu and select **Open Search Palette Window**.
4. Drag the type of math variable needed from the Palette window to the Configuration Area.
 - **General:** Commonly used math variables, such as CURRENCY, EXPRESSION, VALUE, POLICYFIELD and STOREDPROCEDURE.
 - **Activity:** Math variables used with spawning multiple activities, in creating an array and populating the fields for the spawned activities.
 - **Array:** Math variables used to create and perform operations on an array.
 - **Commutation:** Math variables used to perform commutation functions.
 - **Conditional:** Math Variables that perform IIF or MathIF operations.
 - **Loop:** Math variables that perform math loop operations, including FOR, SEGMENT and OBJECT.
 - **Rate Lookup:** Math variables that retrieve a RateArray or Rate.
 - **Rule:** Math variables that call a CopyBook, FunctionCall or Function.
 - **Solve:** Math variables that create an IRR, SolveFor or Target

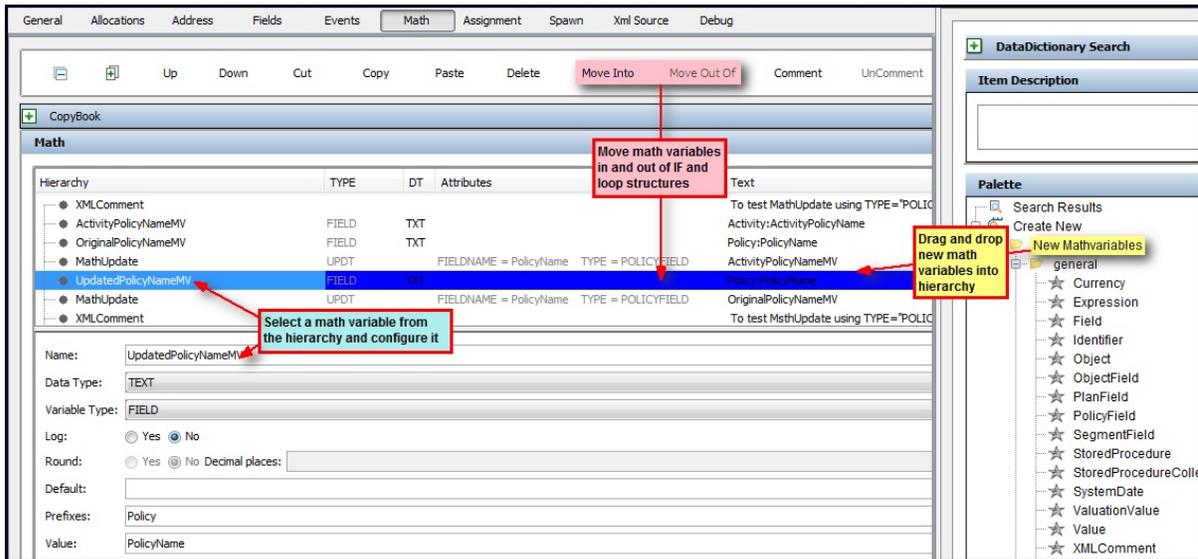
math variable.

- **SQL:** Math variables that retrieve results from a SQL statement.
 - **Update:** Math variables that allow values to be stored in memory and updated without affecting the database values.
 - **WebService** Math variables that create a Web Service call math variable.
 - **XML:** Math variables that use XPATH to obtain a value from an XML blob. This should only be used with external calls and not in general configuration.
5. Depending on the type of math variable selected there will be various operations, behaviors and characteristics defined. The following are required:
- Name of the math variable.
 - Select the DataType if necessary.
 - Math variable type automatically is selected based on the drag and drop variable that was selected.
 - Only those attributes associated with TYPE will be displayed and available for entry.



The scroll bar for the Math pane is hidden from view. To scroll to the bottom of the Math Pane, click the scroll bar at the bottom of the window and drag it to the right. This will reveal the vertical scroll bar used to move to the bottom of the Math Pane.

6. Move the math variables in the hierarchy using the **Up** or **Down** buttons at the top of the Configuration Area.
7. Create or delete math folder structures by using the **Move Into** and **Move Out Of** buttons at the top of the Configuration Area. Move Into means that a math variable will be moved inside a MathLoop or MathIF. Move Out Of means that a math variable will be moved outside and above the MathLoop or MathIF that it is currently inside.



Functionality of Math Pane

- Use the **Comment** and **Uncomment** buttons to comment or uncomment a math variable.

Use an Existing Math Variable

From the SearchPalette window, use the [DataDictionary](#) to search for existing math variables. If the exact math variable name is unknown, then use the % sign as a wildcard. This wildcard can be used at the beginning and/or the end of a partial search term. The category section can also be searched to find a math variable term name.

Steps to Use an Existing Math Variable

1. Open the SearchPalette window. This can be done by clicking **Window** on the Main Menu and selecting **Open SearchPalette Window**. Open a transaction in the Configuration Area. Make sure to check-out the transaction or updates cannot be made.
2. Click the Math pane at the top of the window.
3. Click the plus box  to expand the Data Dictionary search section at the top of the SearchPalette window.
4. Search for the math variable using the **Search Field** text box or Search Category list.
5. Select the **Search** button.
6. Select the math variable from the **Search Results** section in the **Palette** window.
7. Drag and drop the math variable from the **Palette** window to the Configuration Area.
8. Use the **Up** or **Down** button to place the math variable in the correct spot in the hierarchy.
9. Edit the math variable properties as necessary.



When configuring a general math variable that is a PlanField or PolicyField type, you can use the auto-complete feature. Hitting Ctrl + Space will bring up the plan's policy fields or plan fields (not mathvariables). The auto-complete feature is also available for the Field type MathVariable, in which case, depending on the prefix that is used (Activity, Policy, Plan), auto-complete will bring up the Activity, Policy or Plan fields. StoreProcedure and StoredProcedureCollection also have drop-down boxes populated with all available stored procedures. An array with Operation FillyBy-StoreProcedure also has the StoredProcedure dropdown available.



When checking-out a transaction that contains a Disbursement section, a pop-up notification will display indicating that there is a

Disbursement section in the transaction and that the XML Source pane should be used to edit that section. When the XML Source pane is open, scroll to the Disbursement section and make the changes. Check the transaction back in to save the changes.



Disbursement Message

Cut, Copy and Paste Math Variables

The Math pane supports the ability to cut, copy and paste math variables. Math variables can be pasted into either the same business rule from which they were cut/copied, or into the Math pane of a different rule. Click and drag the mouse, or use the **Ctrl** or **Shift** key, to select multiple math variables.

To cut math variables:

1. Check out the rule containing the math variable(s) to be cut.
2. Select the math variable(s) to cut.
3. Click the **Cut** button or press **Ctrl-X** on the keyboard. The selected math variable(s) will be removed and copied to the system's clipboard.

To copy math variables:

1. Select the math variable(s) to copy. The rule does not need to be checked out to use the Copy function.
2. Click the **Copy** button or press **Ctrl-C** on the keyboard. The selected math variable(s) will be copied to the system's clipboard.

To paste math variables:

1. Check out the rule into which the math variable(s) will be pasted. Math variables can only be pasted into other math sections—an error message will be displayed if the user attempts to paste math variables into a different configuration section.
2. Select the math variable directly above the desired location for the cut/copied math variables.
3. Click the **Paste** button or press **Ctrl-V** on the keyboard. The math variables will be pasted below the previously selected line, and will be highlighted.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Math Pane > AllocationField

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

AllocationField

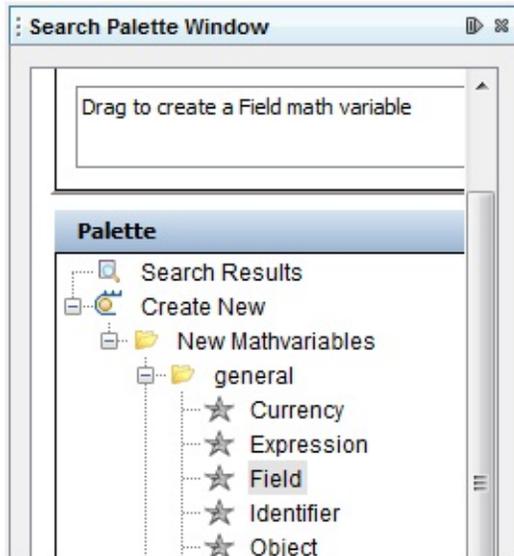
Allocations will be available in screen and transaction math using FillBy Operations. When the funds of the allocation records are in memory, the funds value is parsed through a "For" type loop using an aggregate count of the length of the array. so within this Math Loop, a new math type of **AllocationField** is available to refer the values in the field columns of the activity allocation records in memory. The value of the field may be accessed via screen or transaction math using the new allocation field type where the math variable value will be the name of the allocation column to return. The field syntax like FundGUID, TypeCode, AllocationMethodCode, PercentInAllocation are always be available e. and the AllocationAmount, AllocationPercent, AllocationUnits may be Null if the AllocationMethod does not allow it.

An **AllocationType** attribute will indicate if the allocation values are to be returned from the positive and negative allocations

From : This field is required to retrieve the negative allocation value from allocation records

TO : This is the default value and is required to retrieve the positive allocation value from allocation records

The MathVariable type attribute needs to support a new value "**AllocationField**". Within the general node of the Palette, a new ALLOCATIONFIELD is listed.



Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Math Pane > Currency

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Currency

The currency math variable is used to define operations that [convert](#) a value from one currency to another using specific exchange markets and dates.



Currency Math Variable in Palette



Currency conversion only converts other math variables. A field needs to be mapped to a math variable before it can convert to another currency.

Define the Currency Math Variable

Drag and drop the Currency math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- Make sure data type is defined as Currency. This is required.
- Make sure variable type is defined as Currency. This is required.
- Select **Yes** or **No** from Log field. Default is No. This is an optional field.
- Make sure the operation is defined as **Convert**. This is required.
- Type the currency code that the currency is coming from in the From Currency field. Currency codes are listed in the [Currency editor](#) in Admin Explorer. **This is required only if the value is not already defined as a currency.**
- Type the currency code that the currency should be converted to in the To Currency field. Currency codes are listed in the [Currency editor](#) in Admin Explorer. This is required.
- Type a Market Maker code in the Market Maker field to identify the Market Maker to use during conversion. Market Maker codes are listed in the [Market Maker editor](#) in Admin Explorer. This is required.
- Type the date the system will use for the currency conversion calculation in the Exchange Date field. This is optional.



If the exchange date is specified, rates must exist for that date or the exchange cannot process. If the exchange date is not specified, the system will use the exchange rates available as of the current system date when the activity processes.

- Enter the value of the math variable. This is required.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > Expression

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Expression

The Expression math variable is used to perform calculations. Expression can return one value. The Expression math variable allows for adding, subtracting, multiplying, or dividing previously defined math variables or literal numeric values. Expression can also be used to copy the value of one math variable into another.



Expression Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the Expression Math Variable

Drag and drop the Expression Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- The data types are Activity, Boolean, Date, Decimal, Integer, Map, Object, and Text. This is required.
- Make sure variable type is defined as Expression. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the options of **Yes** or **No**. Round is only enabled if you select the Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to round the result value. **Note:** When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- In the **Expression** property, enter the calculation or arithmetic you would like to perform using previously defined Math variables, activity fields, hard-code numbers, and operators.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > Field

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Field

The Field math variable allows for an activity, policy, plan, or valuation value to be retrieved and used in subsequent Math calculations. The field must be defined first within the activity, plan screen, policy screen, or valuation prior to using it within the Math.



Field Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the Field Math Variable

Drag and drop the Field Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a **Name** in the Name field. This is required. Note: The name of Math variable for a Field you wish to use as defined in the activity must have a different naming convention in the Math. For example, if the name of the Field in the activity is IssueDate and you wish to pull IssueDate in to the math for further manipulation, you must name the Math variable some other than IssueDate. You may use MV at the end of the Math variable name to indicate Math Variable, IssueDateMV.
- The data types are Activity, [BigText](#), Boolean, Date, Decimal, Integer, Map, Object, and Text. This is required.
- Make sure variable type is defined as Field. This is required.
- Select Yes or No from the **Log** field. The default is No. This is an optional field.
- **Round** offers the options of **Yes** or **No**. Round is only enabled if you select the Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to round the result value. **Note**: When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- **Default** is used for a Decimal or Integer value so the system will not return a null value. For example, the default for an Integer data type could be -999999999.
- Type in the following for the **Prefix**: Activity:, Plan:, Policy:, or Valuation:.
- The **Value** is the name of the Field you wish to retrieve. Type in the Name of the field.

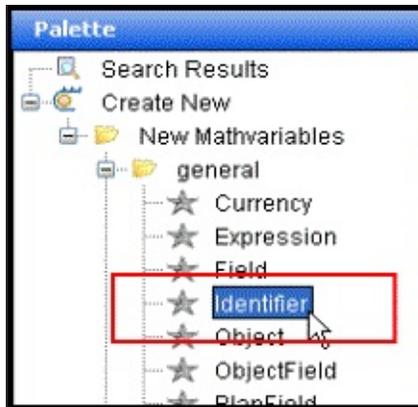
Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > Identifier

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

IDENTIFIER Math Variable

The IDENTIFIER math variable is typically used to generate confirmation numbers for spawned activities using alphabetic and/or numeric parts. The numeric parts may be incremented in order to provide a unique ID constructed with the same elements.



IDENTIFIER Option in Palette Window

Prerequisites

When setting up a new Identifier, you must insert the first record into the AsSequence table, then add the database component that allows it to work correctly. The scripts required for each type of database are provided below.

ALL (ORACLE, DB2, SQL):

```
Insert into ASSEQUENCE (SEQUENCENAME, SEQUENCEINTEGER, SEQUENCEINTEG  
Values ('ConfirmationNumber', 1, 'ConfirmationNumber', 'Conf.
```

```
Insert into ASSEQUENCE (SEQUENCENAME, SEQUENCEINTEGER, SEQUENCEINTEG  
Values ('ConfirmationNumberMV', 1, 'ConfirmationNumberMV', 'C
```

ORACLE and DB2 Only:

```
Create sequence ConfirmationNumber start with 1 increment 1
```

```
Create sequence ConfirmationNumberMV start with 1 increment 1
```

Configuration Considerations

Keep in mind the following points when configuring IDENTIFIER math variables.

1. An IDENTIFIER math variable should be used to confirmation numbers for spawned activities.
2. An Identifier field should be used to generate the originating (parent) activity's confirmation number.
3. An IDENTIFIER math variable can be used in ScreenMath.
4. The Identifier field configuration can include the `<ClearOnRecycle>` element that will clear a field when an activity is manually recycled, allowing for the generation of a new confirmation number in such cases. When an activity is recycled due to a system-generated undo/redo, the original confirmation number is retained.
5. Each Part is added to the Parts node in the math variable window of the Rules Palette Math pane. Each type translates to a `<Part>` element in the XML.
6. Identifier MathVariable configuration includes `LOG="Yes"` attribute, in case an activity is re-done due to system-generated undo/redo. The original Identifier MathVariable value can be accessed in the database via configuration.

Define IDENTIFIER Math Variable and Parts

Drag and drop the IDENTIFIER math variable from the Palette window onto the configuration area of the Math Pane. Enter the math variable information such as the math variable name and Log value (Yes or No). The data type and variable type are preset. (For additional information on adding math variables, see [Math Pane](#).)

Click on the Parts node and begin to configure the individual components that will make up the unique identifier. An explanation of each type is given in the section below. Click the **Add** button after each individual part is added until all parts are complete.



If the fields under the Parts node are not visible, drag the bottom scroll bar all the way to the right. Then, use the vertical scroll bar to scroll down to reveal the fields.

Hierarchy	TYPE	DT	Attributes
• NewMathVariable	IDNT	TEXT	

Name:

Data Type:

Variable Type:

Log: Yes No

Type:

Parse: Parse Value:

Format:

Prefix:

Value:

IDENTIFIER Math Variable in Math Pane

IDENTIFIER Parts

The following list explains the various IDENTIFIER Parts options. Each type is listed, along with the configuration options that accompany the type. Each type translates to a <Part> element in the XML.

- **When Type=Value is selected:**

- The value of the <Part> element is the literal text, entered in the Value field, that is concatenated to the IDENTIFIER math variable.

- **When Type=SystemDate is selected:**

- **Format:** Portions of the system date or all of the system date may be concatenated to the IDENTIFIER math variable. The required attribute FORMAT defines the portions that will make up the resulting value to concatenate. The following is a list of the attribute values and what they accomplish.

- YY and YYYY results in 2 and 4 digit years, respectively.
 - MM results in a 2 digit month.
 - DD results in a 2 digit day.
 - YYMM and YYYYMM results in a 2 digit year, 2 digit month value or a 4 digit year, 2 digit month value, respectively. Both create a value in year-month order.
 - YYYYMMDD results is a full date value with a 4 digit year in year-month-day order.
 - YYMMDD results is a full date value with a 2 digit year in year-month-day order.

- **When Type=Sequence is selected:**

- A sequence number is generated from a sequence set identified by the sequence name referenced in the <Part> element's value. Once a sequence number becomes part of the value of an IDENTIFIER math variable, it cannot be duplicated by any subsequent IDENTIFIER math variable using the same sequence set. Concurrent accesses to the same sequence will yield different sequence values.

A required FORMAT attribute specifies the padding characters and length of the output that is concatenated to the IDENTIFIER math variable. The padding character may be any visible character or mix of characters that define the padding character(s). The length of the output is defined by the number of characters in the FORMAT attribute's value. For example, if FORMAT="000", the complete length of the <Part> value is 3. If the sequence number is 1, then the output is 001. If the sequence number is 100, then the output is 100.

The optional Sequence Date Prefix text box allows entry of a context prefix (Activity, Policy, Plan, or screen math ID). This is a freeform text entry field.

The optional Sequence Date text box filters the sequence records for a specific sequence set. It may be used to restart the sequence numbering as of a specific date. The value of the attribute indicates the data source for the date. One data source for the attribute's value is a field name. The field must be defined prior to the IDENTIFIER math variable and must be a date data type. Other data sources for the attribute's value may be the system date and an activity's effective date.

The following table shows data that resides in the sequence data store for a specific sequence set.

	Sequence date	Sequence number
1	1/1/2008	501
2	1/1/2009	25
3	1/1/2010	1

Sequence Data Store for a Sequence Set

- For any date value provided to SEQUENCEDATE that falls between 1/1/2008 and 12/31/2008, inclusively, record #1 is used to supply sequence numbers. The next number is 501.
- For any date value provided to SEQUENCEDATE that falls between 1/1/2009 to 12/31/2009, inclusively, record #2 is used to supply sequence numbers. The next number is 25.

- For any date value provided to SEQUENCEDATE that is equal to or greater than 1/1/2010, record #3 is used. The next number is 1.
- Any positive integer value may be used as a starting value. Maximum value for a sequence number is about 2.1 billion.

The Value text box allows entry of sequence names from AsSequence, such as PolicyNumber, SuspenseNumber, etc.

■ **When Type=Field is selected:**

Parse: select **Left, Right, Mid** and enter a Parse Value. These indicators are used to tell the system what part of the value to concatenate to the identifier's field value. A sample of each option is given below:

- **Left & Parse Value:** specifies the number of characters extracted from the start of the field's value. Ex: If the Field value is **12345**, then LEFT with Parse Value=2 would mean that **12** is extracted and concatenated to the identifier's value.
- **Right & Parse Value:** specifies the number of characters extracted from the end of the field's value. Ex: If the Field value is **12345**, then RIGHT with Parse Value=2 would mean that **45** is concatenated to the identifier's value.
- **Mid & Parse Value:** specifies the starting and ending location in the field's value from which to extract the value. Ex: If the Field value is **12345**, then MID with Parse Value 2 and 5 would mean that **2345** is concatenated to the identifier's value.

Format: indicates the padding character and the length of the resulting value. Any character or mixture of characters define the padding character(s). The length of the output is defined by the number of characters in the FORMAT attribute's value. Padding will always occur to the left of the parsed value to create a result that is the length of FORMAT attribute. For example:

- FieldA contains the value of **Mon**.

- FieldB is an identifier data type field with a Part type of Field defined with a FORMAT=**ddddd**.
- The resulting value is **ddMon**, which is then concatenated to the IDENTIFIER math variable's value.
- The padding character is **d** and the resulting value's length is 5 because **d** was repeated 5 times in the FORMAT attribute's value.

Value: The name of the field whose value may be parsed and padded, then concatenated to the IDENTIFIER math variable's value.

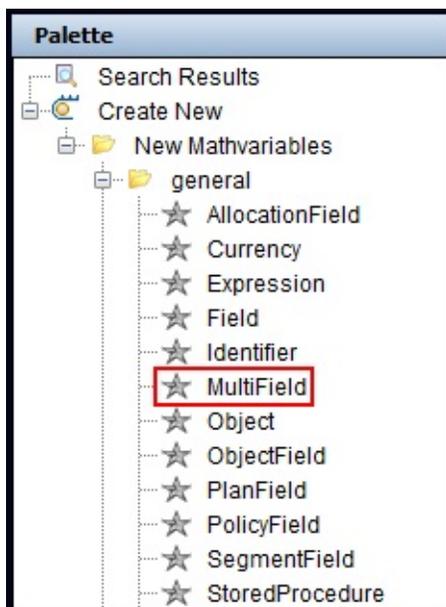
You are here: Configuration > Configuration Panes > Math Pane > MultiField

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

MultiField

The MULTIFIELD math variable is used to access multifield values (using INDEX as an attribute) in Screen Math and Math sections.

The INDEX attribute can be any integer value ranging from 0-99. When the INDEX attribute is equal to 0, the first instance of a multifield will be retrieved. Each additional instance of a multifield will be accessed by incrementing the INDEX value in a subsequent math variable. A math variable that is configured with an INDEX greater than the number of instances of the multifield selected on the activity will result in the value of the math variable being equal to null. For math variables that are data type STRING or DATE, null is a valid value and no error will be returned. For math variables that are data type DECIMAL or INTEGER null is not a valid value and a system error will be returned, unless a default value is specified using configuration. A math variable that is configured with the TYPE equal to MULTIFIELD, but has no INDEX attribute specified will result in a code-generated system error.



Field Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the **Help** menu of the

Rules Palette.

Define the Field Math Variable

Drag and drop the MultiField Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable:

- Type a **Name** in the Name field. This is required.
- The data types are Activity, BigText, Boolean, Date, Decimal, Integer, Map, Object, Text and XML. This field is required.
- Make sure variable type is defined as MULTIFIELD. This is required.
- Specify an index. This specifies the particular instance of the multifield to access.
- Select Yes or No from the **Log** field. The default is No. This is an optional field.
- **Round** offers the options of **Yes** or **No**. Round is only enabled if you select the Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to round the result value. **Note:** When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- **Default** is used for a Decimal or Integer value so the system will not return a null value. For example, the default for an Integer data type could be -999999999.
- **MultiField** is the name of the MultiField you wish to retrieve. Type in the Name of the field.

Check the transaction in to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > Object

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Object

The Object math variable sets up a discrete item so data within an OBJECTLOOP can be accessed via OBJECTFIELD. Many items associated to a policy are multiple instances of complex data types. The Object Math variable type returns a single instance of an object from the object collection (Source Array).



Object Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the Object Math Variable

Drag and drop the Object math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a **Name** in the Name field. This is required.
- Make sure variable type is defined Object. This is required.
- The **Data Type** selection must be Object. This property is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the option of **Yes** or **No**. Round is only enabled if you select the Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to round the result value.

Note: When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.

- **Object Name** is the Name of the database table you would like to access. The table name on which the loop will occur. The "As" table prefix must be dropped / not be used. This is a required property.
- The **SourceArray** will define the name of the previously defined ObjectLoop. The source array will represent the name of a previously defined Math variable to the type of MathLoop equal to Object.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > ObjectField

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

ObjectField

The ObjectField math variable provides a method for retrieving values from an object instance created in a previously defined Object math variable.



ObjectField Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the ObjectField Math Variable

Drag and drop the ObjectField math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the **Name** field. This is required.
- Make sure variable type is defined ObjectField. This is required.
- The **Data Type** selections are: Activity, Boolean, Date, Decimal, Integer, Map, Object, and Text. This property is required.
- **Round** offers the option of **Yes** or **No**. Round is only enabled if you select the Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to round the result value. **Note:** When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- The **Source Object** will define the name of the previously defined Object Loop. The source array will represent the name of a previously defined Math variable to the type of Math Loop equal to Object.
- The **Field Name** is the name of the Field you wish to extract the value where the data resides. The field name you wish to retrieve from the table.
- **Object Name** is the Name of the Math variable where TYPE="Object". The name of the object on which the loop will occur. This is the name of the OIPA table that will be iterated through.
Note: Do not use the "As" prefix in the table name.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > PlanField

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

PlanField

The PlanField allows for assigning the value of a field from AsPlan or AsPlanField (Plan Data) to a Math variable so the value can be used in to perform calculations. The PlanField Math variable has been designed for efficiency and therefore its use is preferred over an SQL statement in the configuration.



PlanField Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the PlanField Math Variable

Drag and drop the PlanField Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the **Name** field. This is required.
- The applicable **Data Types** are Activity, Boolean, Date, Decimal, Integer, Map, Object, and Text. This is required.
- Make sure variable type is defined as PlanField. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the option of **Yes** or **No**. Round is only enabled if you select the Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to round the result value. **Note:** When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- **Default** is used for a Decimal or Integer value so the system will not return a null value. For example, the default for an Integer data type could be -999999999.
- The PlanField **Value** is the name of the Field you wish to retrieve. Type in the Name of the field that resides in AsPlanField table or a column on AsPlan.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > PolicyField

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

PolicyField

The PolicyField allows for assigning the value of a field from AsPolicy or AsPolicyField to a math variable so the value can be used to perform calculations. The PolicyField math variable has been designed for efficiency and therefore its use is preferred over an SQL statement in the configuration.



PolicyField Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the PolicyField Math Variable

Drag and drop the PolicyField Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the **Name** field. This is required.
- The **Data Types** are: Activity, Boolean, Date, Decimal, Integer, Map, Object, and Text. This is required.
- Make sure variable type is defined as PolicyField. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the options of **Yes** or **No**. Round is only enabled if you select the Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to round the result value. **Note:** When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- **Default** is used for a Decimal or Integer value so the system will not return a null value. For example, the default for an Integer data type could be -999999999.
- The PolicyField **Value** is the name of the Field you wish to retrieve. Type in the Name of the field that resides in AsPolicyField or a column on AsPolicy.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > SegmentField

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

SegmentField

The SegmentField allows for assigning the value of a field from AsSegment or AsSegmentField to a math variable so the value can be used to perform calculations. The SegmentField math variable has been designed for efficiency and therefore its use is preferred over an SQL statement in the configuration.



SegmentField Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the SegmentField Math Variable

Drag and drop the SegmentField Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the **Name** field. This is required.
- The applicable Data Types are: Boolean, Date, Decimal, Integer, Map and Text. This is required.
- Make sure variable type is defined as SegmentField. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the option of **Yes** or **No**. Round is only enabled if the user selects the Decimal in the data type property. Select **Yes** to round the Decimal. Enter the number of decimal places to round the result value.
 - **Note:** When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- **Default** is used for a Decimal or Integer value so the system will not return a null value. For example, the default for an Integer data type could be -999999999.
- The **Segment GUID** value is the GUID for the specific segment on the policy that contains the field to be retrieved. The value of this attribute will be a math variable and the Rules Palette provides a list of all configured math variables in the transaction. Only ones containing SegmentGUIDs should be used here.
- The **Source Array** lists all previously defined MathLoop variables. Use the Source Array if you are extracting the value from a field inside a loop and the policy has multiple segments.
- The **Segment Name Filter** provides a drop down selection box pre-populated with all of the currently configured Segment Names for the plan. It is only informational and not part of the configuration of the variable.

Note: When Ctrl+Space bar is clicked , a selectable list of fields for the filtered segment name will appear based on the data type

selected. If the segment name filter is not used, then all fields of that data type for all segments of the plan will be displayed.

- The **Segment Field** is the name of the Field you wish to retrieve. Type in the Name of the field that resides in AsSegmentField.

Check-in the transaction to save the information to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Math Pane > StoredProcedure

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

StoredProcedure

The StoredProcedure allows for execution of a Stored Procedure within the Math to return a single value, one column with multiple rows.



StoredProcedure Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the StoredProcedure Math Variable

Drag and drop the StoredProcedure Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the **Name** field. This is required.
- The applicable **Data Types** are Boolean, Date, Decimal, Integer, and Text. This is required. The data type of the Math variable must be similar to the defined return value from the Stored Procedure.
- Make sure variable type is defined as Stored Procedure. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the option of **Yes** or **No**. Round is only enabled if select the Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to round the result value. **Note:** When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- **Default** is used for a Decimal or Integer value so the system will not return a null value. For example, the default for an Integer data type could be -999999999.
- The **Stored Procedure** value is the name of the Stored Procedure you wish to execute within the Math. A pre-populated drop down selection box is provided. Select the applicable Stored Procedure.
- The **Input** section is required in terms of the Value column. The **Parameter** name is pre-populated according to the requirements defined within the Stored Procedure configuration. The **Data Type** column is pre-populated using the data type for the Parameter as specified within the Stored Procedure configuration. The **Value** column requires that you select from the pre-populated drop down selection box with all previously defined Math variables of the same data type.

Check-in the transaction to save the information to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Math Pane > StoredProcedureCollection

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

StoredProcedureCollection

The StoredProcedureCollection math variable allows for the execution of a Stored Procedure within the Math to return multiple columns with multiple rows of data. A Collection in Math is a series of key-value pairs where each key is unique. The first column from the data return is the key and the second is the value.



StoredProcedureCollection Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the StoredProcedureCollection Math Variable

Drag and drop the StoredProcedureCollection Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the **Name** field. This is required.
- The **Data Type** is MAP, always.
- Make sure variable type is defined as Stored Procedure Collection. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the option of **Yes** or **No**. Round is only enabled if select the Decimal in the data type property. If you would like to round the Decimal or Integer, select **Yes**. Enter the number of decimal places to round the result value. **Note:** When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- **Default** is used for a Decimal or Integer value so the system will not return a null value. For example, the default for an Integer data type could be -999999999.
- The **Stored Procedure** value is the name of the Stored Procedure you wish to execute within the Math. A pre-populated drop down selection box is provided. Select the applicable Stored Procedure name.
- The **Input** section is required in terms of the **Value** column. The **Parameter** name is pre-populated according to the requirements defined within the Stored Procedure configuration. The **Data Type** column is pre-populated using the data type for the Parameter as specified within the Stored Procedure configuration. The Value column requires that you select from the pre-populated drop down selection box with all previously defined Math variables of the same data type.
NoteThe Math variable filing the parameter can be a previously defined Math variable, a literal value, or an expression.

Check-in the transaction to save the information to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Math Pane > SystemDate

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

SystemDate

The SystemDate math variable is used to pull the system date in effect from the AsSystemDate table when the activity math processes. This date can be used in Validations or for further manipulation of the date value. For example, using the SystemDate within a Function such as DaysAdd(SystemDateMV,BillLeadDays).



SystemDate Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the System Date Math Variable

Drag and drop the System Date Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the **Name** field. This is required.
- The applicable **Data Type** is Date. This is required.
- Make sure variable type is defined as SystemDate. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.

Check-in the transaction to save the information to the database.

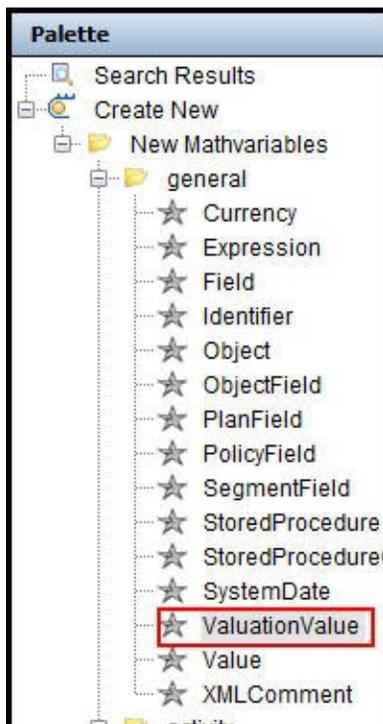
You are here: Configuration > Configuration Panes > Math Pane > ValuationValue

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

ValuationValue

The ValuationValue math variable indicates that the math variable will retrieve a value from the math valuation object. Data created in the [MathStatement](#) valuation can be retrieved using the syntax of the ValuationValue math variable type. This math variable type should only be used in transaction math and not in attached rules or screen math.

Data may only be extracted from the math statement valuation object inside transaction math. It is not available in business rule math sections or spawns.



ValuationValue Math Variable in Palette Window

Define the ValuationValue Math Variable

Drag and drop the ValuationValue math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- The math variable data type will equal that of the data being retrieved from the valuation source. This is required. GUIDs are TEXT and numeric values are DECIMAL.



Currency is not currently supported for values returned from valuation. It is assumed these values are in the plan default currency.

- Select the Variable type from the Variable Type drop down box.
- Select **Yes** if the value should be logged. Otherwise, select No.
- Select **Yes** to round the result then enter the number of decimal places to round. Otherwise, select No.
- Select a context from the Context drop down box. The context tells the math variable to use the math valuation object as the source. The value of this attribute must be the literal MATH. This is required.
- Select a Valuation Type from the Valuation Type drop down box. This indicates the kind of value to retrieve.
 - BUCKET: indicates that the value to return is related to a specific fund GUID and bucket (as done in activity valuation using Valuation:Fund:[FundGUID]:Bucket). Available values for a math variable of this valuation type are CashValue and blank. When this type is selected, the following additional fields display.
 - BucketNumber: this can be a literal integer, field or math variable containing an integer.
 - FundGUID: the value of the FUNDGUID attribute can be a field or math variable containing a fund GUID.
 - DEPOSIT: indicates that the value to return is related to a specific valuation GUID (as done in activity valuation using Valuation:Deposit). Available values for a math variable of this valuation type are blank, ValuationGUID, EffectiveDate,

RateLockDate, MoneyTypeCode, InitialUnits, InitialCashValue, Units, CashValue, SurrenderRate, SurrenderValue, Principal, Gain, NetGain, InterestBonus, TransactionName, MVAAmount, RedemptionAmount, ShortTermRedemptionUnits, GuaranteedAmount, ActivityGUID.

When this type is selected, the following additional field displays.

- ValuationGUID: the value of the VALUATIONGUID attribute may be a field or math variable containing a Valuation GUID.
 - FUND: indicates that the value to return is related to a specific fund GUID (as done in activity valuation using Valuation:Fund). Available values for a math variable of this valuation type are blank, FundName, TypeCode, UnitValue, Units, CashValue, SurrenderValue, Principal, Gain, NetGain, FreeAmount, MVAAmount, RedemptionAmount, ShortTermRedemptionUnits, GuaranteedAmount. When this type is selected, the following additional field displays.
 - FundGUID: the value of the FUNDGUID attribute can be a field or math variable containing a fund GUID.
 - POLICY: indicates that the value to return is from the policy valuation (as done in activity valuation using Valuation:Policy). Available values for a math variable of this valuation type are blank, EffectiveDate, ActiveDate, CashValue, SurrenderValue, Principal, Gain, NetGain, FreeAmount, MVAAmount, RedemptionAmount, GuaranteedAmount.
 - POLICYVALUE: indicates that the value to return is from the PolicyValues rule (as done in activity valuation using Valuation:PolicyValues). Available values for a math variable of this valuation type are blank and any math variable name from the PolicyValues business rule.
- A DEFAULT attribute may optionally be included in the math variable in case the returned valuation data is null.
 - The value should be entered in the Value field.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > Value

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Value

The Value math variable provides a literal value stored in a variable. The hard-coded or literal value can be initialized at the beginning of Math and then reset based on a condition, calculation, or mapping. The system will store the value of the last instance of the math variable in the Activity math.



Value Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the Value Math Variable

Drag and drop the Value Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the **Name** field. This is required.
- The applicable **Data Types** applicable are Activity, Boolean, Date, Decimal, Integer, Map, Object, and Text. This is required.
- Make sure variable type is defined as Value. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the option of **Yes** or **No**. Round is only enabled if select the Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to round the result value.
 - **Note:**When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- Enter the measurement in terms of the **Value**. For example, a numeric measurement of 0, or a text measurement such as PremiumPayment (the activity name).

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > XMLComment

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

XMLComment

The XMLComment math variable provides commentary, an explanation, or interpretation about a variable or the entire transaction. For example, a comment may provide information instructing the configuration analyst on what specifically the next section of configuration is calculating, a reference to an interface that needs to occur, or a 'to be determined' set of configuration lines.



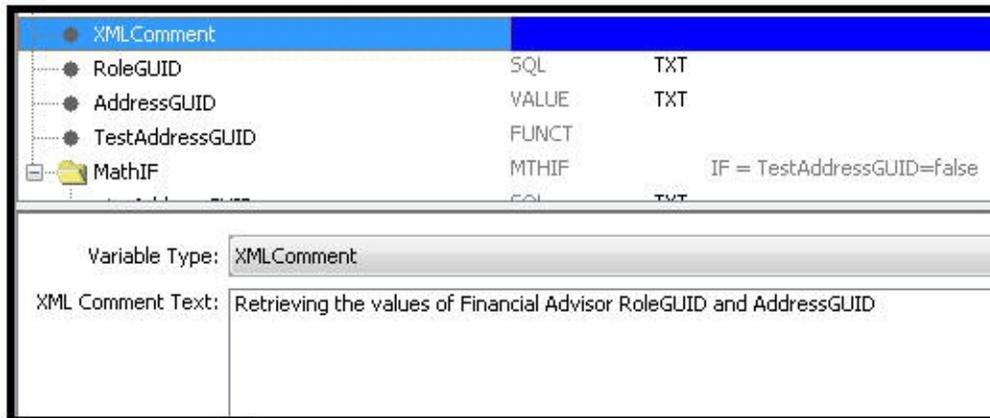
XMLComment Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the XMLComment Math Variable

Drag and drop the XMLComment Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Make sure variable type is defined as **XMLComment**. This is required. XMLComment is the default.
- Enter the explanation or information in the text box.



Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > XPath

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

XPATH

The XPATH math variable is part of a math syntax that can parse XML by variable name and extract its value into a math variable. It is used in conjunction with a math variable with the [XML datatype](#), which allows OIPA to treat an XML string returned from an external source as an XML object. The XPATH math variable will parse through the stored XML as a source object for the data element name and return the value from the XML.



This math variable is most effectively used in situations where an external call is received by OIPA and provides information that needs to be parsed. It is not recommended for general configuration.

Configuration Considerations

The supported data types for XPATH are DATE, TEXT, INTEGER and DECIMAL. The SOURCE attribute is required and should be a math variable with a data type of XML. Each value that needs to be extracted from the SOURCE math variable will require a separate call to the XPATH math variable.

The CURRENCY data type is not supported with XPATH. To create a math variable with a CURRENCY data type, three steps must be performed.

1. Extract the numeric value from XML.
2. Extract the currency code from XML.
3. Use ToCurrency to combine the numeric value and the currency code to create a CURRENCY data type.

Define the XPATH Math Variable

Drag and drop the XPATH math variable from the Palette window onto the Math pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required. The name will populate the VARIABLENAME attribute of the XML math variable for TYPE="XPATH".
- The data types applicable are Date, Text, Integer and Decimal. This is required. The data type will populate the data type attribute of the XML math variable for TYPE="XPATH".
- Make sure variable type is defined as XPATH. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- Type a default value in the Default field, if needed.
- Click in the XML Source field and press **CTRL spacebar** to see a list of available XML math variables that can be used as the source.
- Click in the Value field and type the value you want to assign to the math variable. Value can be literal or it can be the name of a field or math variable.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > Activity

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Activity

The Activity Math variable represents an object containing a group of activity fields for use with spawning multiple activities in creating an array and populating the fields for the spawned activities. The activity fields are defined in the configuration of the transaction to be spawned.

Works to achieve multi spawning activities [ActivityArray](#).



Activity Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the Activity Math Variable

Drag and drop the Activity Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- The data types applicable are Activity, Boolean, Date, Decimal, Integer, Map, Object, and Text. This is required.
- Make sure variable type is defined as Activity. This is required.
- Select the applicable Operation from the drop down selection box. The options are: Create and SetValue.
 - **Create** provides a Method in which you can select a **GUID** or **Transaction Name** radio button. Selecting GUID allows you to populate the Transaction text box with a TransactionGUID. Selecting Transaction Name provides you with a drop down selection box of currently configured Transactions. You may also, in either case, supply a previously defined Math variable that represents the TransactionGUID or TransactionName.
 - **SetValue** allows you to designate a field and the **Value** for the field as it will replace any default calculated value in the Activity Math object.



The value of an Activity math object cannot be evaluated inside a MathLoop. If the Set Value requires a calculation, the calculation must be done first with another math variable type (other than MathLoop, or MathIF) which will be referenced by the Set Value operation. For example, if you wish to manipulate the Effective Date by adding days, the Function must be handled first by another math variable and then Set Value can reference the Function math variable's name in the Value field.

Math Loop with Activity and ActivityArray: In order to populate the array with more than one Activity math object, a standard loop can be used (optional).

- Steps include: Create the ActivityArray, Begin the MathLoop, Create the Activity Object, Set Value of the Activity Object, Add the object to

the ActivityArray, and Close the MathLoop.

- Note: The ActivityArray variable must be instantiated outside of the loop and the Activity object math object inside the loop.

Check-in the transaction to save the information to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Math Pane > ActivityArray

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

ActivityArray

Activity Array allows for the spawning of the same transaction multiple times from a single activity. By allowing each spawned transaction to reference unique data sets, any number of transactions may be spawned any number of times on the policy from a single parent transaction. For example, this could be used for an annuity payout that has multiple individuals receiving disbursements, to ensure both individuals get their correct disbursement amounts. Also, multiple activities could be spawned for Bank Drafts, Invoices or Anniversaries.



This replaces Spawn Code 08 that was used with Version 8 and prior releases. This also deprecates the use of the MultiRowProcessing business rule.

To configure multiple spawning activities, the following three MathVariable types are needed :

- **Activity** MathVariables contain instructions to create and set up Activity Math objects for a given transaction rule.
- **Activity Array** holds multiples of the activity math type record (Activity Math Object).
- **ActivityField** extracts the value of a specified field within the Activity Math object.

When configuring the creation of the Activity Array the following logic is needed:

1. Create the Activity Array.
2. Open the Loop.
 - Create the Activity Math object.
 - Set values of the Activity Math object.
 - Add an Activity Math object to the Activity Array.
3. Close the Loop.

To extract values, use the ActivityField MathVariable.

The Activity Array math variable is an arrangement of one or more Activity type math objects. Once field values are set using the Activity math variable, the object is added to the ActivityArray math variable to be used in the spawn section of the parent transaction.



ActivityArray Math Variable in Palette



For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the ActivityArray Math Variable

Drag and drop the ActivityArray math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- The data types must be Activity.
- Make sure variable type is defined as ActivityArray. This is required.
- The applicable Operation selections are: **Create**, **Append**, and **Insert**. First, **Create** the ActivityArray before the array is fed into the Spawn section of the parent transaction. Once the array is created, **Insert** an object into the array. Insert allows an **Index** or location of the object to be specified. The Index property must be populated when using the Insert Operation. **Append** will place the object at the end of the array.

Aggregate Function methods **Count** and **Index** can be used with an ActivityArray.

- **Count**: Returns the number of records in the array.
- **Index**: Returns the value of the specific number element. Indexing allows the system to identify the value by the elements location. Then the value can be used for storage, a subsequent calculation or display purposes.

For example, creating a Disbursement spawn activity array will use a Count of the number of Payee records. Index will return an Activity math object from the referenced location of the array. The Index value must be designated as the numeric location of the object. If there is no object at the referenced index or the array is empty, no value will be returned. Defining a default value object may be required for system stability.

Math Loop with Activity and ActivityArray: In order to populate the array with more than one Activity math object, a standard loop can be used (optional).

- Create the Activity Array.
- Begin the MathLoop.

- Create the Activity object.
- Set the Value of the Activity object.
- Add the object to the Activity Array.
- Close the MathLoop.
- Check-in the transaction to save the information to the database.



The Activity Array math variable must be instantiated outside of the loop and the Activity object math object inside the loop.

You are here: Configuration > Configuration Panes > Math Pane > ActivityField

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

ActivityField

The ActivityField MathVariable is used to extract the value of the field within the Activity math object. You can either use the default value or you can set the value and use ActivityField to pull the 'new' value from the field.

Provides for the retrieval of a field from the [ActivityArray](#) previously defined within the MathVariables.



ActivityField Math Variable in Palette



For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the ActivityField Math Variable

Drag and drop the ActivityField Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- The data types applicable are Activity, Boolean, Date, Decimal, Integer, Map, Object, and Text. This is required.
- Make sure variable type is defined as ActivityField. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- Select the applicable Source, the name of the Object you created with Activity math variable.
- Enter the Name of the Field you wish to extract.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > AggregateFunction

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

AggregateFunction

The AggregateFunction Math variable provides a method for manipulation of an array. The method is defined in the Math variable properties.



AggregateFunction Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the AggregateFunction Math Variable

Drag and drop the AggregateFunction math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- Make sure variable type is defined AggregateFunction. This is required.
- Select the appropriate DataType from the drop down selection box. DataType is required. The applicable data types are: Activity, Boolean, Date, Decimal, Integer, Map, Object, and Text.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the options of **Yes** or **No**. Round is only enabled if you select Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to Round the result value. **Note:** When the data type of Decimal is selected, then the default value for Round is set to **No**. You must change the selection of the radio button to **Yes** if you wish to round.
- Select the appropriate Method. Method is a required property. The options for Method are as follows:
 - Count: Returns the number of elements in the array.
 - Index: Returns the value of the specific number element. Indexing allows the system to identify the value by the elements location, we can then use the value for storage, a subsequent calculation, or display purposes.
 - Max: Returns the highest element value.
 - Min: Returns the lowest element value.
 - Sum: Returns the total value/sum of all elements.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > DateArray

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

DateArray

The DateArray math variable provides a method for creating an arrangement of date values. For example, this could be used as an array created to gather the Premium Payment Dates for initial and additional payments. Operations can perform calculations on the date array values by using the Operation math variable property.

Each array is created using a specific math variable type (ActivityArray, NumericArray, **DateArray**, IntegerArray, or StringArray) as well as the Operation to denote the action executed on the array. The Operation is defined within the math variable properties.



DateArray Math Variable in Palette



For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the DateArray Math Variable

Drag and drop the DateArray math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- Make sure variable type is defined DateArray. This is required.
- Select the appropriate **DataType** from the drop down selection box. DataType is required. The data type must equal Date.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- Select the applicable **Operation**. Operation is a required property. The selections for Operation are as follows:
 - **Append**: Used to add a value to the end of an array. Define the **Source Array**, which is the name of the array the operation is being performed on, if the original array is being altered during the operation. **Value** must be defined with the value, GUID, etc... placed at the end of the array.
 - **Copy**: Takes a Source Array and creates a separate but identical array. The **Source Array** property must be defined with the name of the array the operation is being performed on if the original array is being altered during the operation.
 - **Create**: A new array is created with the length equal to element value and each value defaulting to the value of the **Defaultattribute**. The Default property must be populated with a value to serve as the automatic value to be returned so the system does not return a null value. The **Length** property must be configured to indicate how many elements are within the new array.
 - **Expand**: Used for creating a larger array in repeating the first, middle and last values of an array. Specific Multipliers can be used for the first and last values, but the middle values must be repeated the same number of times. To use Expand, the array must have at least three elements to constitute a first, middle, and last. Define the **Source Array**, which is the name of the array the operation is being performed on, if the original array is

to be copied and expanded during the operation. **Value** is not used if the operation is Expand.



If using Expand, an additional attribute called Multiplier will need to be manually added in the XML Source pane. Multiplier will hold a comma separated list of integer multiplier factors for the first, middle and last elements in the array being expanded. The syntax is MULTIPLIER="X,Y,Z" where X, Y and Z represent the multiple factors for the first, middle and last elements. Example: If the source array contains the element values of (1,2,3,4) and the Multipliers are 2,3,2 then the target array becomes (1,1,2,2,2,3,3,3,4,4).

- **Insert:** A value will be placed into the array at a particular location as indicated by Index. Insert results in a new array equal to the array indicated by the **Source Array** property with an element value placed at the location specified by the index property. **Index** must be populated with the value of the location of the element. The Source Array property must be populated with the name of the array to be altered.
- **Fill-By Deposit:** Fills the array with all deposits for funds on a policy. Fill-By Deposit should be the preferred operation as opposed to Fill-By SQL. The **Value** property must be populated with FundGUID.
- **Fill-By Fund:** Fills the array with Fund GUIDs from a policy where the Funds currently have positive value. Fill-By Fund should be the preferred operation as opposed to Fill-By SQL.
- **Fill-By List:** Manually fills an array with a list of values. These values can be hard-coded (For example: 20) or reside in a previously defined Math variable (For example: "Twenty"). The **Value** property must be populated with the manual listing of items/values. The values within the listing can be previously defined math variables as well as hard-coded values, or a combination of both and the data type of the items must match the data type of the array.

- **Fill-By SQL:** Fills the array with the value of a SELECT query statement. The SQL query statement used must return only one single column and the data type of the value must match the data type of the array. The **SQL Query** property must be populated with the SELECT statement.
- **Fill-By Stored Procedure:** Fills the array with values from the Stored Procedure execution. You must name the procedure in the **Stored Procedure** drop down selection box. The box is populated with a current list of Stored Procedures. **Input** property is required. Defines the input parameter Name, Data Type, and allows for population/configuration of Values (previously defined Math variables).
- **Remove:** Takes a value from a specific element within the array using index to provide the location of the element. The **Index** must be specified by using the location of the element in the property. The **Source Array** name must be populated as the array you are altering with the removal operation.
- **Stride:** Creates a new array containing every *n*th element of SourceArray where *n* is the element value. The **Source Array** property must be defined with the name of the array you are altering. The **Value** property represents the element value defining *n*.
- **Transform:** Copies the array and uses the array for calculations. Copying the array using the Transform operation create a duplicate array with a different Variable Name. Calculations using Transform could include the addition of two arrays or multiplying all values in the array by a number. The **Value** property represents the expression (calculation being performed) or an array name (copy the array then perform the calculation).

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > IntegerArray

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

IntegerArray

The IntegerArray Math variable provides a method for creating an arrangement of integer values. Operations can perform calculations on the integer values as defined using the Operation Math variable property.

Each array will be created using a specific Math variable type (NumericArray, DateArray, IntegerArray, or StringArray) as well as the Operation to denote the action executed on the array. The Operation is defined within the Math variable properties.



IntegerArray Math Variable in Palette



For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the IntegerArray Math Variable

Drag and drop the IntegerArray math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a **Name** in the Name field. This is required.
- Make sure variable type is defined as IntegerArray. This is required.
- Select the appropriate **Data Type** from the drop down selection box. DataType is required. The data type must be Integer. Keep in mind that the maximum value for an integer is 2147483647.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
-
- Select the applicable **Operation**. Operation is a required property. The selections for Operation are as follows:
 - **Append**: Used to add a value to the end of an array. You must define the **Source Array** which is the name of the array the operation is being performed on if the original array is being altered during the operation. **Value** must be defined with the value, GUID, etc... you are placing at the end of the array.
 - **Copy**: Takes a Source Array and creates a separate but identical array. The **Source Array** property must be defined with the name of the array the operation is being performed on if the original array is being altered during the operation.
 - **Create**: A new array will be created with the length equal to element value and each value defaulting to the value of the **Defaultattribute**. The Default property must be populated with a value to serve as the automatic value to be returned so the system does not return a null value. The **Length** property must be configured to indicate how many elements are within the new array.
 - **Expand**: Used for creating a larger array in repeating the first, middle and last values of an array. Specific Multipliers can be used for the first and last values, but the middle values must be repeated the same number of times. To use Expand, the array

must have at least three elements to constitute a first, middle, and last. You must define the Source Array, which is the name of the array the operation is being performed on if the original array to be copied and expanded during the operation. Value is not used if the operation is Expand. **Note:** If using Expand, an additional attribute called Multiplier will need to be manually added in the XML Source pane. Multiplier will hold a comma separated list of integer multiplier factors for the first, middle and last elements in the array being expanded. The syntax is MULTIPLIER="X,Y,Z" where X, Y and Z represent the multiple factors for the first middle and last elements. For example, if the source array contains the element values of (1,2,3,4) and the Multipliers are 2,3,2 then the target array becomes (1,1,2,2,2,3,3,3,4,4).

- **Insert:** A value will be placed into the array at a particular location as indicated by Index. Insert results in a new array equal to the array indicated by the **Source Array** property with an element value placed at the location specified by the index property. **Index** must be populated with the value of the location of the element. The Source Array property must be populated with the name of the array to be altered.
- **Fill-By Deposit:** Fills the array with all deposits for funds on a policy. Fill-By Deposit should be the preferred operation as opposed to Fill-By SQL. The **Value** property must be populated with FundGUID.
- **Fill-By Fund:** Fills the array with Fund GUIDs from a policy where the Funds currently have positive value. Fill-By Fund should be the preferred operation as opposed to Fill-By SQL.
- **Fill-By List:** Manually fills an array with a list of values. These values can be hard-coded (For example: 20) or reside in a previously defined Math variable (For example: "Twenty"). The **Value** property must be populated with the manual listing of items/values. The values within the listing can be previously defined math variables as well as hard-coded values, or a combination of both and the data type of the items must match the data type of the array.

- **Fill-By SQL:** Fills the array with the value of a SELECT query statement. The SQL query statement used must return only one single column and the data type of the value must match the data type of the array. The **SQL Query** property must be populated with the SELECT statement.
- **Fill-By Stored Procedure:** Fills the array with values from the Stored Procedure execution. You must name the procedure in the **Stored Procedure** drop down selection box. The box is populated with a current list of Stored Procedures. **Input** property is required. Defines the input parameter Name, Data Type, and allows for population/configuration of Values (previously defined Math variables).
- **Remove:** Takes a value from a specific element within the array using index to provide the location of the element. The **Index** must be specified by using the location of the element in the property. The **Source Array** name must be populated as the array you are altering with the removal operation.
- **Stride:** Creates a new array containing every *n*th element of SourceArray where *n* is the element value. The **Source Array** property must be defined with the name of the array you are altering. The **Value** property represents the element value defining *n*.
- **Transform:** Copies the array and uses the array for calculations. Copying the array using the Transform operation create a duplicate array with a different Variable Name. Calculations using Transform could include the addition of two arrays or multiplying all values in the array by a number. The **Value** property represents the expression (calculation being performed) or an array name (copy the array then perform the calculation).

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > NumericArray

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

NumericArray

The NumericArray math variable provides a method for creating an arrangement of integer or decimal values. However, within the NumericArray, you can have different data types including date and text values.



Currently, there are no internal validations to prevent the configuration of a NumericArray for non-numeric values, or different data types.

Each array will be created using a specific math variable type (**NumericArray**, DateArray, IntegerArray, or StringArray) as well as the Operation to denote the action executed on the array. The Operation is defined within the math variable properties.



NumericArray Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the NumericArray Math Variable

Drag and drop the NumericArray math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the **Name** field. This is required.
- Make sure variable type is defined NumericArray. This is required.
- Select the appropriate **Data Type** from the drop down selection box. Data Type is required. The applicable data types are: Activity, Boolean, Date, Decimal, Integer, Map, Object, and Text.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the option of **Yes** or **No**. Round is only enabled if you select the Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to round the result value. **Note:** When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- Select the applicable **Operation**. Operation is a required property. The selections for Operation are as follows:
 - **Append:** Used to add a value to the end of an array. You must define the **Source Array** which is the name of the array the operation is being performed on if the original array is being altered during the operation. **Value** must be defined with the value, GUID, etc... you are placing at the end of the array.
 - **Copy:** Takes a Source Array and creates a separate but identical array. The **Source Array** property must be defined with the name of the array the operation is being performed on if the original array is being altered during the operation.
 - **Create:** A new array will be created with the length equal to element value and each value defaulting to the value of the **Default** attribute. The Default property must be populated with a value to serve as the automatic value to be returned so the system does not return a null value. The **Length** property must be configured to indicate how many elements are within the new

array.

- **Expand**: : Used for creating a larger array in repeating the first, middle and last values of an array. Specific Multipliers can be used for the first and last values, but the middle values must be repeated the same number of times. To use Expand, the array must have at least three elements to constitute a first, middle, and last. You must define the Source Array, which is the name of the array the operation is being performed on if the original array to be copied and expanded during the operation. Value is not used if the operation is Expand. **Note:** If using Expand, an additional attribute called Multiplier will need to be manually added in the XML Source pane. Multiplier will hold a comma separated list of integer multiplier factors for the first, middle and last elements in the array being expanded, The syntax is MULTIPLIER="X,Y,Z" where X, Y and Z represent the multiple factors for the first middle and last elements. For example, if the source array contains the element values of (1,2,3,4) and the Multipliers are 2,3,2 then the target array becomes (1,1,2,2,2,3,3,3,4,4) .
- **Insert**: A value will be placed into the array at a particular location as indicated by Index. Insert results in a new array equal to the array indicated by the **Source Array** property with an element value placed at the location specified by the index property. **Index** must be populated with the value of the location of the element. The Source Array property must be populated with the name of the array to be altered.
- **Fill-By Deposit**: Fills the array with all deposits for funds on a policy. Fill-By Deposit should be the preferred operation as opposed to Fill-By SQL. The **Value** property must be populated with FundGUID.
- **Fill-By Fund**: Fills the array with Fund GUIDs from a policy where the Funds currently have positive value. Fill-By Fund should be the preferred operation as opposed to Fill-By SQL.
- **Fill-By List**: Manually fills an array with a list of values. These values can be hard-coded (For example: 20) or reside in a previously defined Math variable (For example: "Twenty"). The

Value property must be populated with the manual listing of items/values. The values within the listing can be previously defined math variables as well as hard-coded values, or a combination of both and the data type of the items must match the data type of the array.

- **Fill-By SQL**: Fills the array with the value of a SELECT query statement. The SQL query statement used must return only one single column and the data type of the value must match the data type of the array. The **SQL Query** property must be populated with the SELECT statement.
- **Fill-By Stored Procedure**: Fills the array with values from the Stored Procedure execution. You must name the procedure in the **Stored Procedure** drop down selection box. The box is populated with a current list of Stored Procedures. **Input** property is required. Defines the input parameter Name, Data Type, and allows for population/configuration of Values (previously defined Math variables).
- **Remove**: Takes a value from a specific element within the array using index to provide the location of the element. The **Index** must be specified by using the location of the element in the property. The **Source Array** name must be populated as the array you are altering with the removal operation.
- **Stride**: Creates a new array containing every *n*th element of SourceArray where *n* is the element value. The **Source Array** property must be defined with the name of the array you are altering. The **Value** property represents the element value defining *n*.
- **Transform**: Copies the array and uses the array for calculations. Copying the array using the Transform operation create a duplicate array with a different Variable Name. Calculations using Transform could include the addition of two arrays or multiplying all values in the array by a number. The **Value** property represents the expression (calculation being performed) or an array name (copy the array then perform the calculation).

Check-in the transaction to save the information to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Math Pane > StringArray

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

StringArray

The StringArray Math variable provides a method for creating an arrangement of data values. For example, a String Array to represent each Segment's Active Code, an array of SuspenseGUIDs, or Requirements. Operations can perform calculations on the string values as defined using the Operation Math variable property.

Each array will be created using a specific Math variable type (NumericArray, DateArray, IntegerArray, or **StringArray**) as well as the Operation to denote the action executed on the array. The Operation is defined within the Math variable properties.



StringArray Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the StringArray Math Variable

Drag and drop the StringArray math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the **Name** field. This is required.
- Make sure variable type is defined StingArray. This is required.
- Select the appropriate **DataType** from the drop down selection box. DataType is required. The data type must equal Text.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- Select the applicable **Operation**. Operation is a required property. The selections for Operation are as follows:
 - **Append**: Used to add a value to the end of an array. You must define the **Source Array** which is the name of the array the operation is being performed on if the original array is being altered during the operation. **Value** must be defined with the value, GUID, etc... you are placing at the end of the array.
 - **Copy**: Takes a Source Array and creates a separate but identical array. The **Source Array** property must be defined with the name of the array the operation is being performed on if the original array is being altered during the operation.
 - **Create**: A new array will be created with the length equal to element value and each value defaulting to the value of the **Default** attribute. The Default property must be populated with a value to serve as the automatic value to be returned so the system does not return a null value. The **Length** property must be configured to indicate how many elements are within the new array.
 - **Expand**: Used for creating a larger array in repeating the first, middle and last values of an array. Specific Multipliers can be used for the first and last values, but the middle values must be repeated the same number of times. To use Expand, the array must have at least three elements to constitute a first, middle, and last. You must define the Source Array which is the name of

the array the operation is being performed on if the original array to be copied and expanded during the operation. Value is not used if the operation is Expand. **Note:** If using Expand, an additional attribute called Multiplier will need to be manually added in the XML Source pane. Multiplier will hold a comma separated list of integer multiplier factors for the first middle and last elements in the array being expanded, The syntax is MULTIPLIER="X,Y,Z" where X, Y and Z represent the multiple factors for the first middle and last elements. For example, if the source array contains the element values of (1,2,3,4) and the Multipliers are 2,3,2 then the target array becomes (1,1,2,2,2,3,3,3,4,4) .

- **Insert:** A value will be placed into the array at a particular location as indicated by Index. Insert results in a new array equal to the array indicated by the **Source Array** property with an element value placed at the location specified by the index property. **Index** must be populated with the value of the location of the element. The Source Array property must be populated with the name of the array to be altered.
- **Fill-By Deposit:** Fills the array with all deposits for funds on a policy. Fill-By Deposit should be the preferred operation as opposed to Fill-By SQL. The **Value** property must be populated with FundGUID.
- **Fill-By Fund:** Fills the array with Fund GUIDs from a policy where the Funds currently have positive value. Fill-By Fund should be the preferred operation as opposed to Fill-By SQL.
- **Fill-By AllocationFrom** Fills the array with all FundGUIDs from the activity allocation records in memory that have a negative allocation value . If there are no allocations with negative values, the allocation array will be empty with a length of "0" and no error will result.
- **Fill-By AllocationTo** Fills the array with all FundGUIDs from the activity allocation records in memory that have a positive allocation value. If there are no allocations with positive values, the allocation array will be empty with a length of "0" and no error will result.

- **Fill-By List:** Manually fills an array with a list of values. These values can be hard-coded (For example: 20) or reside in a previously defined Math variable (For example: "Twenty"). The **Value** property must be populated with the manual listing of items/values. The values within the listing can be previously defined math variables as well as hard-coded values, or a combination of both and the data type of the items must match the data type of the array.
- **Fill-By SQL:** Fills the array with the value of a SELECT query statement. The SQL query statement used must return only one single column and the data type of the value must match the data type of the array. The **SQL Query** property must be populated with the SELECT statement.
- **Fill-By Stored Procedure:** Fills the array with values from the Stored Procedure execution. You must name the procedure in the **Stored Procedure** drop down selection box. The box is populated with a current list of Stored Procedures. **Input** property is required. Defines the input parameter Name, Data Type, and allows for population/configuration of Values (previously defined Math variables).
- **Remove:** Takes a value from a specific element within the array using index to provide the location of the element. The **Index** must be specified by using the location of the element in the property. The **Source Array** name must be populated as the array you are altering with the removal operation.
- **Stride:** Creates a new array containing every *n*th element of SourceArray where *n* is the element value. The **Source Array** property must be defined with the name of the array you are altering. The **Value** property represents the element value defining *n*.
- **Transform:** Copies the array and uses the array for calculations. Copying the array using the Transform operation create a duplicate array with a different Variable Name. Calculations using Transform could include the addition of two arrays or multiplying all values in the array by a number. The **Value** property represents the expression (calculation being

performed) or an array name (copy the array then perform the calculation).

Check-in the transaction to save the information to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Math Pane > IIF

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

IIF

The IIF math variable is used as an IF-Else statement for evaluation and assignment of a new math variable. If the expression in the EXPRESSION attribute condition is true, the True Result is assigned to the New Math Variable. If the expression in the EXPRESSION attribute condition is false, the False Result is assigned to the new math variable.

The EXPRESSION, TRUE RESULT, and FALSE RESULT may use hard-coded default values/literal values, previously defined math variables, or a combination of both.



IIF Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the IIF Math Variable

Drag and drop the IIF Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- The data types are Activity, Boolean, Date, Decimal, Integer, Map, Object, and Text. This is required.
- Make sure variable type is defined as IIF. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the options of **Yes** or **No**. Round is only enabled if you select the Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to round the result value. **Note:** When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- In the **Expression** property, enter the condition you would like to evaluate.
- In the **True Result**, enter the value you would like to assign to the math variable in the case the expression result is evaluated to True.
- In the **False Result**, enter the value you would like to assign the math variable in the case the expression result is evaluated to False.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > MathIF

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

MathIF

The MathIF math variable allows for a conditional comparison to be evaluated. If the Condition attribute statement is true, then all of the math variables are executed within the MathIF. If false, the entire section is ignored/skipped and the system proceeds to the next math variable after the MathIF.

The benefit of using the MathIF is to assign multiple/several math variables for each conditional result (True and /or False).



MathIF Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the MathIF Math Variable

Drag and drop the MathIF math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Make sure variable type is defined as MathIF. This is required.
- In the **Condition** property, enter the comparison statement you wish to evaluate. You may use hard-code literal values, previously defined math variables, or a combination of both.

Check-in the transaction to save the information to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Math Pane > LoopIndex

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

LoopIndex

The LoopIndex math variable provides a method of counting the number of iterations for a Math Loop. The index is a zero index base counter used and maintained by the system.



LoopIndex Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the LoopIndex Math Variable

Drag and drop the LoopIndex math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a **Name** in the Name field. This is required.
- The applicable **Data Type** selection is defined as: Activity, Boolean, Date, Decimal, **Integer**, Map, Object, or Text. The data type must be set to Integer. This is required.
- Make sure variable type is defined LoopIndex. This is required.
- The **Source Array** is the name of the Math Loop you wish to index associated with this math variable.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > ExitLoop

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

ExitLoop

The ExitLoop math variable provides a method of discontinuing the loop. For the ExitLoop math variable, you will provide an IF condition and if the condition is true, the system will exit the math loop.



ExitLoop Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the ExitLoop Math Variable

Drag and drop the ExitLoop math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- Make sure variable type is defined ExitLoop. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- The **IF** text box property allows you to write the comparison requirement/statement. If the comparison statement is True, the system will exit the loop. You may use previously defined Math variables, hard-coded values, a combination of both, and operators within the configuration of the condition.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > MathLoop

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

MathLoop

The MathLoop math variable provides a method of executing a set of instructions that are to be repeated. Each type of repetition is called an iteration. There are four types of Loops currently supported in OIPA: FOR, SEGMENT, OBJECT and VALUATION.

The **FOR** Loop allows the system to repeat instructions to loop for a specified number of iterations. The number of iterations can be specified by using a hard-coded integer value or a defined math variable. If a math variable is used the math variable can hold a calculated integer value.

The **SEGMENT** Loop allows the system to loop through segments listed on/for a specific policy/certificate. By default, the SEGMENTLOOP will loop through all segments, inactive as well as active, unless the math variable Condition is specified. The Condition attribute/property is typically used to instruct the system to only loop through active segments based on the value of a specified segment field (e.g. `CONDITION="FieldName = 'SegmentActiveCode' And TextValue = '01'"`).

The **OBJECT** Loop provides a method to loop through any OIPA database table. OBJECT and OBJECTFIELD math variables are used inside the OBJECTLOOP to set up the table to be looped through and then retrieve the values from the table.

The **VALUATION** Loop provides a method to loop through the values of Valuation records held in memory as the activity processes but before it is written to the database tables.

Note: The Valuation looping method is only available to the PostAssignmentValidateExpressions rule and is not expressed as an option inside the Math pane of the Rules Palette.



MathLoop Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the MathLoop Math Variable

Drag and drop the MathLoop math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- Make sure variable type is defined MathLoop. This is required.
- LoopType is required. The options in the drop down selection are FOR, SEGMENT, OBJECT or VALUATION loops must be manually configured in the XML Source pane for the transaction's attached PostAssignmentValidateExpressions rule.
 - FOR Loop Only: Iterations requires the number of times the system will loop through the configuration. The value of this property may be hard-coded with a value or designated as a previously defined Math variable of the Integer data type.
 - SEGMENT and OBJECT Loops only: Condition requires a comparison statement for evaluation. SEE OPERATORS LINK list for details to configure the condition. The Condition allows the system to filter keys to be returned within the array. For example, if you are retrieving values from Client, you may filter via a condition stating TypeCode = '02' to return values where the client is an individual.
 - SEGMENT Loop Only: Policy property will always be set to [Policy:PolicyGUID] always. This is required.
 - OBJECT Loop Only: Object Name allows you to select a radio button option of Object or Free Form. Selecting Object will provide you with a list of all database tables. Free Form allows you to enter the table name (No As prefix when typing the table name free form).
 - OBJECT Loop Only: Key is the property that defines the primary GUID / key of the database table. For example, ClientGUID is the Key for Client database table.
 - VALUATION Loop Only: This loop type will allow the math to loop though the values of the valuation record that the activity will write or has already written but has not committed to the database yet. These values are held in system memory as the

transaction processes. The loop can only be run in the PostAssignmentValidateExpressions rule and will not work in normal transaction Math. It can either loop through all the valuation records for the activity or just those records for a specific fund. All columns on the AsValuation and AsUnitLinkedCalcValue tables are available by using a math variable with a VALUATIONFIELD type inside the loop to retrieve the column value.

Math Loop with Activity and ActivityArray: In order to populate the array with more than one Activity math object, a standard loop can be used (optional).

- Steps include: Create the ActivityArray, Begin the MathLoop, Create the Activity Object, Set Value of the Activity Object, Add the object to the ActivityArray, and Close the MathLoop.

Note: The ActivityArray variable must be instantiated outside of the loop and the Activity object math object inside the loop.

Check-in the transaction to save the information to the database.

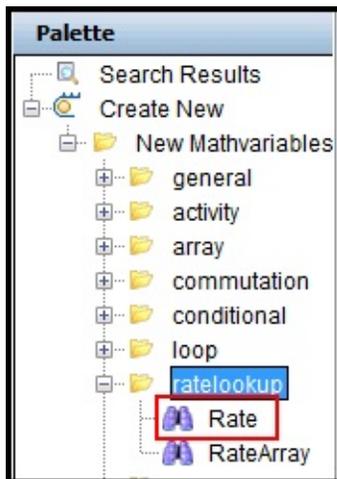
You are here: Configuration > Configuration Panes > Math Pane > Rate

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Rate

The RATE math variable is used to retrieve a single rate value of a DECIMAL data type. RATE can be used in screen math, transaction math and PostAssignmentValidateExpressions math.

The Rate Lookup folder in the Palette window contains the RATE math variable type.



RATE Option in Palette Window

Define RATE Math Variable

Drag and drop the RATE math variable type from the Palette window onto the configuration area of the Math Pane. Enter the math variable information.

- Name: type the name of the math variable.
- Data type: select the DECIMAL datatype.
- Variable type: RATE variable type is pre-selected..
- Rate Description: enter a math variable valued with the desired rate's RateDescription.
- Effective Date: enter a math variable valued with a date representing a legal effective date for an entity such as the policy or segment.
- Active Date: enter a math variable name valued with a date representing the processing date such as the system date or activity's effective date.
- Default: enter a value that becomes the math variable's value if no rate is retrieved.
- Click **Add** to specify criteria for the rate. Criteria options are populated from the business names for the criteria for the rate description entered earlier. The available math variables are populated from the math variables prior to the current math variable.
- To remove a criteria, check the box to the right of a criteria and then click the **Remove** button.
- Check-in the file to save the changes.

You are here: Configuration > Configuration Panes > Math Pane > RateArray

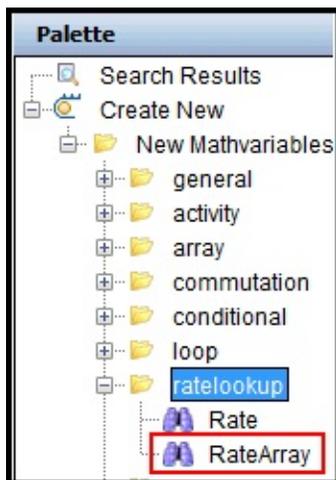
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

RateArray

The RATEARRAY math variable is used to retrieve multiple rate values into a NUMERICARRAY data type. The resulting array may be 0 to 'n' elements in size.

RATEARRAY can be used in screen math, transaction math and PostAssignmentValidateExpressions math.

The Rate Lookup folder in the Palette window contains the RATEARRAY math variable.



RATEARRAY Option in Palette Window

Define RATEARRAY Math Variable

Drag and drop the RATEARRAY math variable from the Palette window onto the configuration area of the Math Pane. Enter the math variable information.

- Name: type the name of the math variable.
- Data type: select the DECIMAL datatype.
- Variable type: this should be RATEARRAY.
- Start Index: This is the starting integer value returned to the array. The array will contain rates whose IntegerCriteria are equal to and greater than this value.
- End Index: This is the ending integer value returned to the array. The array will contain rates whose IntegerCriteria are equal to and less than this value.
- Rate Description: enter a math variable valued with the desired rate's RateDescription.
- Effective Date: enter a math variable valued with a date representing a legal effective date for an entity such as the policy or segment.
- Active Date: : enter a math variable name valued with a date representing the processing date such as the system date or activity's effective date.
- Click **Add** to specify criteria for the rate from the business names for the criteria for the rate description enter earlier. The available math variables are populated from the math variables prior to the current math variable.
- To remove a criteria, check the box to the right of a criteria and then click the **Remove** button.
- Check-in the file to save the changes.

You are here: Configuration > Configuration Panes > Math Pane > CopyBook

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

CopyBook

The CopyBook math variable is used to reference a CopyBook business rule in order to execute reusable configuration within the Math. A CopyBook business rule allows for plans, transactions, and other rules to share common functionality. The content of the CopyBook is resolved and inserted inside the transaction at the point the Math variable designates the call.



CopyBook Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the CopyBook Math Variable

Drag and drop the CopyBook math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Make sure variable type is defined as CopyBook. This is required.
- The **CopyBook** selection drop down box contains all configured CopyBooks. Select the applicable CopyBook you wish to use / assign the Math variable.

Check-in the transaction to save the information to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Math Pane > FunCall

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

FunCall

The FunCall math variable allows the system to execute a Function business rule inside the Math. A Function is a rule that calls a calculation routine and houses discrete and logical piece of a calculation. Functions are often selected over CopyBooks for specific calculations because they offer increased readability as the parameters are clearly defined. The Function contains input and output parameters in addition to one return value. The output and return value may be utilized by other math variables and calculations once the Function is executed.



FunCall Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the FunCall Math Variable

Drag and drop the FunCall math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- The **Data Type** must reflect the corresponding data type of the Return value you are assigning to this FunCall math variable. The applicable data type selection is defined as: Activity, Boolean, Date, Decimal, Integer, Map, Object, or Text. This is required.
- Make sure variable type is defined as FunCall. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the options of **Yes** or **No**. Round is only enabled if you select the Integer or Decimal in the data type property. If you would like to round the Decimal or Integer, select **Yes**. Enter the number of decimal places to round the result value. **Note:** When the data type of Decimal or Integer is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- The **Functions** selection drop down box provides a listing of all configured Function business rules. Select the applicable Function name.
- **Input** parameters are required to feed or pass into the FunCall. The Parameter Name column has a listing of p Parameter types that must be defined prior to calling the FunCall. The p ParameterName is pre-populated with the Parameter column. The corresponding Data Type column is pre-populated. You must designate a previously defined Activity Field or Math variable in the Value column. Only those fields and math variables with the corresponding data type will appear in the drop down selection box. The DataType column has a corresponding list of each data type for the input parameter names listed. By placing the cursor inside the Value column drop down selection boxes, you will be presented with previously defined Math variables with that corresponding data type.
- **Output** parameters are optional depending how the Function

business rule was configured. If the Output Parameters are pre-populated within the Parameter Name and Data Type columns, a Value is Required. The Parameter Name has a list of oOutput Parameter Names and the corresponding Data Type pre-populated in the Data Type column. By placing the cursor inside the Value column drop down selection box, you will be presented with the list of previously defined Math variables with the corresponding Data Type.

Check-in the transaction to save the information to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Math Pane > Function

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Function

The Function math variable calls a pre-defined system function and returns a single value. The functions allow for manipulation of values for all data types. The Function math variable is commonly used for the manipulation of dates. For example, adding lead days to the billing effective date to ensure the billing notice arrives prior to the due date. Another example is to add time on to a date, such as using the DaysAdd function to increment a date value.



Function Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the Function Math Variable

Drag and drop the Function math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a **Name** in the Name field. This is required.
- The **Data Type** for the value of the math variable. This data type corresponds with the result or outcome of the Function Call. The applicable data type selection is defined as: Activity, Boolean, Date, Decimal, Integer, Map, Object, or Text. This is required.
- Make sure variable type is defined as Function. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- **Round** offers the options of **Yes** or **No**. Round is only enabled if you select the Decimal in the data type property. If you would like to round the Decimal, select **Yes**. Enter the number of decimal places to round the result value. **Note:** When the data type of Decimal is selected, the default value for Round is set to No. You must change the selection of the radio button to Yes if you wish to round.
- The **Call** requires the Function Name and parameters. For example, DaysAdd(IssueDate, BillingLeadDays). The parameters may be a previously defined math variable or a combination of a previously defined math variable and a hard-coded value. For example, MonthsAdd(EffectiveDate, 1).

Check-in the transaction to save the information to the database.

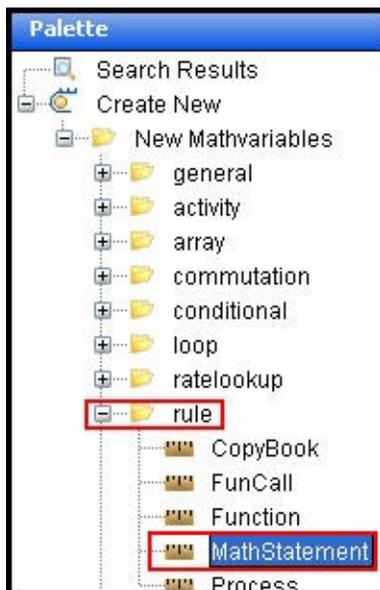
You are here: Configuration > Configuration Panes > Math Pane > Math Statement

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

MathStatement

The MathStatement math variable is a type of system function that is similar to the existing type **FunctionCall**. Unlike business rule functions, functions called by the MathStatement exist in the base code; however, the Rules Palette guides the configurator through the available functions and the required parameters for each function.

MathStatement will not appear in the math tab of the Activity Results screen. Valuation objects created by MathStatements will not write valuation records to the data base.



MathStatement Math Variable in Palette Window

Four functions and their parameters are currently supported:

- **FindNextExchangeDatePricing**: Use when the activity is to use the next available currency exchange date on or after the activity effective date and fund unit values available on or after the calculated exchange date.
- **FindLastExchangeDatePricing**: Use when the activity is to use the last known currency exchange date and fund prices on or prior to the activity effective date.
- **GetBenefitSplit**: This function reads parameters to calculate the

benefit split record for the activity calculation.

- **MathValuation:** This will execute the policy valuation as of a specified date for use inside transaction math.
- **GetFundPositions:** This function will retrieve a collection of parent funds that have fund balances, as well as the values of these parent funds. This can only be used in transaction Math when the activity is configured to perform valuation.

Define the MathStatement

Drag and drop the MathStatement function from the Palette window onto the Math Pane. Enter the following information for the MathStatement.

- Select a variable type from the Variable Type drop down list. field.
- Click the Functions field for a list of the supported types of functions. Select one of the supported functions.
- Enter the Input or Output values in the Values column for each parameter as needed. Input parameters are required. Output parameters are optional.

Check-in the transaction to save the information to the database.

FindNextExchangeDatePricing

Uses exchange date offset as the input. The output is a guaranteed exchange date (exchange date the system will use) and the gain/loss exchange date (current system date) as well as a boolean result that identifies whether rates exist in the Rates table for the guaranteed exchange date. If rates do not exist, the activity cannot process.

Below are the configuration parameters for FindNextExchangeDatePricing:

- Variable Type: ACTIVITYFUNCTION
- Input
 - ExchangeDateOffset (INTEGER). This is required.
- Output
 - GuaranteedExchangeDate (DATE). This is optional.
 - GainLossExchangeDate (DATE). This is optional.
 - RatesExist (BOOLEAN). This is optional.

The parameters GuaranteedExchangeDate and GainLossExchangeDate are passed into the Assignment. Assignment simply checks that MathStatementActivityFunction:GuaranteedExchangeDate is logged into

AsActivityMath. These parameters are needed for unit linked funds to the help system determine exchange dates. Mapping the output parameters to math variables is optional as the system will automatically log the values. The output parameters should be configured only if the values are to be used elsewhere in the activity math.

XML Example

```
<MathStatement TYPE="ACTIVITYFUNCTION" FUNCTIONNAME="F.  
  <Parameters>  
    <Parameter NAME="ExchangeDateOffset" TYPE="INPUT">I  
    <Parameter NAME="GuaranteedExchangeDate" TYPE="OUTP  
    <Parameter NAME="GainLossExchangeDate" TYPE="OUTPUT  
    <Parameter NAME="RatesExist" TYPE="OUTPUT">RatesExi  
  </Parameters>  
</MathStatement>
```

FindLastExchangeDatePricing

Below are the configuration parameters for FindLastExchangeDatePricing:

- Variable Type: ACTIVITYFUNCTION
- Input
 - none
- Output
 - GuaranteedExchangeDate (DATE). This is optional.
 - GainLossExchangeDate (DATE). This is optional.

XML Example

```
<MathStatement TYPE="ACTIVITYFUNCTION" FUNCTIONNAME="F.  
    <ParameterNAME="GuaranteedExchangeDate" TYPE="OU  
<Parameter  
    NAME="GainLossExchangeDate" TYPE="OUTPUT">GainLossExchangeD  
    </Parameters>  
</MathStatement>
```

GetBenefitSplit

This function reads parameters to calculate the benefit split record for the activity calculation. Below are the configuration parameters for the function.

- Variable Type: MathStatement
- Input:
 - SegmentGUID: a math variable holding the related segment for the Benefit Split record.
 - EffectiveDate: a math variable holding the valuation effective date for AsNetAssetValue records.
 - BenefitSplitTypeCode: a math variable or literal value indicating the Benefit Split type code to retrieve.
 - ABLIndicator: a math variable holding either Yes or No indicates if ABL is in effect and if the variable benefit is to be suppressed.
 - ExcludeFunds: a math variable holding an array of one or more benefit funds to exclude from the calculated output.
- Output: (optionally configured to map to math variables for use elsewhere in transaction processing)
 - BenefitAmount: gross benefit amount in the Plan default currency.
 - ParentFundAmountCollection: key/value pairs of Parent Fund GUID and Benefit Allocation Amounts in decimal values.

XML Example

```

<MathStatement TYPE="ACTIVITYFUNCTION" FUNCTIONNAME="GetE
<Parameters>
<Parameter NAME="SegmentGUID" TYPE="INPUT">SegmentGUID</F
<Parameter NAME="Effectivedate" TYPE="INPUT">Activity:Eff
<Parameter NAME="BenefitSplitTypeCode" TYPE="INPUT">TypeC
<Parameter NAME="ABLIndicator" TYPE="INPUT">ABLActive</Pa
<Parameter NAME="BenefitAmount" TYPE="OUTPUT">OutputAmour
<Parameter NAME="ParentFundAmountCollection" TYPE="OUTPUT
<Parameters>
</MathStatement>

```

MathValuation

This executes the policy valuation as of a specified date for use inside transaction math. The valuation date specified in the Activity valuation is not read by OIPA and only the activity's effective date is applied to the activity valuation. The new valuation object created in math holds all of the same data that the activity valuation contains. Retrieving data from the math variable valuation object uses a different syntax but all of the same data is available (Fund, Deposit or Policy values). Once created, a math valuation object remains in memory until it is replaced by another math valuation object. Only one math valuation object exists in memory at the same time as activity valuation. Activity valuation is calculated independently of the math valuation and the two co-exist inside the transaction processing and can be independently accessed.

The presence of the activity <Valuation> element is not a prerequisite to calculating valuation in transaction math. Calculation of the math valuation is not available to screen math or math in attached rules. A stack trace is thrown if configured in an invalid location. All valuation is executed using the plan configuration (point in time or from inception).

Below are the configuration parameters for MathValuation:

- Variable Type: ACTIVITYFUNCTION
- Input:
 - Valuation Date (DATE): holds a field or math variable containing a date value.

- PolicyValues (TEXT): holds a literal or a field or math variable containing a “Yes” or “No” text string. This indicates whether the PolicyValues business rule is executed as part of the math valuation.
- Nearest Price (TEXT): holds a literal or a field or math variable containing a “Yes” or “No” text string. This indicates that valuation should be calculated with the exact prices as of the ValuationDate (No) or with the last known prices as of the ValuationDate (Yes).
- Output:
 - ExactPriceDateUsed (TEXT): returns a Boolean (true/false) indicator if the fund prices used to value the policy were as of the exact ValuationDates specified (true) or if the nearest price dates were used (false). Mapping this output value to a configured math variable is optional. The NearestPrice must be set to Yes in order to make use of this output. If NearestPrice is set to No and the activity processes, then the output value will always be True.

XML Example

```
<MathStatement TYPE="ACTIVITYFUNCTION" FUNCTIONNAME="MathVali
  <Parameters>
    <Parameter NAME="ValuationDate" TYPE="INPUT">MyValuation[
    <Parameter NAME="PolicyValues" TYPE="INPUT">Yes</Paramete
    <Parameter NAME="NearestPrice" TYPE="INPUT">Yes</Paramete
    <Parameter NAME="ExactPriceDateUsed" TYPE="OUTPUT">ExactF
  </Parameters>
</MathStatement>
```

GetFundPositions

This function will retrieve a collection of parent funds that have fund balances, as well as the values of these parent funds. GetFundPositions

can only be used in transaction math if the transaction is configured to perform valuation. If this function is added to a transaction that is not configured to perform validation, a system error will occur in OIPA, which should indicate that the configuration was not set up properly.

Below are the configuration parameters for GetFundPositions:

- Variable Type: MATHSTATEMENT
- Input:
 - ExcludeFundTypes: A math variable holding an array of one or more fund types to exclude from the calculated output.
- Output:
 - ParentFundAmountCollection: Key/value pairs of Parent Fund GUID and Fund Amounts in a currency object

 This collection may be further modified by the transaction math for use in the ReassignAllocations rule.

XML Example

```
<MathStatement TYPE="ACTIVITYFUNCTION" FUNCTIONNAME="GetFundPositions">
  <Parameters>
    <Parameter NAME="ExcludeFunds" TYPE="INPUT">FundTypeArray
    <Parameter NAME="ParentFundAmountCollection" TYPE="OUTPUT">ParentFundAmountCollection
  </Parameters>
</MathStatement>
```

You are here: Configuration > Configuration Panes > Math Pane > Process

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Process

The Process math variable is used to call or execute a custom java class extension from within the Math. Custom configuration called during Math execution using extension and Process math variables allows the system to perform custom calculations/math and return the results to OIPA. You may also pass the math variables as parameters to other java classes.



Process Math Variable in Palette



For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the Process Math Variable

Drag and drop the Process math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the **Name** field. This is required.
- The applicable **Data Type** selection is defined as: Activity, Boolean, Date, Decimal, Integer, Map, Object, or Text. This is required.
- Make sure variable type is defined as Process. This is required.
- The **Object** is the name of the class name of the extension. For example, TestExtension.
- The **NameSpace** is the package the class resides in. For example, com.oracle.oipa.ext.
- **Input** provides a list for Parameter Name as required input. Value is a drop down selection box that contains previously defined Activity Fields and Math variables.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > Collection

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Collection

The Collection math variable is used to hold a grouping of key and value pairs of data. The key-value pairings can be created by executing a SQL statement that will return two result columns as a key-value pair.

Alternately, a Collection may be built using a SetValue operation where the key is defined as an attribute of the math variable to attach to its value. The math variable CollectionValue is used to retrieve its associated value.



Collection Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the Collection Math Variable

Drag and drop the Collection math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- Make sure variable type is defined as Collection. This is required.
- The data type selection is: Boolean, Date, Decimal, Integer, Map, and Text. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- The operation is defined as **Create**, **FillBy-SQL** or **SetValue**.
 - Create will initialize the collection and prepare it to be populated by SetValue operations in subsequent math variables. FillBy-SQL does not require that the collection first be created. When using Create the data type selection should be either Map or Currency.
 - The SetValue operation is a way to build the collection after it is created using singly added values statement by statement instead of using a SQL query. When using SetValue, the data type must match that of the value being set.
 - FillBy-SQL populates the collection using a SELECT statement that returns two columns; key and its value. This is required.

Note: If using FillBy-SQL, then a third column must be added if the collection is of currency values. This third column will hold the Currency Code. The datatype in this case must be Currency otherwise the data type must be Map.

- Type the Key to be used to map the value to be inserted into the collection when the operation is SetValue. The key could be a math variable or a field or a literal value but it must be unique to the value being inserted. This is required when the operation is SetValue.

Check-in the transaction to save the information to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Math Pane > CollectionValue

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

CollectionValue

The CollectionValue math variable is used to retrieve a value from a collection based on its key. A math variable statement of type CollectionValue returns the value associated to a given key. The Collection math variable must be defined/created prior to using CollectionValue.



CollectionValue Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the CollectionValue Math Variable

Drag and drop the CollectionValue math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the Name field. This is required.
- The applicable data type selection is defined as: Boolean, Date, Decimal, Integer, or Text. This is required.
- Make sure variable type is defined as CollectionValue. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. If Yes is selected, the number of decimal places is enabled. The radio button values are disabled unless you select the data type Decimal. This is an optional field.
- The value of the math variable is the math variable name of the previously defined Collection math variable.
- The **Key** is the literal name of the key of the value in the Collection. KeyField is a math variable or field that holds the value of the key of the value in the Collection.
- Type in a **Default** value for an Integer or Date data type. For example, an Integer default value could be -999999999. Required so the system does not return a null value.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > SQL

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

SQL

The SQL math variable is used to retrieve data results using a SELECT statement. SQL is used to retrieve a value from the database using a query statement. The SQL statement should be constructed to return one value.

SQL Math Variable in Palette

Note: For a comprehensive explanation of the characteristics of this math variable, refer to the XML Configuration Guide in the Help menu.

Define the SQLMath Variable

Drag and drop the SQL Math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a name in the **Name** field. This is required.
- The applicable **Data Types** are: Boolean, Date, Decimal, Integer and Text. This is required.
- Make sure variable type is defined as SQL. This is required.
- Select **Yes** or **No** from **Log** field. Default is No. This is an optional field.
- The **Default** is left blank and is an optional field. Required for Date and Integer data types so the system does not return a null value. For example, an integer default could be set to -999999999. Numeric and Date data types must consider a default whereas, text data type is more flexible and will not require a default value.
- Enter the **SQLQuery** / SQL query statement. This is required.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Math Pane > Math Update

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Math Update

MathUpdate is a unique operation that allows a math variable to update a policy or segment field value held in memory when an activity processes. It resides inside transaction math much like MathIF and MathLoop. This math variable is similar to the CopyToPolicyFields or CopyToSegmentFields attached rules, except that it does not actually update database values.

The MathUpdate element in transaction math indicates an activity that processes projections. The actual results of projection processing are maintained in memory only, and no database contents are modified.

MathUpdate leverages the input context for activity processing. This input context is also used by CopyTo rules to retrieve the original values of fields. A CopyTo will not create an update for the field if the newValue and the oldValue are equal.

If MathUpdate updates a field which is also a target in a CopyTo, MathUpdate needs to set the original value in the input context before math processing completes.



It is **not recommended** to use MathUpdate on transactions that have attached business rules. Using MathUpdate with attached business rules can affect data persisted in the database.



Math Update Math Variable in Palette

Define the Math Update Math Variable

Drag and drop the Math Update math variable from the Palette window onto the Math Pane. Enter the following information for the math variable.

- Type a field name in the **FieldName** field. This is the name of the table Field to update in memory. This is required.
- Select a Type from the **Type** drop down box. The type identifies the source table of the record type to update. The available options are: POLICYFIELD, SEGMENTFIELD or SEGMENTFIELDCOLLECTION. SegmentFieldCollection updates the named Segment field records with the collection values. This is required.
- Type the Segment field name in the **Segment** field. This field is only available if Type=SEGMENTFIELD. The Segment field name appears in the XML as the value of the SEGMENTGUID attribute.
- Type the name of the math variable that will be used to update the field name defined in the **FieldName** box above. This is required. This will be an auto complete field consisting of all the configured math variables in the transaction defined before the MathUpdate plus empty for an initial value. The field will be saved in the XML as the value of the MathUpdate element.

Note: The Debugger does not display the values of MathUpdate variables or the field itself.

Check-in the transaction to save the information to the database.

You are here: Configuration > Configuration Panes > Assignment Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Assignment Pane

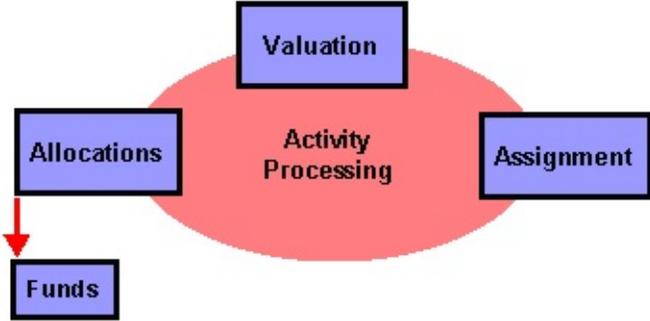
The Assignment pane allows for visual configuration of assignments that are made to a policy through transaction processing. The Assignment element, located in the transaction XML, describes how values should be assigned to the policy and written to the valuation database tables for the [Valuation](#) process. AsValuation is the table that holds all of a policy's records that describe how each activity either moved money in or out. Assignment uses allocations and writes a record for each allocation by assignment and money type.

Assignment allows money to be deposited, withdrawn or transferred from the funds. [Allocation](#) provides a request as to how to affect each fund, such as a deposit, withdrawal or transfer. [Valuation](#) is the result after money movements are completed.

The Rules Palette offers a variety of ways to assign money-in and money-out. Depending on the assignment chosen, the Rules Palette will require certain input and the Oracle Insurance Policy Administration system will process money in or out in a certain manner. For a complete list of assignment types, see the XML Configuration Guide in this help system. Assignment information can be found under **Transaction Rule | Transaction Elements | Assignment Elements**.

An assignment must have a money type associated with it, since money can be applied to a policy from different sources and it must be tracked for auditing and tax purposes. Money types are stored, edited and updated via the AsCodeMoneyType table in the [Codes Names](#) folder in the Admin Explorer window. Each money type is assigned a code value that is included in the assignment, which also allows for it to process and display the money values correctly.

Valuation Associations



Valuation Associations

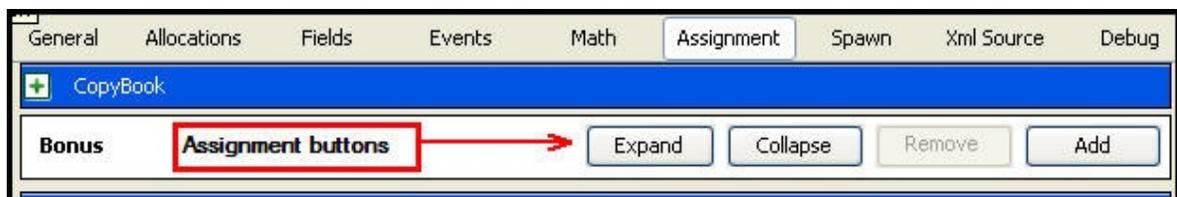
AsValuation Record Illustration

ValuationGUID	FundGUID	PolicyGUID	ActivityGUID	EffectiveDate	MoneyTypeCode	ValuationAmount	ValuationUnits
F1334192-8B69-4E17E5CFDC-03DAC36CCE-D0B8AE4D1D-CE141	29/2031	12:00:00	AM	01	100000	8354.2189	

AsValuation record from database

There are four buttons on the Assignment pane.

- **Expand:** shows the attribute and money type node under each assignment.
- **Collapse:** hides the attribute and money type node under each assignment.
- **Remove:** removes assignments, attributes or money types.
- **Add:** adds assignment, attributes or money types.



Assignment Pane and Assignment Buttons

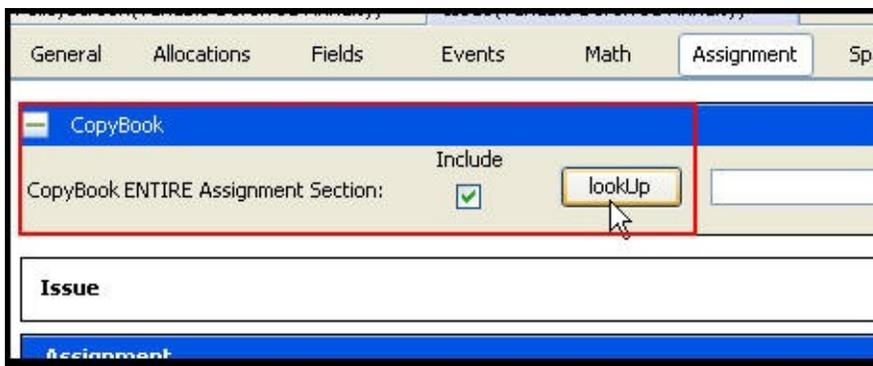
Insert [an entire Assignment section from a CopyBook](#) or configure [the Assignment section through the Assignment pane](#).



Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Transaction Rules | Transaction Elements | Assignment Elements**.

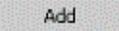
Insert Entire Assignment Section from CopyBook

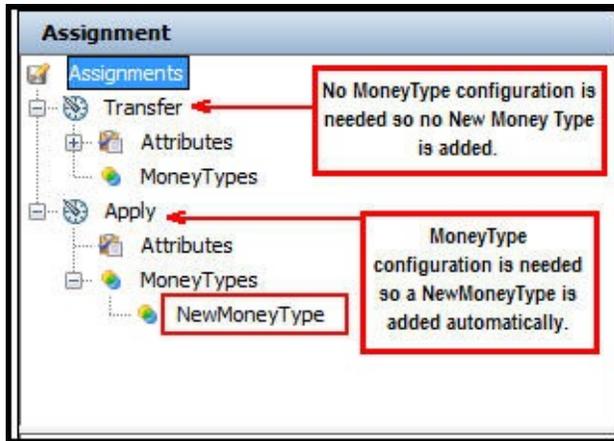
1. Navigate to the transaction and click the Assignment pane.
2. Expand the CopyBook section at the top of the pane.
3. Click the Include box.
4. Click **lookUp** and select the CopyBook from the CopyBook Lookup window.
5. Click **OK**. The CopyBook assignment information will be added to this pane.



CopyBook Section on Assignment Pane

Steps to Configure an Assignment

1. Navigate to the transaction or [create a new transaction](#).
2. Check-out the transaction's XML file.
3. Open the Assignment pane.
4. Select  **Assignments**.
5. Right-click and select **Add** or select the **Add**  button. A new assignment will be added with two nodes: attributes and money types. If the assignment requires a money type configuration element, then it will be automatically added.



Assignment Pane of a Transaction with Two Types Added

6. Select an assignment from the **Assignment type** drop-down box.
7. Right-click on the attributes node under the new assignment and select **Add**. A new attribute will appear.
8. Click the new attribute. The bottom portion of the screen will display the attribute options that can be configured.
9. Configure the attribute.
10. Right-click on the money type node under the new assignment and select **Add**.
11. Select a money type from the Name drop down box on the bottom section of the screen.
12. Select a fund if applicable.
13. Select a money type from the drop-down box.
14. Click **Save** to save the changes to the database.
15. **Check-in** the file once all necessary assignments have been added.

You are here: Configuration > Configuration Panes > Sequences Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Sequences Pane

The Sequences pane is a specialized pane available in the Transaction Editor for [Intake-Record transactions](#). This pane is used to visually configure Activity Sequences—ordered, dynamic collections of activities used to execute changes to member records when such records are being altered by information processed through Data Intake. Sequenced activities may be created for companies, group customers, clients, plans or policies depending on the business need.

Sequences Pane Sections

Sequences

The Sequences table allows for the addition of Activity Sequences by using the **Add** button. The position of each Sequence can be altered by using the **Up** and **Down** buttons. Sequences will execute in the order in which they are displayed in the Sequences table. Sequences can be deleted by using the **Remove** button.

Sequence Condition

When an Activity Sequence is selected from the Sequences table, a conditional expression, if defined, will display in the Sequence Condition field. This expression is what dictates whether the Sequence will execute—if true, the Sequence will execute; if false, the Sequence will not execute. If an expression is not yet defined for the Sequence, one can be entered in this field.

Sequence Activities

When an Activity Sequence is selected from the Sequences table, all of the activities/ActivityArrays contained within that Sequence will display in the Sequence Activities table.

Activities/ActivityArrays can be added to the Sequence by using the **Add** button in this section of the pane. Doing so will insert a blank row into the Sequence Activities table. Using auto-complete, the Rules Palette will present a list of configured activities and ActivityArray math objects for insertion into the Sequence. If the **Process Immediate** checkbox is checked, the activity/ActivityArray will be processed immediately.

The position of each activity/ActivityArray can be altered by using the **Up** and **Down** buttons. Activities/ActivityArrays will execute within the

Sequence in the order in which they are displayed in the Sequence Activities table. Activities/ActivityArrays can be deleted by using the **Remove** button.

Activity Condition

When an activity/ActivityArray is selected from the Sequence Activities table, a conditional expression, if defined, will display in the Activity Condition field. This expression is what dictates whether the activity/ActivityArray will execute—if true, the activity/ActivityArray will execute; if false, the activity/ActivityArray will not execute. If an expression is not yet defined for the activity/ActivityArray, one can be entered in this field.

You are here: Configuration > Configuration Panes > Spawn Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Spawn Pane

Spawn is the term used to describe an activity that was system generated as a result of another activity being processed. A transaction can be configured to spawn other transactions through the Spawn pane. The configuration can be customized for each transaction or Spawn sections can be copybooked and used across multiple transactions.

If the spawn section of a transaction is configured in a copybook, the copybook should be at the plan level because transactions reside at the plan level.

If an activity that spawned other activities is deleted or recycled, then the spawned activities will be deleted or recycled as well, unless the transaction is of the type [Nonreversible-Nonreversing](#). A Nonreversible-Nonreversing transaction cannot be deleted or recycled by another activity, even if the transaction that spawned it is reversed or recycled. The Nonreversible-Nonreversing transaction can be deleted or recycled if a user has the proper security access, which will cause all of the spawns to be deleted as well.



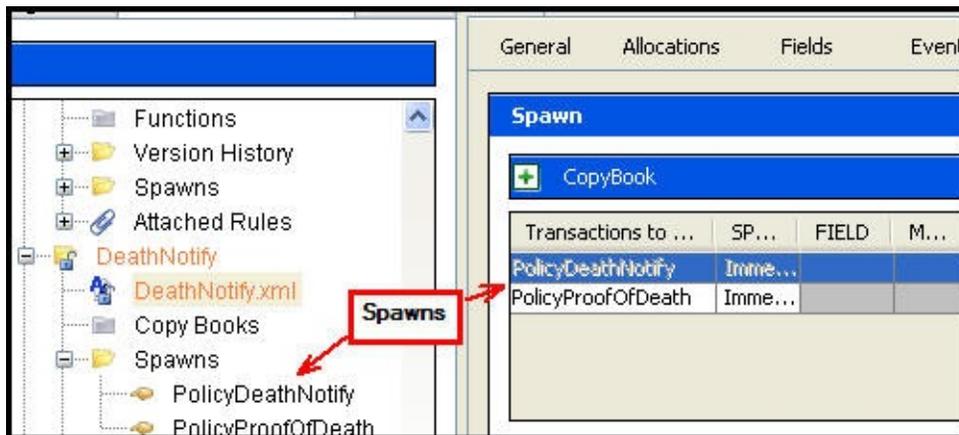
Spawning using Nonreversible-Nonreversing transactions should be extremely limited as spawned transactions could exist on the policy when the original parent transaction no longer exists.

Deleting Spawns

A spawned activity should never be deleted or recycled in a production environment. Best practice is to delete or recycle the originating parent activity, which will in turn delete/recycle the spawn. This prevents activity data from becoming corrupt.

In the rare cases when a spawn needs to be deleted in a development environment, a Delete checkbox is provided on the Activity screen. Users with the proper security privileges will be able to check that box and gain access to the delete and recycle icons next to spawned activities.

A Delete checkbox can be added on the Policy, Client, Plan and Company Activity screens in OIPA. The security for this button is configured at the [PolicyActivity](#), [ClientActivity](#), [PlanActivity](#) and/or [CompanyActivity](#) levels.

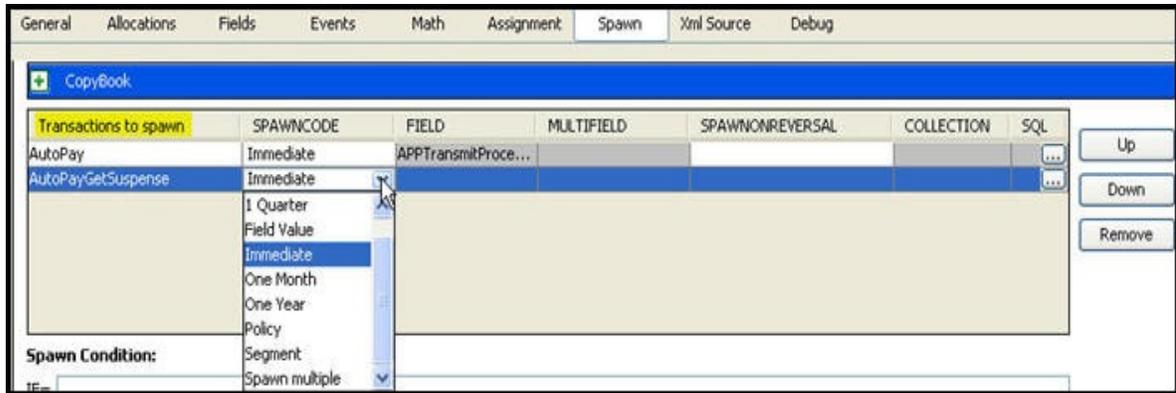


Transaction with Spawn

Sections of Spawn Configuration Pane

1. The first section on the Spawn pane is the Copybook section. Click the green plus sign to expand this section. This is used when an entire Spawn section from a different transaction needs to be copied and used in the transaction you are configuring. If the **Include** checkbox is checked then a Lookup field will appear so that the transaction with the Spawn section you want to copy can be selected. All other fields on the Spawn pane will be disabled if a copybook is used. Refer to the steps below for instructions on [adding a spawn copybook](#).
2. Directly below the Copybook section is the first gray table, which holds a list of spawns the transaction will trigger. The **Transaction to spawn** column displays a list of transactions that will be spawned. Highlight a transaction in the list and double click in a cell to edit the information. Refer to the steps below for instructions on [adding transactions to this table](#).

The SPAWNCODE indicates when the transaction should be spawned. A list of SpawnCodes from AsCodeSpawn in the AsCode table are listed below. The SpawnCode selected will determine if additional columns in the table are available. For example, if the spawn's occurrence is based on a date field then select the field value for SPAWNCODE and the FIELD combo box will be enabled so that the correct variable can be chosen from the combo box. Double-click in a column to see if the option can be used with the type of SpawnCode selected.



Transactions to Spawn listed on Spawn Pane

Code Value	Description
01	Immediate: spawn on the effective date.
02	One Year: spawn a year from the effective date.
03	Field Value: spawn on a date defined from a field in the activity.
04	1 Business Day: spawn on the next business day.
05	One Month: spawn a month from the effective date.
06	Segment: spawn one for each segment.
07	1 Quarter: spawn one quarter from the effective date.
08	Spawn multiple: spawn an activity for each Pattern Field (Field0, Field1, etc.).
09	Policy: spawn an activity for each policy in the plan.
10	SpawnMultiFields: This spawn code is used with transactions that have MultiFields. On selecting this spawn code, the MultiFields column on the screen is enabled with two options: Yes and No. Yes- spawns the current activity's MultiFields to the spawned activity. No- MutiFields values will not be spawned. If this attribute is not present, No will be assumed.

- The Spawn Condition section gives the user the option to configure a condition when the transaction should be spawned. If no condition is given then the transaction will always be spawned.

Spawn Condition:

IF=

Spawn Condition Statement

- Spawn fields allows the user to indicate the variables from the processed activity that should be passed to the spawned transaction.

Select an option from a combo box in the **FROM** field, which is filled with all the fields and math variables available in the spawning transaction. If no selection is made in the **FROM** field, the field will not be spawned. In a majority of cases all **FROM** fields will be filled in. **FROM** fields allow date entry and are sorted by the datatype of the **TO** fields.

Spawn Fields:	
From	To
GrossAmountMV	GrossAmount (Money)
FrequencyMV	Frequency (Combo)
NextAPPTransmitDate	ActualAPPTransmitDate (Date)
EndDateMV	EndDate (Date)

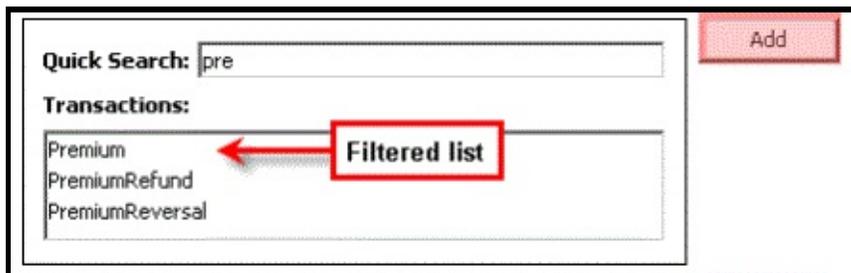
Spawn Fields Section on Spawn Pane

5. The bottom section is used to define suspense, if needed. Checking the box next to suspense or multisuspense will activate the attribute fields allowing additional parameters to be entered.
 - a. Suspense Attributes: Select a type from the drop down box and enter a value. The value must be at least the value of the Activity field; it cannot be less.
 - b. Multisuspense Attributes: Enter the name of a Collection type math variable in the Collection field. Then select a type and enter a maximum number of suspense records to include in the collection.

The screenshot shows the 'Suspense Attribute Section on Spawn Pane'. It contains two main sections: 'Suspense Attributes' and 'MultiSuspense Attributes'. The 'Suspense' checkbox is checked, and a red box highlights the instruction 'select one, but not both' with arrows pointing to both checkboxes. The 'Suspense Attributes' section has a 'Type' dropdown and a 'Value' input field. The 'MultiSuspense Attributes' section has a 'Collection' input field, a 'Type' dropdown, and a 'Max Records' input field. Below these are 'Allocation Type' and 'Allocation Value' dropdowns.

Suspense Attribute Section on Spawn Pane

6. The **Quick Search** section on the far right side of the pane is used to add a new spawn to the transaction. To add a spawn, type the transaction name in the **Quick Search** box. A filtered list reveals all available transactions in the Transaction box. Highlight the desired transaction and select the **Add** button. The transaction will be added to the spawn list. Use the **Up**, **Down** and **Remove** buttons to set the order of spawns or remove spawns from the list.



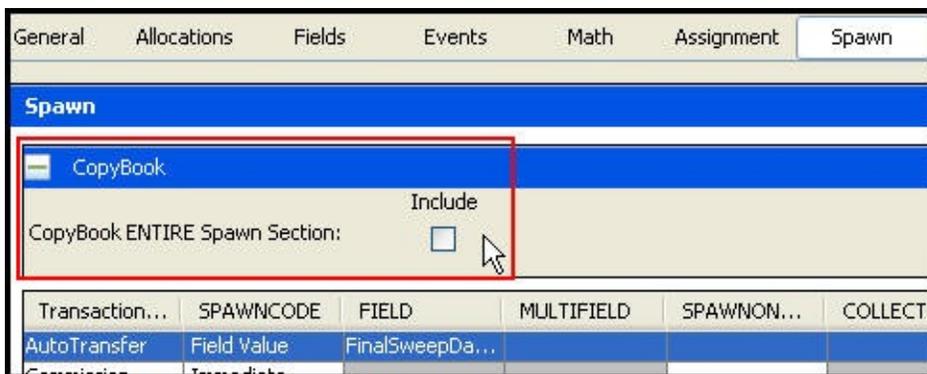
Quick Search Section on Spawn Pane

An activity is an instance of a transaction. When an activity is selected, the transaction configuration is executed.

Insert [an entire Spawn section from a CopyBook](#) or configure [the Spawn section through the Spawn pane](#).

Insert Entire Spawn Section from CopyBook

1. Navigate to the transaction and click the Spawn pane.
2. Expand the CopyBook section at the top of the pane.
3. Click the Include box.
4. Click **lookUp** and select the CopyBook from the CopyBook Lookup window.
5. Click **OK**. The CopyBook spawn information will be added to this pane.



CopyBook Section on the Spawn Pane

Steps to Create a Spawn

1. Select the **Spawn** pane.
2. Search for the transaction to spawn from the Quick Search box.
3. Select the transaction name from the Transactions list.
4. Select **Add**, which adds the transaction to be spawned.
5. Use the **Up** or **Down** button to move the spawn to the correct order. This is important if one activity relies on the output of another activity.
6. Select the SpawnCode from the drop-down box. The SpawnCode determines how or when the transaction should be spawned. Additional fields in the table will become available based on the SpawnCode selected. Double-click in a column to see if the option is available with the type of SpawnCode selected.
7. Set the Spawn Condition if the spawn should only occur when some condition result is true. If left blank, the spawn will always occur.
8. Select the spawn fields needed to complete the spawn. The To section automatically populates with the applicable math variables or fields for the activity the values are going to. The From section has drop-down boxes to select a field or math variable that stores the values needed to pass to the activity being spawned. The From value will be passed to the specified To field of the spawned transaction.
9. Click the suspense option needed. Only one can be selected.
10. Select the attributes needed for suspense from the fields provided.
11. Check-in the transaction to save the changes to the database.

Use the **Remove** button next to the first gray table to remove a spawn from the list.

After a spawned activity is created, it will be listed in the rule's folder. A finger icon will appear next to each of the spawns. When the icon is double-clicked, the spawned transaction's folder opens.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Configuration Panes > Debug Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Debug Pane

The Debug pane is a tool that can be used to debug errors in the math section of a policy- or client-level transaction. This tool moves through execution steps in sequence, and displays all math variables and their associated values at each step. Debugging not only walks through the transaction's configuration, but automatically opens any called functions, steps through all of the math variables and displays values according to what was passed in from the transaction configuration.

Before Using Debug

Before using the debug feature, make sure remote debugging is set up in the Rules Palette.

Steps to Check Remote Debugging in Rules Palette

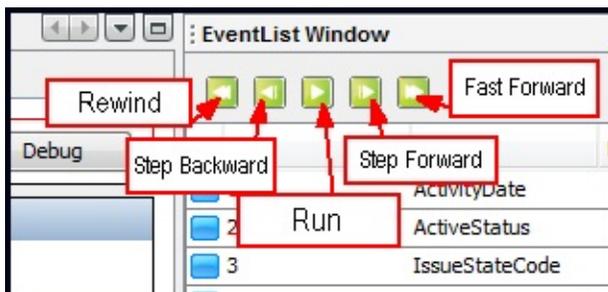
1. Right-click on the environment connection in the Main Explorer tab.
2. Select **Properties**. The Properties Window will open.
3. Scroll down to the row for the Debug Web Service URL.
4. Click the button to the right of the field. An Environment Properties window will open for the Debug Web Service. The URL should be listed.

Start and Run a Debug Session

Debug functionality is available for transactions at both the policy level and the client level.

The methods for debugging the configuration are as follows. The buttons for these functions are located at the top of the [EventList Window](#).

1. **Run:** Ignore all [breakpoints](#) and process the entire transaction.
2. **Step Forward:** Advance through the configuration one line at a time. On advancing, the line advanced to will be calculated and displayed. The Step Forward will step into and out of CopyBooks and Functions, as determined by the configuration.
3. **Fast Forward:** Advance through the configuration to the next breakpoint. The debugger will stop on the line where the breakpoint is checked. Breakpoints may be placed in CopyBooks and Functions. If no breakpoints are set, the debugger will process the entire transaction.
4. **Step Backward:** Step back one line in the configuration. This will “undo” or “clear” the step that was just performed, and take the user to the previous step, calculating and displaying that step as if the one that occurred after it was never determined.
5. **Rewind:** Return to the beginning of the configuration, allowing the user to start debugging over. All prior values calculated and displayed will be cleared, with the exception of the input values.



Buttons Available in the Debug Pane

To start a debug session for a **client-level transaction**:

1. Open the transaction in the Rules Palette. Make sure the transaction is checked in.
2. Go into the OIPA application and search for a client. Once the client is

found, open the client record.

3. Add the client transaction to the client. This can be done by clicking the **Add Activity** link on the Secondary menu and searching for the client transaction. When found, select it and click **Add**.
4. Leave the activity (transaction) in a pending status.
5. Go back to the Rules Palette and click the Debug pane in the transaction.
6. Click the drop-down box next to **Client** and select the client where you just added the activity (transaction).
7. If the client is not listed, click the **Search** button to the right of the Client field. Select a client type from the drop-down box. Then enter any search criteria. The % sign can be used as a wildcard, but it cannot be the only value in a search field.
8. If the transaction is listed multiple times for the client in OIPA, then the **Activity Date** drop down will populate according to effective date, with the most recent one at the top of the list.
9. Click the **Debug** button at the top of the window. It will take a few minutes for the debugger to load. When complete, a message will appear indicating that the session loaded successfully.
10. Initiate the debug session by using one of the methods described above. The EventList Window steps through the math variables, highlighting the variable currently being processed, and the ContextList Window displays the current state of each variable throughout the debug session.



All translation, compilation and runtime errors must first be fixed in order to use the Debug pane. Debug can be performed when the transaction is checked-in or checked-out; however, configuration problems can only be fixed if the transaction is checked out.

To start a debug session for a **policy-level transaction**:

1. Open the transaction in the Rules Palette. Make sure the transaction is checked in.
2. Go into the OIPA application and search for a policy. Once the policy is found, open the policy record.

3. Add the transaction to the policy. This can be done by clicking the **Add Activity** link on the Secondary menu and searching for the transaction. When found, select it and click **Add**.
4. Leave the activity (transaction) in a pending status.
5. Go back to the Rules Palette and click the Debug pane in the transaction.
6. Click the drop-down box next to **Policy** and select the policy where you just added the activity (transaction).
7. If the policy is not listed, click the **Search** button to the right of the Policy field, then enter any search criteria. The % sign can be used as a wildcard, but it cannot be the only value in a search field.
8. Click the **Debug** button at the top of the window. It will take a few minutes for the debugger to load. When complete, a message will appear indicating that the session loaded successfully.
9. Initiate the debug session by using one of the methods described above. The EventList Window steps through the math variables, highlighting the variable currently being processed, and the ContextList Window displays the current state of each variable throughout the debug session.



All translation, compilation and runtime errors must first be fixed in order to use the Debug pane. Debug can be performed when the transaction is checked in or checked out; however, configuration problems can only be fixed if the transaction is checked out.

The screenshot displays a software interface with several windows and a debug pane. The main window has tabs for Allocations, Fields, Events, Math, Assignment, Spawn, and Xml Source. The debug pane is active, showing the following information:

- Debugging: Issue
- Buttons: **Select policy to debug**, **Start Debug**, and Debug
- Policy: AVA31ALIC010843 - CIM VA 5
- ActivityDate: EffectiveDate(01/05/2005) InsertionDate(04/08/2009)
- System Date: Tue 06/23/2009
- Table of function calls:

Hierarchy	TYPE	DT	Attributes	Text	DT	BP
	VALUE	DATE			11/2	
	VALUE	INT		6	12/3	
	VALUE	INT		70	12/3	
	PLNFD	INT		MaturityAge	12/3	
	FUNCTIONCALL	DATE	FUNCTIONNAME = F		11/2	
	FUNCTIONCALL	DATE	FUNCTIONNAME = F		11/2	
	EXP	INT		True	12/3	
	EXP	DATE		AnnuitantYears70H:	11/2	

Below the table, the Name field is set to OwnerMaturityDate and the Data Type is DATE.

The EventList Window on the right shows a list of variables and their values:

#	Variable	Loop Na...
7	GMABExists	
8	AutoPayPro...	
9	Yes	
10	No	
11	AnnuitantA...	
12	AnnuitantY...	
13	AnnuitantY...	
14	AnnuitantAge	
15	OwnerAnnu...	
16	OwnerYear...	
17	OwnerYear...	

The ContextList Window at the bottom right shows a list of context variables and their values:

Name	Value
EffectiveDate	2005-01-05 00:00:...
OverrideSweepDate	
OverrideFreelookD...	
SystemDate	2031-01-29 00:00:...
QualType	02
EffectiveDateMV	
SystemDateMV	

Red boxes and arrows highlight the following elements:

- Select policy to debug** (button)
- Start Debug** (button)
- Step through math variables** (text box pointing to the table)
- Function call opened function configuration in next tab** (text box pointing to the function call rows)
- Math variable values by step** (text box pointing to the ContextList Window)

Debug Pane and Related Windows

EventList Window

In the EventList Window, a blue  or green  button appears next to the math variable execution steps. The blue button indicates the math variable is part of the transaction. The green button indicates the math variable is part of a called function. Double-click on either button to be brought to the math variable in the transaction or function configuration.



The remote debugger does not debug Function rules separately, only as called within a transaction.

ContextList Window

Use the ContextList Window to view the state of the math variable at each step in the debug session. If the value does not fully display in the column, click the ellipsis (...) button to see the full details. When a Valuation element is debugged, a dollar sign icon  is available instead of the ellipsis button. Clicking it will open a Policy Valuation window that displays valuation and fund information.

Note: To clear all values in the ContextList Window and restart debugging, click the **Rewind** button at the top of the EventList Window.

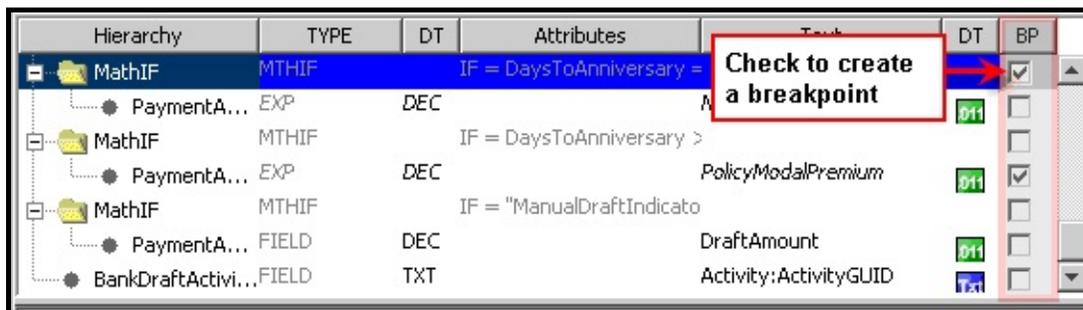


The Debugger will not display the values of MathUpdate math variables or the field itself.

Breakpoints

Another debugging feature is the ability to set-up breakpoint(s) in the configuration. Breakpoints are places in the code where execution is stopped for debugging purposes. In the Rules Palette, select as many breakpoints as needed by checking the **BP** checkbox. After breakpoints are selected, use the Run button  in EventList window to step through configuration.

Important: Breakpoints can be set for a loop that then displays math variables and associated values for each iteration of the loop. Use the Run button  to step through each iteration of a loop.



Hierarchy	TYPE	DT	Attributes	DT	BP
MathIF	MTHIF		IF = DaysToAnniversary =		<input checked="" type="checkbox"/>
PaymentA...	EXP	DEC		011	<input type="checkbox"/>
MathIF	MTHIF		IF = DaysToAnniversary >		<input type="checkbox"/>
PaymentA...	EXP	DEC	PolicyModalPremium	011	<input checked="" type="checkbox"/>
MathIF	MTHIF		IF = "ManualDraftIndicato		<input type="checkbox"/>
PaymentA...	FIELD	DEC	DraftAmount	011	<input type="checkbox"/>
BankDraftActivi...	FIELD	TXT	Activity:ActivityGUID	1x1	<input type="checkbox"/>

Breakpoint Column on Debug Pane

To clear all the breakpoints, select the Clear All  button.

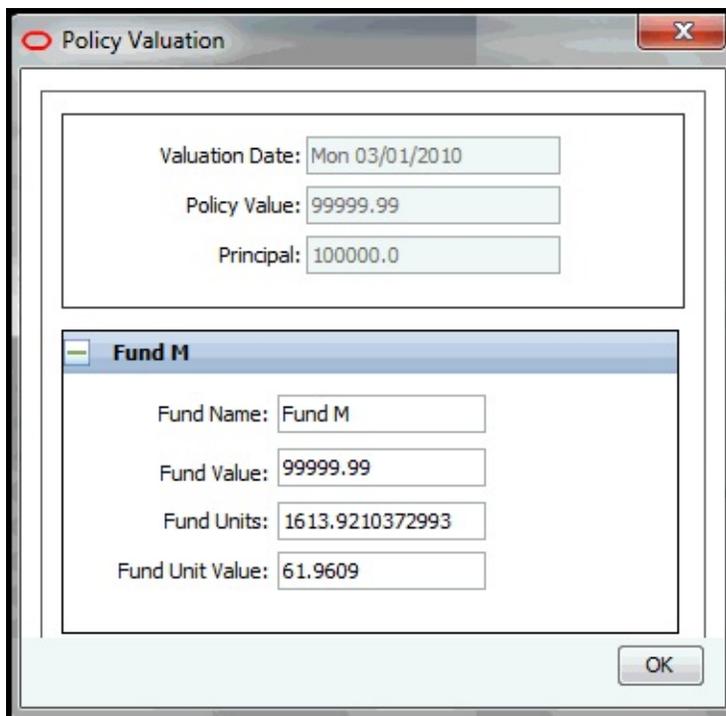
Policy Valuation in Debugger

You are here: Configuration > Configuration Panes > [Debug Pane](#) > Policy Valuation in Debugger

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Policy Valuation in Debugger

When a valuation element is debugged in the Rules Palette, a special icon is listed to the right of the item in the Context Window. Clicking the dollar sign icon  will open a Policy Valuation window. This window displays valuation values as calculated by OIPA. Valuation is executed prior to activity math so it is available once the debugger initializes for the activity. Currently policy valuation is executed only once and it is as of the activity effective date.



Valuation Date:	Mon 03/01/2010
Policy Value:	99999.99
Principal:	100000.0

Fund M	
Fund Name:	Fund M
Fund Value:	99999.99
Fund Units:	1613.9210372993
Fund Unit Value:	61.9609

OK

Policy Valuation Window in Debugger

You are here: Configuration > Configuration Panes > Shadow Policy Pane

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Shadow Policy Pane

The Shadow Policy pane configures the **Delete/Shadow** (soft delete) button on the [Policy Screen](#). The **Delete/Shadow** button enables an OIPA user to delete a policy *after* the policy has been saved. This might be used, for example, when a policy has been created in error and is still in Pending status. The statuses that can be deleted are configured in the [CompanyCosmetics rule](#); the Shadow Policy pane allows the Rules Palette user to select which of the configured statuses in CompanyCosmetics will make a policy eligible for deletion, which predefined validations are performed, and what status is assigned to a policy when it is deleted by the user. When a policy is eligible for deletion, the Delete/Shadow button is visible on the secondary menu of the Policy Screen. If the policy status is not eligible, the Delete/Shadow button is hidden.

The Policy Status codes are selected from AsCodeStatus in the AsCode table. The list of status codes is configurable in CompanyCosmetics. It specifies statuses that allow deletion; any status not listed in CompanyCosmetics does not allow deletion. The typical use for the Delete/Shadow button is to delete a policy that was just saved, has not been assigned roles, activities, and segments, and is not yet active.

To configure the Delete/Shadow button, the following must be set up:

- The **Shadow Status Code** field specifies the status that is assigned to a policy when it is shadowed.
- The **Allow Shadow** button specifies the policy statuses that are eligible for shadowing. At least one status must be configured.
- The **Validations** button specifies optional validations to be performed when the policy is shadowed. Four validations are pre-configured in OIPA. Any or all of them can be configured. The validations are:
 - **Activity** – A validation message is displayed if there are activities on the policy with the following activity status and type codes:

- The activity type code is 02 or 03, and the status code is 02, 09, 13 or 14.
- The activity type code is 01 or 04, and the status code is 01, 02, 09, 11, 13, 14, 57, or 58.

The validation message text (subject to translation) is: "Policy has activities associated. Please remove activities to continue."

- **Role** – A validation message is displayed if the policy has any policy or segment roles associated with it other than the CSR (RoleCode 11), .

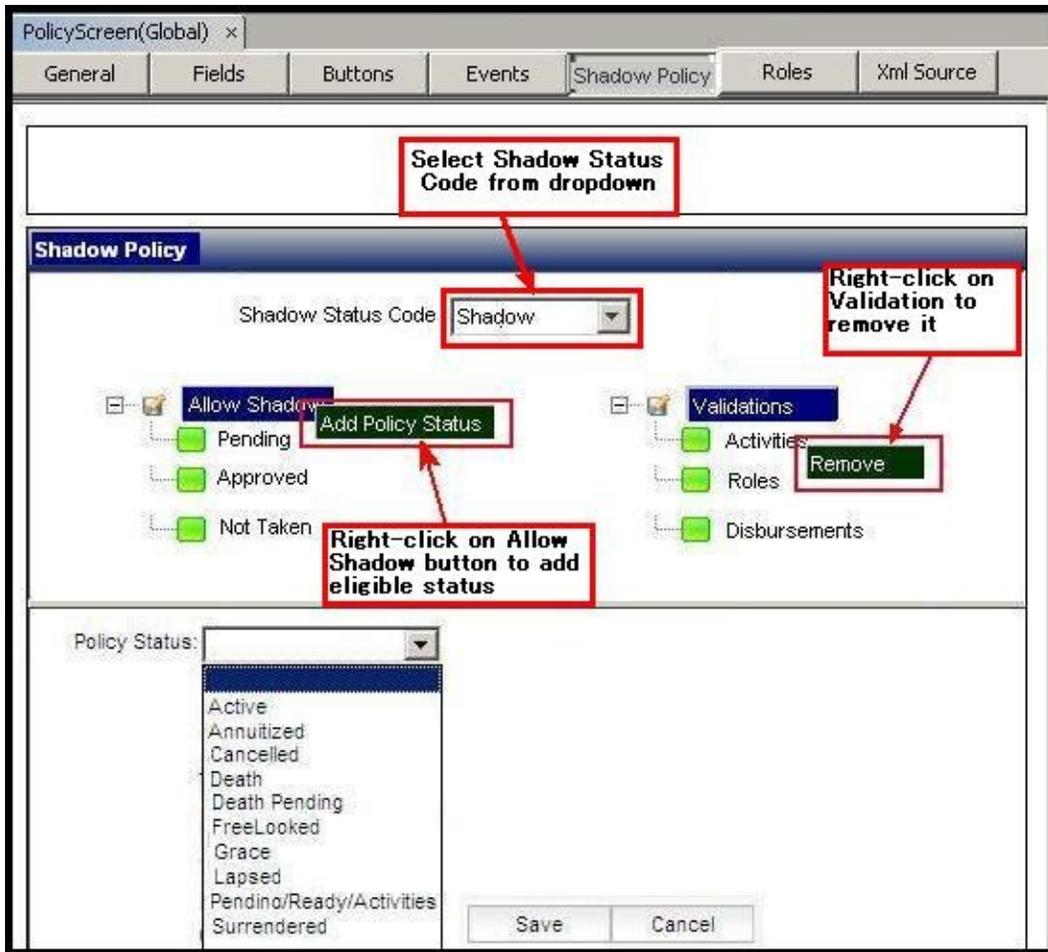
The validation message text (subject to translation) is: "Policy has policy or segment roles associated. Please remove all associated roles except CSR role to continue."

- **Segment** – A validation message is displayed if there are any segments associated with the policy.

The validation message text (subject to translation) is: "Policy has segments associated. Please remove all segments to continue."

- **Disbursement** –A validation message is displayed if the policy has any disbursements in status code 01, 02, or 12.

The validation message text (subject to translation) is: "Policy has disbursements associated. Delete action cannot be performed on policies with disbursements."



PolicyScreen Shadow Policy Pane

Steps to Configure the Shadow Policy Pane

1. Check out the **PolicyScreen** business rule.
2. Click the **Shadow Policy** pane in the Configuration Area.
3. Select one Shadow Status Code from the drop-down box.
4. Right-click on the **Allow Shadow** field and select “**Add Policy Status**.”

This action splits the pane into two sections (upper and lower). The lower pane contains the Status Code drop-down box, and the **Save** and **Cancel** buttons.

The **Cancel** button closes the lower pane without saving any changes.
5. Select an appropriate policy status and click on the **Save** button.

This action adds a Policy Status sub-node (e.g. “Pending”) to the **Allow Shadow** tree node. Once a policy status is saved, the lower pane closes.
6. To add another policy status, repeat Steps 4 and 5.
7. To remove a policy status from the Allow Shadow node, right-click on the status to be removed, and select **Remove**.
8. Right-click on the **Validations** field and select **Add Validation Item**.

This action splits the pane into two sections (upper and lower). The lower pane contains the Validation Item drop-down box, and the **Save** and **Cancel** buttons.

The **Cancel** button closes the lower pane without saving any changes.
9. Select the appropriate validation item and click on the **Save** button.

This action adds a Validation Item sub-node (e.g. “Activities”) to the Validation tree node. Once a Validation item is saved, the lower pane closes.
10. To add another Validation item, repeat Steps 8 and 9.
11. To remove a Validation item from the Validations node, right-click on the Validation item to be removed, and select **Remove**.
12. When complete, save and check in the rule.
13. Under the AdminExplorer, enable the ShadowPolicy button for the

You are here: Configuration > Business Rules > Business Rules Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Business Rules Overview

The core processing within the Oracle Insurance Policy Administration (OIPA) system is controlled by a set of rules. Rules are used to configure business processing. They control the creation, display and validation of screens and provide the means for displaying, entering and processing transactions. OIPA rules are stored to the database in XML.

The Rules Palette provides visual editors that create the XML for the user for many of the screen business rules. Some rules must be configured in XML. The [XML Editor](#) is used to configure the XML and should be used in conjunction with the V9 XML Configuration Guide. View the **XML Configuration Guide** topic in this help system for additional information on rule configuration.

Important: For any configuration in the Rules Palette, if an element or attribute is not available in the visual editor, it can be manually configured that is necessary for your configuration in a visual editor, you can manually configure it in the rule through the [XML Source Pane](#) when the rule is open in the Configuration Area.

Organization

Business rules are organized by [type](#). Global rules can be accessed from the Global Rules Explorer. Overrides to those rules can be created there. Once an override is created, it can be accessed from the Global Rules Explorer in the folder for the global rule or in the Main Explorer, under the **Company | Plan | Business Rules** folder for the specific override.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Business Rules > Types of Rules in the Rules Palette

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Types of Rules in the Rules Palette

The Rules Palette assigns each rule to a rule category according to the type of rule. The TypeCode is stored in a column called XMLSchema in the database. It is updated by scripts and is used by the Rules Palette for organization and performance. If a rule is created outside of the Rules Palette and a rule category is not assigned to the rule, then it will not be viewable through the Rules Palette. Rules Palette TypeCodes are different than the type code in [Code Names](#) in the [Admin Explorer](#) window.

The various types of business rules are listed below. All rule types are defined in the [Global Rules Explorer Tab](#) section.

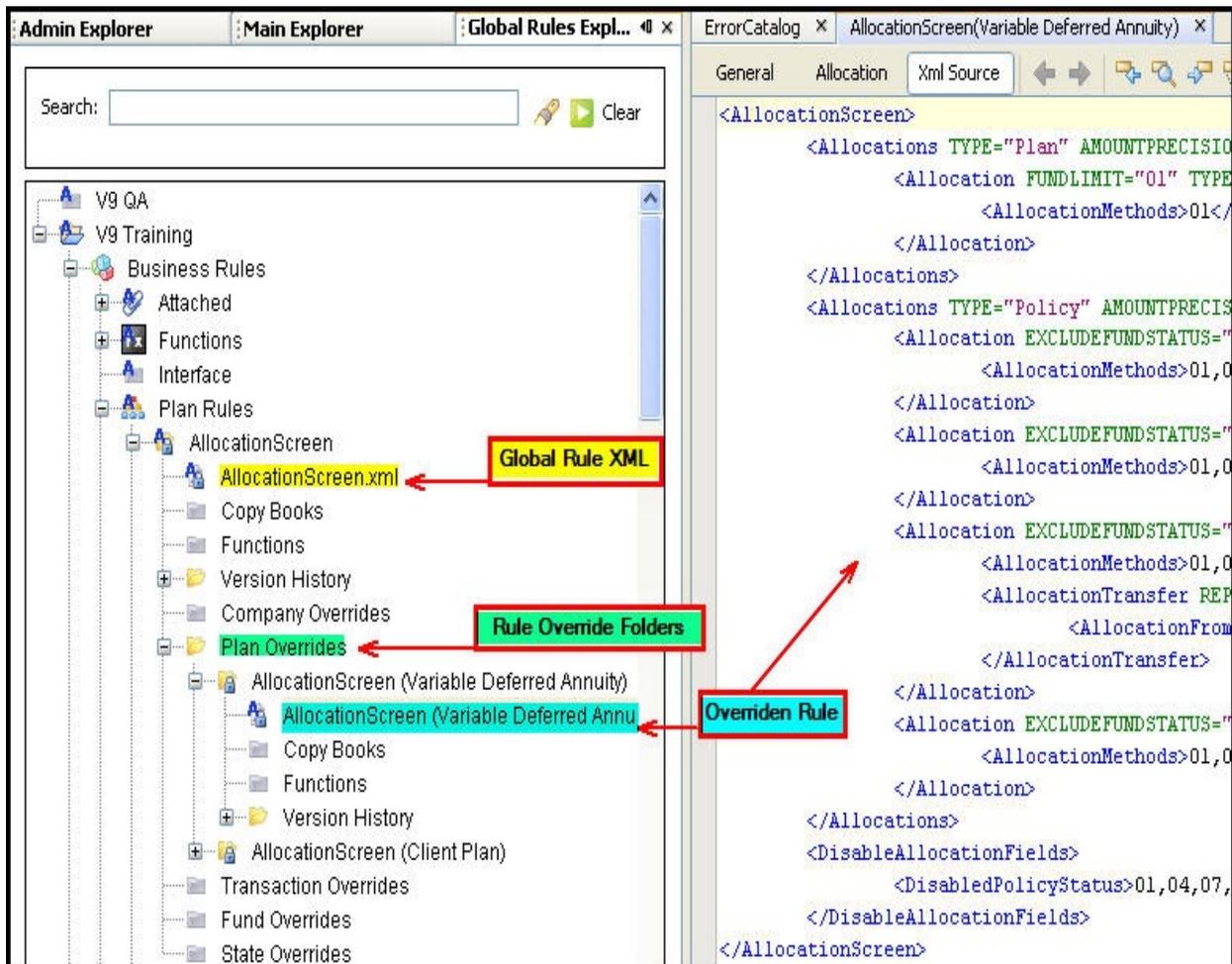
- Attached
- Calculate
- CopyBooks
- File
- Functions
- Interface
- Plan Rules
- Screen Rules
- System Rules
- System Calculations
- System Validations
- Table File
- User Defined

You are here: Configuration > Business Rules > Global Business Rules

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

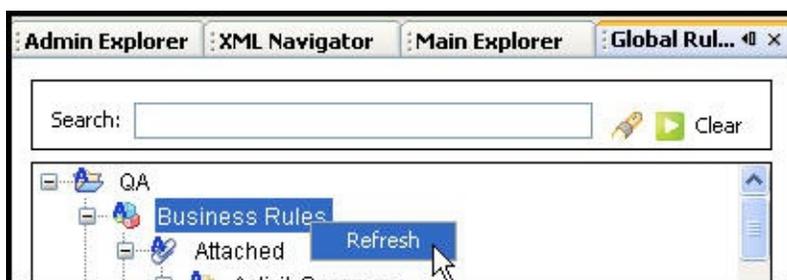
Global Rules

Global rules can be viewed via the [Global Rules Explorer](#) tab. Each global rule has its own folder, which includes the XML for the rule, rule meta data and associated CopyBooks and functions. Rule overrides are displayed in the global rule's folder with the same folder breakdown as the global rule. If the rule is overridden at the plan level then the rule is also viewable in the Main Explorer window under the plan's folder.



Rule Folder Structure in Global Rules Explorer

The Business Rules folder has a right-click **Refresh** option. Select one of the main rule folders and click the Refresh button to update the display of the rules. Any recently added or revised rules that other configurators have been working on will display.

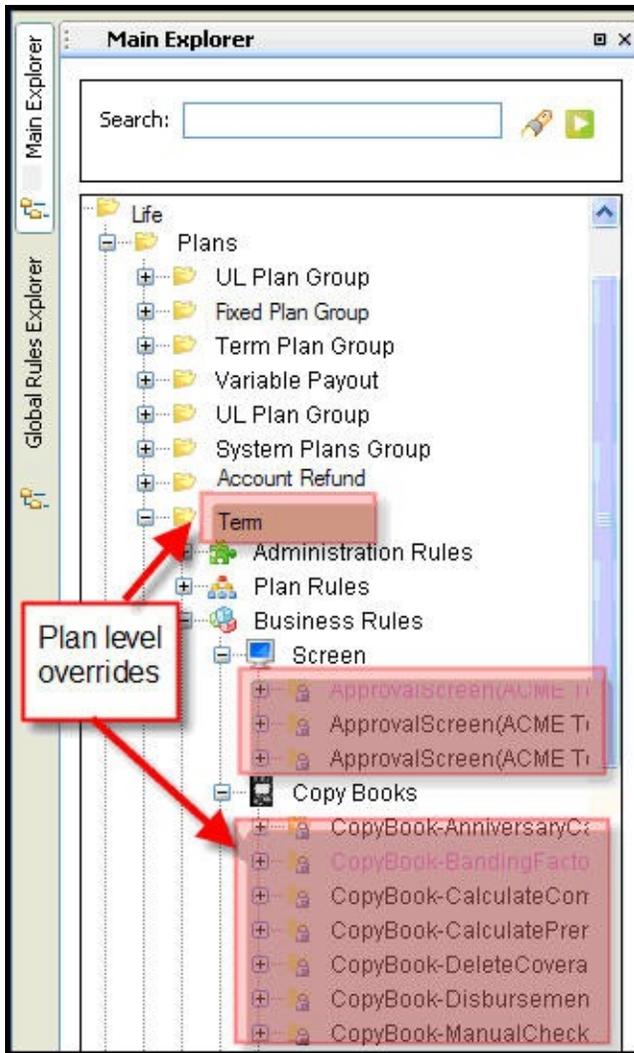


Business Rules Right-Click Option

Global Rules Overrides

A global rule must exist in order to [create an override of the rule](#). If there is an exception to a global rule, an override can be used to support this. Rules that are overridden at the plan level are located in the plan folder in their respective rule folder via the Main Explorer window.

Note: Rules may also be overridden at the Product level. If Product functionality is enabled, then Product will be a valid override level..



Plan Level Overrides

You are here: Configuration > Business Rules > Create a New Rule Override

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create a New Rule or Rule Override

There are two ways to add business rules through the Rules Palette. A new global rule can be created or an existing rule can be overridden and configured specifically for a company, plan, transaction, state or fund. All rules and rule overrides are created in the Global Explorer.

Steps to Create a New Global Rule

1. Navigate to the Global Rules Explorer.
2. Open the Business Rules folder.
3. Right-click on a rule category and select **New Global Rule for category " "**.
4. Type the rule name in the first step of the wizard and click **Next**.
5. Select the [TypeCode](#) from the drop down box.
6. Click **Finish** on step three of the wizard. The template information is not required.
7. [Configure](#) the rule.

The new global rule will appear in the category folder where it was created.



Attached Rules Right-Click Option

Overrides

When a rule override is created, a copy of the original rule (usually the global version) is used as the starting point for configuration. Then any changes made to the rule can be saved and used at the specified override level in OIPA. For example, a plan override of the PolicyScreen rule would allow a particular plan to have a Policy screen that is different from the Policy screens used by other plans. A rule without any overrides is a global version of the rule. OIPA always looks for override levels starting with the most specific override level and moving up to the least specific (global - meaning no override) level. Override levels are listed below from most specific to least specific.

- Fund
- State
- Requirement
- Transaction
- System (if functionality is enabled)
- Plan
- Child Product (if functionality is enabled)
- Product (if functionality is enabled)
- Subsidiary Company
- Primary Company



The System override level will only be available if New Business Underwriting is enabled. The Product and Child Product override levels will only be available if Products are enabled.

Overrides can be created for the various types of rules and attached to specific plans. From the Global Rules Explorer, new global business rules can be created for each of the rule categories. Right-click on a rule's folder and select **New Override** to open a wizard window, which will step through the creation process.

A list of all rules and the override levels that are supported in OIPA for

that rule (the rule's **context**) can be found in the **XML Configuration Guide**. The Transaction Rules book contains a page called **Attached Rule Validation**, which lists all rules and their corresponding context.

If [Product](#) functionality is enabled, then an additional override level will be available between the Company and Plan fields. Once the company is selected, the Product drop down will list the applicable Products for that company. Once the Product is selected the Plan drop down will list the applicable plans for that Product.



If Product is disabled (in Properties setting) and there are existing Products, then they will appear in the Plan drop down box.

Company:	<input type="text"/>
Product:	<input type="text"/> Look up
Plan:	<input type="text"/>
System:	<input type="text"/>
Transaction:	<input type="text"/>
State:	<input type="text"/>
Fund:	<input type="text"/>
TypeCode:	Screen

Product Option in Rule Override Wizard

Steps to Create a Rule Override

1. Navigate to the Global Rules Explorer.
2. Open **Business Rules** and the category folder of the rule to override.
3. Right-click on the global rule to override and select the override menu option.



Activity Summary Right-Click Option

4. Click **Next** on the first screen of the wizard. The name of the business rule cannot be changed.
5. Select the appropriate override settings for the business rule. The level of override selected will determine what folder the override appears in under the Global Explorer. When looking for the rule, make sure to open the appropriate Override folder.
6. Click **Finish**.
7. [Configure](#) the rule.

Locate Override in Global Explorer

Override folders are provided for each level supported in OIPA. To find a rule override, open the rule in the Global Explorer. The global rule is listed first and all overrides reside in the corresponding Override folder.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Business Rules > Configure a Rule Override

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Configure a Rule Override

Configuring rule overrides will differ depending on the type of rule. Rules that are associated with screens have visual editing tools in the Rules Palette for configuration. Rules that enhance processing are configured using XML tags, which are described in detailed in the XML Configuration Guide. View the **XML Configuration Guide** topic from the Help option in the Rules Palette Main menu.

The Rules Palette only provides configuration tools for visual editing when applicable. Rules requiring XML coding will have [XML Source editing](#) options available.

Steps to Configure a Rule Override

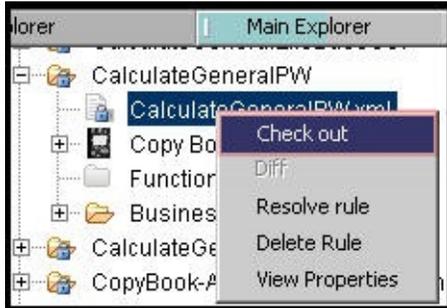
1. Locate the rule's folder in the Main Explorer or in the Global Rules Explorer if the rule is a global business rule.
2. Expand the rule's folder.
3. Right-click on the rule's XML file.
4. Select **Check-out**. See the [Checking In/Out](#) section for further detail.
5. Configure the rule accordingly. Use the XML Configuration Guide in **Help** for information on rule configuration.
6. Right-click on the rule's XML file and select **Check-in**. This will save the changes to the database.

You are here: Configuration > Business Rules > Check-out and Check-in Rules

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

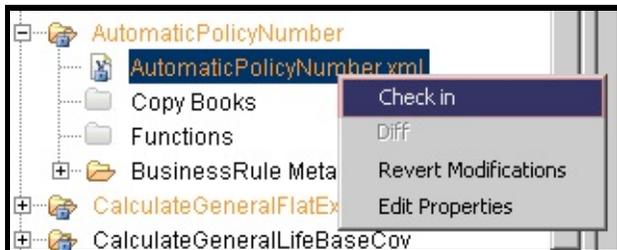
Check-out and Check-in Rules

If a rule is available for check-out, it will be listed in black. Right-click on a rule's XML file and select **Check-out**. Work with the rule as needed.



Rule Right-Click Check-out Option

Once a rule is checked out, it will be listed in orange. After the rule has been configured as needed, right-click on the XML file and select **Check in**. Any changes will be saved to the database and the rule will again be listed in black.



Rule Right-Click Check-in Option

Note: If configuration changes were made to a rule, but then need to be discarded, select **Revert Modifications** from the right-click menu.

Rule Color Scheme

Rules are displayed in various colors. The following list contains an explanation of the colors used for rule display in the Rules Palette.

- **Orange**: Rule is checked out by the user.
- **Black**: Rule is available for check-out.
- **Gray**: Rule is checked out by someone other than the user.
- **Blue**: Rule is new and has not yet been saved to the database.

You are here: Configuration > Transactions > Save Configuration Changes

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Save Configuration Changes

Configuration changes can be saved without actually checking-in the rule. Use the Save  button to save the configuration changes in a file or use the Save All  button to save changes made in all open files. The changes are saved locally to the computer being used, but not written to the database. If the configuration is complex and takes several days to complete, then this is a useful option, as configuration can be saved and the user can log-off the computer without checking-in the rule. Remember to come back to the same computer where the configuration was saved since the Save and Save All buttons only save changes locally.

Other users may view the rule that is checked-out, but they will only see the configuration that is stored in the database. They will not see any configuration changes that were saved locally.

Save All saves all of the rules that are open and validation errors will be presented for each of the open rules. Any validation errors need to be resolved and Save All should be done again.

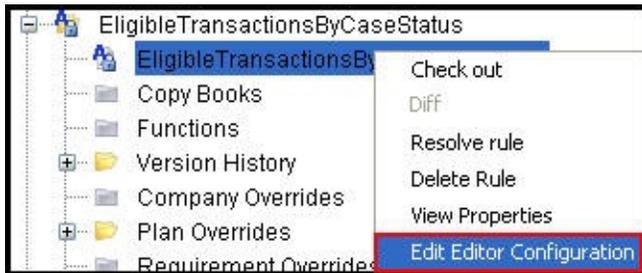
Local changes won't be saved from session to session on rules that have not been saved to the database initially, because new rules need to either be checked in or discarded when logging out of the Palette.

You are here: Configuration > Business Rules > Edit Editor Configuration

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Edit Editor Configuration

This is an option on the business rule right-click menu. It is a development tool and should not be used during configuration.



Edit Editor Configuration Option on Rule Right-Click Menu

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

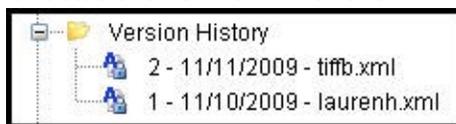
You are here: Configuration > Segments > Version History

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Version History

The Version History feature tracks all of the changes that have been made to [business rules](#), [transactions](#), [segments](#) and [requirements](#). Each time a record is checked-out and updated, a new history file is created and saved.

The Version History node will display the number of the new version, the date the new version was created and the name of the system user who created the new version.



Version History Files

Use the version history files to [replace an existing XML file with a previous history version](#) or [compare two history files](#) to see the differences between them.

The Version History folder can be found from the **Main Explorer** window, or, for requirements, in the **Admin Explorer** window. Navigate down through the directory tree structure to find the appropriate rule, transaction, segment or requirement file. A folder will appear under that specific file's tree. The folder is labeled Version History and will contain a copy of the record for each change that was made.

Important: If changes are made to the rule, transaction, segment or requirement while using different IVS tracks, only changes saved on the track being used will display. It may appear that Version History files are missing; however, they exist on the track where they were saved.

Using Version History to Manage Modifications

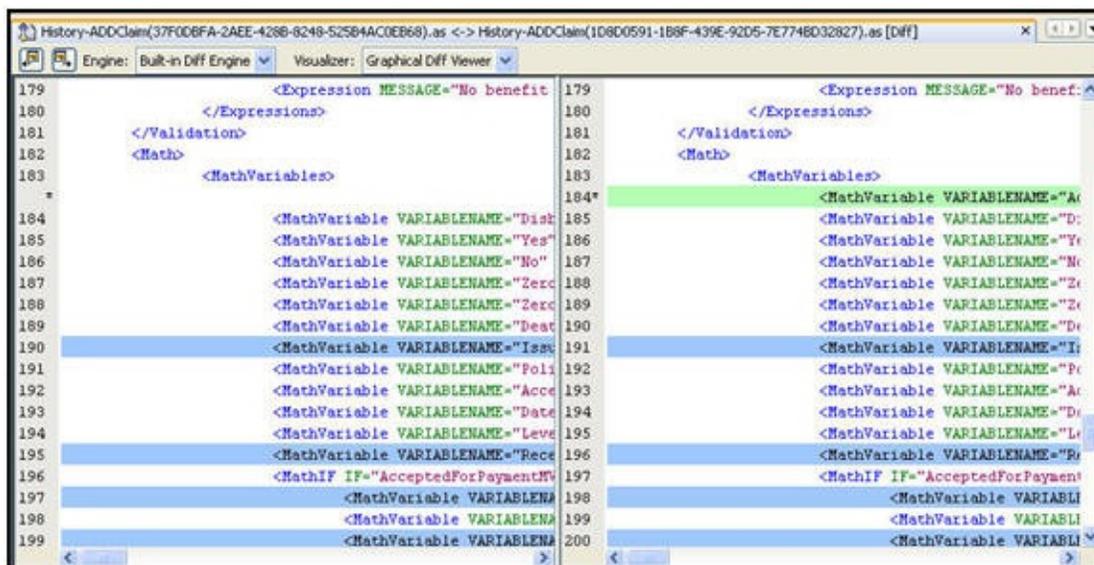
Version History is a helpful tool for managing the changes made to rules, transactions, segments and requirements. Use it to replace an existing XML file with a previous version of the file. It can also be used to compare two different versions of a rule in a side-by-side comparison.

Steps to Replace Existing XML File with a Version History File

1. Open the **Version History** folder for the rule, transaction, segment or requirement.
2. Right-click on the version number to use to overwrite the existing XML file and select **Revert to this version**.
3. Right-click on the file a second time and select **Check-in**. This will save the changes to the database and create a new version history file.

Steps to Compare Different Versions of a Rule

1. Open the **Version History** folder under the appropriate rule, transaction, segment or requirement.
2. Select a **Version History** file by clicking on it.
3. Hold down the **CTRL** key and select a second Version History file.
4. Right-click and select **Diff**. Both files will open up side-by-side in the [Configuration Area](#). Variations will be highlighted in blue and missing XML will be highlighted in green.



```
History-ADDClaim(37F0DBFA-2AEE-426B-8248-525B4AC0EB68).as <-> History-ADDClaim(1D8D0591-1B8F-439E-92D5-7E7748D32827).as [Diff]
Engine: Built-in Diff Engine Visualizer: Graphical Diff Viewer
179 <Expression MESSAGE="No benefit 179 <Expression MESSAGE="No benefi
180 </Expressions> 180 </Expressions>
181 </Validation> 181 </Validation>
182 <Math> 182 <Math>
183 <MathVariables> 183 <MathVariables>
184 184* <MathVariable VARIABLENAME="A
185 <MathVariable VARIABLENAME="Dis 185 <MathVariable VARIABLENAME="D
186 <MathVariable VARIABLENAME="Yes" 186 <MathVariable VARIABLENAME="Ye
187 <MathVariable VARIABLENAME="No" 187 <MathVariable VARIABLENAME="No
188 <MathVariable VARIABLENAME="Zerc 188 <MathVariable VARIABLENAME="Ze
189 <MathVariable VARIABLENAME="Deat 189 <MathVariable VARIABLENAME="De
190 <MathVariable VARIABLENAME="Isav 190 <MathVariable VARIABLENAME="I
191 <MathVariable VARIABLENAME="Poli 191 <MathVariable VARIABLENAME="P
192 <MathVariable VARIABLENAME="Acce 192 <MathVariable VARIABLENAME="Ac
193 <MathVariable VARIABLENAME="Date 193 <MathVariable VARIABLENAME="De
194 <MathVariable VARIABLENAME="Leve 194 <MathVariable VARIABLENAME="L
195 <MathVariable VARIABLENAME="Rece 195 <MathVariable VARIABLENAME="R
196 <MathIF IF="AcceptedForPayment 196 <MathIF IF="AcceptedForPayme
197 <MathVariable VARIABLENAME 197 <MathVariable VARIABLE
198 <MathVariable VARIABLENAME 198 <MathVariable VARIABLE
199 <MathVariable VARIABLENAME 199 <MathVariable VARIABLE
200 <MathVariable VARIABLENAME 200 <MathVariable VARIABLE
```

Version History Records Compared with Diff Option

You are here: Configuration > Transactions > Transactions Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transactions Overview

Businesses have processes that need to be supported for policies and throughout the policy lifecycle. Transactions are configured to be business events that users in the Oracle Insurance Policy Administration system (most likely Customer Service Representatives) need to process when an event occurs that affects a policy. Some transactions are configured to automatically occur without end user initiation. No matter how the transaction processes, it must be configured to support the business event processes that need to occur.

A list of sample transactions that can be configured is shown below.

- Applying Premium to a Policy
- Issue a Policy
- Send a Grace Letter
- Change Beneficiary
- Pay a Death Benefit
- Add or Delete Riders

Transaction Processing and Sequence

A transaction is made up of multiple sections designed to execute a specific type of processing. A transaction can be configured to execute once, with a breakpoint in processing where valuation logic is executed. The [ValidateExpressions](#) business rule is attached to a transaction to accomplish this.

A transaction can also be configured to execute repeatedly. This is necessary for unit linked products when actual fund values are needed, but not yet available at the time the transaction first executes. In cases such as these, a transaction can be configured to allow the latter sections of the transaction to repeatedly execute until the information is available. The PostAssignmentValidateExpressions business rule is attached to a transaction to accomplish this.

The sections of a transaction process in the following order:

1. Suspense
2. Valuation
3. Math
4. **Attached Business Rules** - configurable processing breakpoint
5. Assignment
6. **Post Assignment Business Rules** - second configurable processing breakpoint
7. Disbursement
8. Accounting
9. Spawns

After a transaction has been [configured](#) and has been assigned [security](#), it is located on the Activities Tab in the activity drop-down box. An activity is an instance of a transaction or the actual use of a transaction on a policy. Every time an activity is processed, the transaction configuration is executed.

Transaction Override Levels

Transactions are able to overridden at the following levels, listed from most to least specific:

- Plan
- Client
- Child Product
- Product

Transactions are broken down by type. The following transaction types are available:

- Policy
- Plan
- Client
- Cycle
- Intake

See the [Transaction Types, Status and Processing Order](#) page for more information on transaction types.

If multiple plans, clients, Child Products or Products use the same transaction, a [CopyBook](#) can be created that contains the shared information. Copybooks eliminate the need to configure multiple transactions at the same override level. It also means that the shared configuration is maintained in one place, making updating transaction configuration significantly easier.

You are here: Configuration > Transactions > Transaction Types, Status and Processing Order

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transaction Types, Status and Processing Order

Transaction type indicates the type of transaction processing that is needed. Most transaction rule types are overridden at the plan level and are configured as Policy-Financial. Transaction types are named according to their level and the type of processing they perform. A financial transaction does not necessarily imply that there is a movement of funds. It is a designation within the system that is used to identify activities that impact data or initiate processing at the policy or plan level.

All client level transactions must be performed at the client level. Locate the client folder under the appropriate plan and then create the client level transaction.

The following list contains an explanation of the transaction types that are supported in OIPA.

Transaction Types and Definitions

Transactions can be processed as activities in OIPA at various levels. The type of transaction will determine the level where it is processed and the impact it will have on the activities around it. In order to understand the various types of transactions and their impact on other activities in OIPA, definitions are provided below. An explanation of the various types of transactions are provided below the definitions.

Definitions

Reversible: the activity can be reversed and recycled by clicking the trash can/recycle button next to it. The Delete checkbox is unnecessary. Recycling and reversing any Reversible activities that were processed prior to this activity will cause this activity to be reprocessed.

Self-Reversible: the activity can be reversed and recycled by clicking the trash can/recycle button next to it. The Delete checkbox is unnecessary. Recycling and reversing any Reversible activities that were processed prior to this activity will **NOT** cause this activity to be reprocessed.

Non-Reversible: the activity does not have a trash can/recycle button next to it. It can only be reversed and recycled using the Delete checkbox. Recycling and reversing any Non-reversible activities that were processed prior to this activity will NOT cause this activity to be redone.

Reversing: Recycling and reversing this activity will cause reversible activities above it to be redone. Recycling and reversing this activity prior to processed activities will cause reversible activities above it to be redone.

Non-Reversing: Recycling and reversing this activity will NOT cause reversible activities above it to be redone. Recycling and reversing this activity prior to processed activities, will NOT cause reversible activities above it to be redone.

Types

Keeping in mind the definitions explained above, the transaction type describes the transaction level and impact on surrounding activities. Each transaction type is listed below, according to level. Transaction names that only have a level indicated are assumed to be Reversible-Reversing.

Client

- **Client- Batch:** client level batch transactions that can be reversed and recycled using the trash can/recycle button. Recycling and reversing any activities prior to it will cause this activity to be reprocessed.
- **Client-Document:** client level documents that can be reversed and recycled using the trash can/recycle button. Recycling and reversing any activities prior to it will cause this activity to be reprocessed.
- **Client-Document-Nonreversible-Nonreversing:** client level documents that cannot be reversed and recycled without using the Delete checkbox on the Activity screen. Recycling and reversing any activities prior to it will NOT cause this activity to be reprocessed.
- **Client-Extract:** client level extracts that can be reversed and recycled using the trash can/recycle button. Recycling and reversing any activities prior to it will cause this activity to be reprocessed.
- **Client-Financial:** client level financial transactions that can be reversed and recycled using the trash can/recycle button. Recycling and reversing any activities prior to it will cause this activity to be reprocessed
- **Client-Financial-Reversible-Reversing:** Client level financial transactions that can be reversed and recycled in active status. Once recycled/reversed, it will undo/redo all the active activities lying above it in activity hierarchy unless activity is non-reversible type. **.Processing** : Transaction will undo/redo all the active activities lying above it in activity hierarchy unless activity is non- reversible type, It can be recycled/reprocessed based on normal undo/redo processing.
- **Client-Financial-Reversible-Nonreversing:** Client level financial transactions that can be reversed and recycled by clicking the trash can/recycle button next to it. Recycling and reversing this activity will NOT

cause activities after it in activity hierarchy to be recycled.

Cycle

- **Cycle-Document:** plan level documents during nightly cycle.
- **Cycle-Extract:** plan level extracts during nightly cycle.
- **Cycle-Import:** plan level imports during nightly cycle.

Intake

- **Intake-Record:** transactions that execute business logic (Math) and, using conditional logic, build activities to be inserted and executed relative to a member record. Intake-Record transactions are not eligible for spawning from any other transactions, including other Data Intake transactions, nor will they appear on any user-eligible transaction lists.
- **Intake-File:** transactions that calculate statistics for files received through Data Intake. Intake-File transaction configuration only consists of a <Math> section, which supports all standard MathVariable types. Additionally, the <Math> section supports MathVariables with TYPE="FIELD" in order to retrieve the fixed and dynamic field values from the file.

Plan

Client Relationship transactions are created at the plan level. Create a plan for each of these transaction types.

- **Plan-Batch:** batch entry of activities.
- **Plan-Document:** plan level documents that can be reversed and recycled using the trash can/recycle button. Recycling and reversing any activities prior to it will cause this activity to be reprocessed.
- **Plan-Document-Nonreversible-Nonreversing:** plan level documents that cannot be reversed and recycled without using the Delete checkbox on the Activity screen. Recycling and reversing any activities prior to it will NOT cause this activity to be reprocessed.

- **Plan-Extract:** plan level extracts that can be reversed and recycled using the trash can/recycle button. Recycling and reversing any activities prior to it will cause this activity to be reprocessed.
- **Plan-Financial:** plan level financial transactions that can be reversed and recycled using the trash can/recycle button. Recycling and reversing any activities prior to it will cause this activity to be reprocessed.
- **Plan-Financial-Nonreversible-Nonreversing:** plan level financial transactions that cannot be reversed and recycled without using the Delete checkbox on the Activity screen. Recycling and reversing any activities prior to it will NOT cause this activity to be reprocessed.

Policy

- **Policy-Document:** policy level document that can be reversed and recycled using the trash can/recycle button. Recycling and reversing any activities prior to it will cause this activity to be reprocessed.
- **Policy-Document-Nonreversible-Nonreversing:** policy level documents that cannot be reversed and recycled without using the Delete checkbox on the Activity screen. Recycling and reversing any activities prior to it will NOT cause this activity to be reprocessed.
- **Policy-Document-Reversible-Nonreversing:** policy level documents that can be reversed and recycled by clicking the trash can/recycle button next to it. Recycling and reversing this activity will NOT cause activities after it to be reprocessed.
- **Policy-Extract:** policy level extracts that can be reversed and recycled using the trash can/recycle button. Recycling and reversing any activities prior to it will cause this activity to be reprocessed.
- **Policy-Financial:** policy level financial transactions that can be reversed and recycled using the trash can/recycle button. Recycling and reversing any activities prior to it will cause this activity to be reprocessed.
- **Policy-Financial-Nonreversible-Nonreversing:** policy level financial transactions that cannot be reversed and recycled by clicking the trash can/recycle button next to it. Recycling and reversing this activity will NOT cause reversible activities after it to be reprocessed.

- **Policy-Financial-Reversible-Nonreversing:** policy level financial transactions that can be reversed and recycled by clicking the trash can/recycle button next to it. Recycling and reversing this activity will NOT cause activities after it to be reprocessed.
- **Policy-Illustration:** policy level illustration transactions.

Reversible-Reversing Transaction Types

Reversible/Reversing transactions can be recycled /reversed in active status. Once recycled/reversed, it will undo/redo all the active activities lying above it in activity hierarchy unless activity is non-reversible type.

Processing : Transaction will undo/redo all the active activities lying above it in activity hierarchy unless activity is non-reversible type, It Can be recycled/reprocessed based on normal undo/redo processing.

Reverse/Recycling:

- If reversed, it will back out an Active activity by restoring the affected entity (ies) to prior state (i.e. Undo) but will not create a new instance of the selected activity (i.e. Redo).
- If recycled, It will back out an Active activity by restoring the affected entity (ies) to prior state (i.e. Reverse/Undo) and automatically recreate a new instance of the selected activity (i.e. Redo/Recycle) for reprocessing.
- Once recycled/reversed, it will undo/redo all the active activities lying above it in activity hierarchy unless activity is non-reversible type.
- It will automatically back out any spawned Client-level activity as a result of the Recycle, Reverse or Undo/Redo processing of its originating/source activity (i.e. the activity that spawned it or any of its predecessors in the case of a series of spawned activities) unless spawned activity is non-reversible type.
- Reversal/Undo of activities/transactions spawned from a Client-level activity/transaction to another processing level (e.g. Policy Level) should be handled separately.

See the **Security for Reversible-Reversing Transaction Types** section below for additional information.

The Reversible-reversing transaction may be inserted in between other active activities without invoking any undo/redo actions on the surrounding activities. Security may be given to the Delete checkbox on

the Activity List screen. When that checkbox is selected, all activities get a trashcan and a recycle icon. This new transaction type will also get a trashcan and recycle icon like all others. The transaction can be spawned from all transaction types and can spawn all transaction types.

See the **Security for Reversible-Nonreversing Transaction Types** section below for additional information.

The Reversible-nonreversing transaction may be inserted in between other active activities without invoking any undo/redo actions on the surrounding activities. Once processed, it cannot be reprocessed due to the reprocessing of a previous activity. The transaction can be spawned from all transaction types and can spawn all transaction types.



Spawning using Nonreversible-Nonreversing transactions should be extremely limited as spawned transactions could exist on the policy when the original parent transaction no longer exists, causing orphaned records in the database.

Reversible-Nonreversing Transaction Types

Transactions of this type, once processed, will not cause the reversal of any activities. These transactions can however be reversed or recycled without the additional Delete checkbox privileges. A trash can and recycle icon will be available on the Action column of the Activity screen.



Financial transactions should not as a general rule be configured using the types Non-reversible-Nonreversing or Reversible-Nonreversing. If an activity of these types is used for money movement and is reversed the money movement activities processed after it will not be reversed causing potential issues with policy values.

Security for Delete Checkbox to Delete Reversible-Nonreversing Transactions

A Nonreversible-Nonreversing activity cannot be deleted unless security is configured for the Delete checkbox on the Activity screens in OIPA. A Delete checkbox will be located on the Policy, Client, Plan and Company Activity screens in OIPA. The security for this button is configured at the [PolicyActivity](#), [ClientActivity](#), [PlanActivity](#) and/or [CompanyActivity](#) levels. Users who have the proper security access will be able to check the Delete checkbox, which will grant access to the delete/recycle icons for these types of activities.



Delete Checkbox in OIPA on Policy Activity Screen

Steps to Enable Delete Checkbox on Policy, Client, Plan and Company Activity Screens

1. Navigate to **Admin Explorer | Security | Application Security | Security Groups** |*Name of Security Group* | **Company Security | Company Pages** | *Name of the Company*.
2. Check-out any of the following pages: ClientActivity, CompanyActivity, PlanActivity.
3. Click the **AllowDelete** box.
4. Check-in the file to save the changes.
5. Navigate to **Admin Explorer | Security | Application Security | Security Groups** |*Name of Security Group* | **Plan Security | Plan Pages** | *Name of the Plan*.
6. Check-out the following page: PolicyActivity.
7. Click the **AllowDelete** box.
8. Check-in the file to save the changes.

Status

Status indicates the policy status that the transaction must be in to be available to the end user. This option should be kept as active. The **EligibleTransactionsByPolicyStatus** business rule actually controls the transaction's availability based on policy status. See **EligibleTransactionsByPolicyStatus** [EligibleTransactionsByPolicyStatus](#) section of this guide for further information on the EligibleTransactionsByPolicyStatus business rule.

Processing Order

Processing Order is the order in which the transaction should run. View the order in which all transactions are processed by selecting the Time Line  button. The Time Line button is available when the transaction is first created and also on the transaction's general pane.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Transactions > Transaction Levels in OIPA

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transaction Levels in OIPA

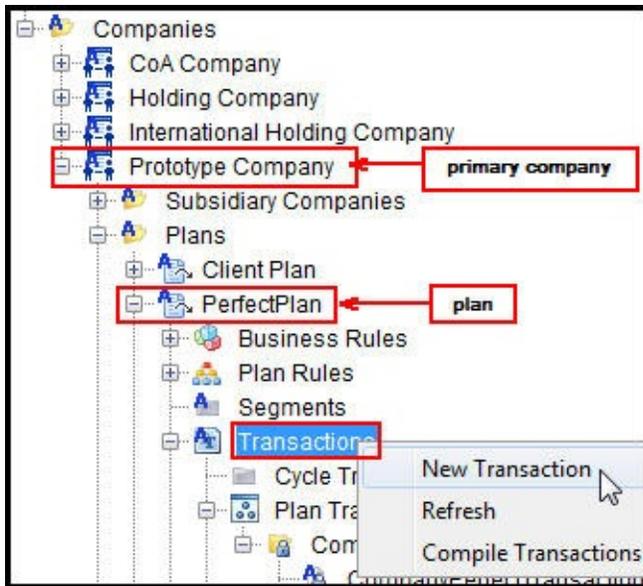
Transactions can be processed at various levels in OIPA. There are several factors that determine the level an activity can be processed at in the application. The steps required for configuring activities at the various levels are defined below.

Company-Level Transaction

These transactions process against all policies in a company. They are accessed in OIPA from the **Company** Main Menu option.

Steps to Configure a Company Level Transaction

1. Navigate to the **Main Explorer** tab.
2. Open the Primary Company folder for the company the transaction will process against.
3. Open the **Plans** folder.
4. Click the plan the transaction will belong to, or create a new plan.
5. Open the plan and right-click the **Transactions** node.
6. Click **New Transaction**.
7. Follow the [normal transaction creation steps](#), EXCEPT select **Plan-** for the transaction type.
8. Check in the transaction and add at least one translation when the translation window displays. A transaction must have at least one translation or it will not display in OIPA.
9. Navigate to the **Admin Explorer** tab.
10. Open **Security | Application Security | Security Group | *Name of Security Group* | Transaction Security | *Name of Company* | Plan**
11. Double-click the **Plan** node, under which the transaction will display.
12. Check out the transaction and add security.
13. Check in the transaction. The transaction can now be viewed in OIPA from the Company screen.



Company Level Transactions

Product-Level Transaction

Transactions defined at the product level can be shared across child products and plans under the product.

Steps to Configure a Product-Level Transaction

1. Navigate to the **Main Explorer** tab.
2. Open the primary company folder for the company the transaction will process against.
3. Open the **Subsidiary Companies** folder.
4. Open the subsidiary company folder for the subsidiary company the transaction will process against.
5. Open the **Products** folder.
6. Open the product folder for the product the transaction will process against.
7. Right-click the **Transactions** node.
8. Select **New Transaction**.
9. Follow the [normal transaction creation steps](#), selecting **Policy-** for the transaction type.

Security for product-level transactions is added at the plan level. See the [Plan-Level Transaction](#) instructions below for information on adding security for plan-level transactions.

Client-Level Transaction

These transactions process against all clients in a company. They are accessed in OIPA from the **Client** Main Menu option.

Steps to Configure a Client Level Transaction

1. Navigate to the **Main Explorer** tab.
2. Open the Primary Company folder for the company the transaction will process against.
3. Open the **Plans** folder.
4. Open the appropriate plan and right-click the **Transactions** node.
5. Click **New Transaction**.
6. Follow the [normal transaction creation steps](#), EXCEPT select **Client-** for the transaction type.
7. Check-in the transaction and add at least one translation when the translation window displays. A transaction must have at least one translation or it will not display in OIPA.
8. Navigate to the **Admin Explorer** tab.
9. *Open **Security** | **Application Security** | **Security Groups** | *Name of Security Group* | **Transaction Security** | *Name of Primary Company* | *Name of Plan*.*
10. Double-click the **Plan** node, under which the transaction will display.
11. Check out the transaction and add security.
12. Check in the transaction. The transaction can now be added in OIPA from the Client screen.

Policy-Level Transaction

These transactions process against a given policy. They are accessed in OIPA by selecting a policy and clicking **Activities** in the left navigation menu.

Steps to Configure a Client-Level Transaction

1. Navigate to the **Main Explorer** tab.
2. Open the Primary Company | Subsidiary Company folder for the company the transaction will process against.
3. Open the **Plans** folder.
4. Open the appropriate plan and right-click the **Transactions** node.
5. Click **New Transaction**.
6. Follow the [normal transaction creation steps](#), EXCEPT select **Policy-** for the transaction type.
7. Check in the transaction and add at least one translation when the translation window displays. A transaction must have at least one translation or it will not display in OIPA.
8. Navigate to the **Admin Explorer** tab.
9. *Open **Security** | **Application Security** | **Security Groups** | *Name of Security Group* | **Transaction Security** | *Name of Subsidiary Company* | *Name of Plan*.*
10. Double-click the **Plan** node and the transaction will display.
11. Check out the transaction, assign it a state context and add security.
12. Check in the transaction. The transaction can now be added in OIPA from the Activities screen.

Plan-Level Transaction

These transactions process against a given plan. They are accessed in OIPA by clicking the **Plan** Main Menu option and selecting **Plan Activities**.

Steps to Configure a Client-Level Transaction

1. Navigate to the **Main Explorer** tab.
2. Open the Primary Company | Subsidiary Company folder for the company the transaction will process against.
3. Open the **Plans** folder.
4. Open the appropriate plan and right-click the **Transactions** node.
5. Click **New Transaction**.
6. Follow the [normal transaction creation steps](#), EXCEPT select **Plan-** for the transaction type.
7. Check in the transaction and add at least one translation when the translation window displays. A transaction must have at least one translation or it will not display in OIPA.
8. Navigate to the **Admin Explorer** tab.
9. *Open **Security** | **Application Security** | **Security Groups** | Name of Security Group | **Transaction Security** | Name of Subsidiary Company | Name of Plan.*
10. Double-click the **Plan** node and the transaction will display.
11. Check out the transaction and add security.
12. Check in the transaction. The transaction can now be added in OIPA from the Plan Activities screen.

Relationship-Level Transaction

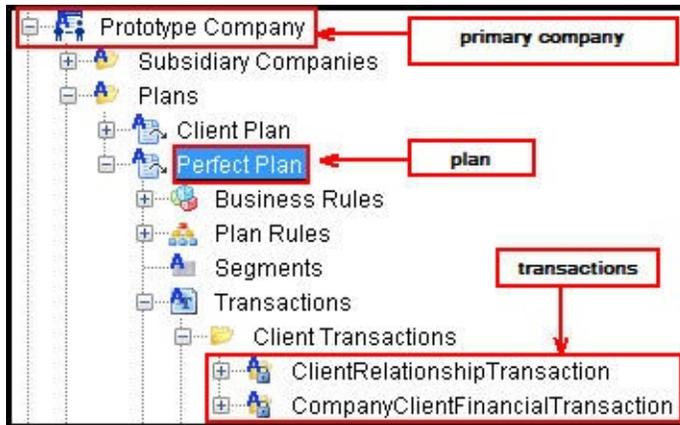
These transactions are processed against client relationships that are identified on the Group Customer screen. They are accessed in OIPA when a user clicks **Relationship** from the Group Customer Left Navigation menu and then right-clicks on a Secondary Relationship and selects **Go to Activities**.

Steps to Configure a Relationship Level Transaction

1. Navigate to the **Main Explorer** tab.
2. Open the Primary Company folder for the company the transaction will process against.
3. Open the **Plans** folder.
4. Click the plan the transaction will belong to or create a new plan.
5. Open the plan and right-click the **Transactions** node.
6. Click **New Transaction**.
7. Follow the [normal transaction creation steps](#), EXCEPT select **Client-Relationship** for the transaction type.
8. Check-in the transaction and add at least one translation when the translation window displays. A transaction must have at least one translation or it will not display in OIPA.
9. Navigate to the **Admin Explorer** tab.
10. Open **Security | Application Security | Security Group | *Name of Security Group* | Transaction Security | *Name of Company* | Plan**.
11. Double-click the **Plan** node and the transaction will display.
12. Check-out the transaction and add security.
13. Check-in the transaction. The transaction can now be viewed in OIPA from the Company screen.

How To View Client Relationship Transactions on Customer Screen in OIPA

Each user logs into OIPA with a username and password that is associated to a company. The plans and transactions created on that company will be available in OIPA.



Company, Plan and Transaction Association

A ClientScreen override must exist for this company. To point to specific ClientRelationship transactions, the GroupCustomer client type must be updated to point to a specific plan (RELATIONSHIPACTIVITYPLAN attribute) on the company where the transactions reside.

```
<Client TYPECODE="20" ACTIVITYPLAN="CustomerPlan" RELATIONSHIPACTIVITYPLAN="ClientRelationship" CLIENTTYPECODE="GROUPCUST">
```

ClientScreen Rule Group Customer Client Type Plan Association

All Client-Relationship transactions created on this plan will display when **GoTo Activities** is clicked from a Secondary Relationship.

All other Client- transactions created on this plan will display when **Add Activity** is clicked from the Customer screen.

You are here: Configuration > Transactions > Create Transactions

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create a Transaction

New transactions can only be created from the Main Explorer tab. There are three important points to keep in mind when creating transactions. First, new transactions must be assigned [security](#) or they will not be available in the OIPA application.

Also, transactions will not be displayed in the Activity drop down list in OIPA unless they have been added to the

EligibleTransactionsByPolicyStatus

[EligibleTransactionsByPolicyStatus](#) plan rule. This rule determines what status a policy must be in before the transaction can be applied to the policy.

And finally, a [translation](#) must also be assigned to the transaction. This is done through the Translation window that appears each time you check-in the transaction.



If the transaction being created runs valuation, refer to the [valuation section](#) for important configuration information.

Steps to Create a New Transaction

Transactions can be created at many different levels. Before creating a transaction, make sure you know the level (policy, plan, client, company) and the steps necessary to create it at that level. Refer to [Transaction Levels in OIPA](#) for an explanation of the various transaction levels and the corresponding creation points.

1. Expand the appropriate company folder in the **Main Explorer** window. Refer to the link above for transaction levels and where they are created.
2. Expand the appropriate plan folder.
3. Right-click the **Transaction** folder.
4. Select **New Transaction** and the wizard will open.
5. Enter a name for the transaction. The transaction name should not include spaces.



Transaction names are not case sensitive and the name can only be used one time. The application will not reuse a transaction name even if the case is altered.

6. Enter the following items: (Refer to [Transaction Type and Processing Order](#) for further information.)
 - a. Select the [transaction type](#).
 - A plan type transaction, such as Plan-Financial, when processed will apply to all policies in that plan. The plan type transaction will reside in the Plan Transactions folder under **Transactions** in the Main Explorer.
 - A policy type transaction, such as Policy-Financial, when processed will apply only to the policy it is added to. The policy type transaction will reside in the Policy Transactions folder under **Transactions** in the Main Explorer.
 - Client transactions when processed will apply to the specific client and will reside in the **Company | Plans** folder in the Main Explorer.
 - [Client-Relationship](#) transactions only apply to primary and

secondary relationship types associated with the Customer that the transaction is being processed against. These relationship types are defined in the transaction in step 2 of the wizard.

- o Cycle transactions are processed in a batch during scheduled cycle processing. They will reside in the Cycle Transactions folder under **Transactions** in the Main Explorer.
 - o Intake transactions...
- b. Enter the processing order for the transaction.
 - c. Select **Yes** in the NAV Correction drop down box if the transaction is a financial transaction.
 - d. Click **Next**.

 If a new Client-Import transaction needs to be added, create a new transaction from the Client Plan folder. All Client related transactions must be created on the Client level.



Step 2 of New Transaction Wizard

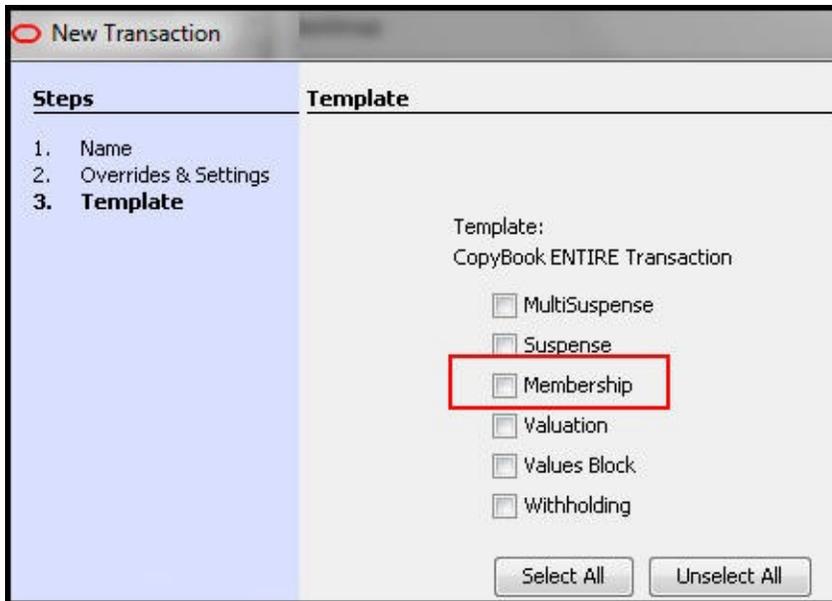
7. Select the configuration sections needed to create the transaction. Use an entire CopyBook or select only the configuration sections

needed to create the transaction. A sample configuration of the sections selected will be given as a starting point when the transaction's XML file is first opened. All panes are available for configuration regardless of whether the pane is checked or not. Not checking a pane simply means that the pane will be a blank slate.

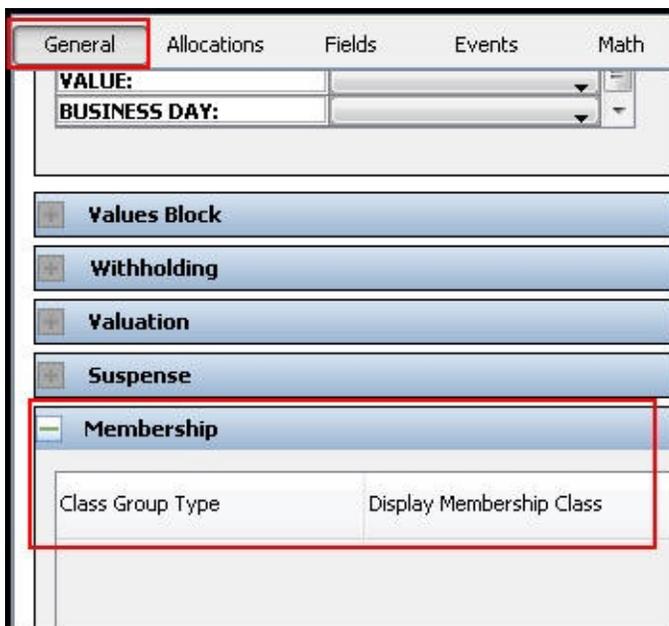
- a. If an entire Copybook is selected, a Copybook Lookup window will appear. Select the Copybook needed.
- b. If a partial Copybook is needed, click **No**. Select the individual tags from the list. The tags that are selected will be added to the transaction's general pane.
- c. If sections are selected a sample of the configuration will be given as a starting point.
- d. All panes are available for configuration regardless of whether the pane is checked.
- e. Not checking a pane simply means that the pane will be a blank slate.

A membership option in the Template section allows a Membership section to appear in the **General Pane** [General Pane](#) of the transaction once it is created.

If this section is added, then at least one membership class must be added to the Membership section before the transaction can be checked-in.



Transaction Wizard with Membership Highlighted



Transaction General with Membership Section



The Transitions feature is only available for Policy-Financial transactions.

8. Click **Finish**. The new transaction is created and displayed in blue (not checked-in).

9. Check-in the transaction. A translation window will appear.
10. Enter the correct [translation](#) for the transaction name for each language and click **OK**. If a locale value is left empty, then this window will appear every time the transaction is checked-in. At least one translation value **MUST** be given.



Let the Security Manager know when a new transaction is created. The Security Manager will have to add [security](#) to the transaction so that the appropriate users will be able to see it in OIPA.



Make sure all policy level transactions are in the EligibleTransactionsByPolicyStatus plan rule. This file controls when a transaction is available for use as an activity in OIPA.

Every new transaction automatically overrides and attaches the [TransactionCosmetics](#) and [TransactionBusinessRulePacket](#) business rule. **Don't** manually override and attach these business rules. These two rules are copied from the existing global rules. They are mandatory for the transaction to process and cannot be deleted from the transaction; however, the information may be edited.

You are here: Configuration > Transactions > Configure Transactions

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Configure Transactions

There are three important points to keep in mind when configuring or creating transactions. First, new transactions must be assigned [security](#) or they will not be available in the OIPA application.

Also, transactions will not be displayed in the Activity drop down list in OIPA unless they have been added to the [EligibleTransactionsByPolicyStatus](#) plan rule. This rule determines what status a policy must be in before the transaction can be applied to the policy.

And finally, a translation must also be assigned to the transaction. This is done through the Translation window that appears each time you check-in the transaction.

If the transaction being configured runs valuation, refer to the [valuation section](#) for important configuration information.

Steps to Configure a Transaction

1. Open the **Transaction** folder.
2. Right-click on the transaction's XML file.
3. Select **Check-out**.
4. Refer to [General](#), [Allocations](#), [Address, Fields](#), [Events](#), [Math](#), [Assignment](#), [Spawn](#), [XML Source](#) or [Debug](#) for information on how to configure.
5. Check-In when finished.
6. Open the application and test the configuration.



Make sure the transaction is in the EligibleTransactionsByPolicyStatus plan rule. This file controls when a transaction is available for use as an activity in OIPA.

You are here: Configuration > Segments > Compile a Transaction

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Compile a Transaction or Segment

Compilation is limited to transactions, segment name rules and calculate general rules. When any of these three types of configuration are compiled, configuration errors can be identified in math. The Rules Palette marks folders and files with a red  icon to easily identify errors. The Compile Transaction feature can be run on an individual transaction or segment or on all of the transactions or segments in a plan. After running Compile, the  icon will appear on folders that contain a transaction or segment with an error. The warning  icon will display if there is a warning. Check-out the transaction or segment's XML file with the error and open the Engine Error Output window to view the explanation of the compilation error. If this window is not visible, click **Window** on the Main Menu and select it.



If the underlying rule has a context that allows state overrides, then the Rules Palette will prompt the user to select a state value from a Context pop-up window.

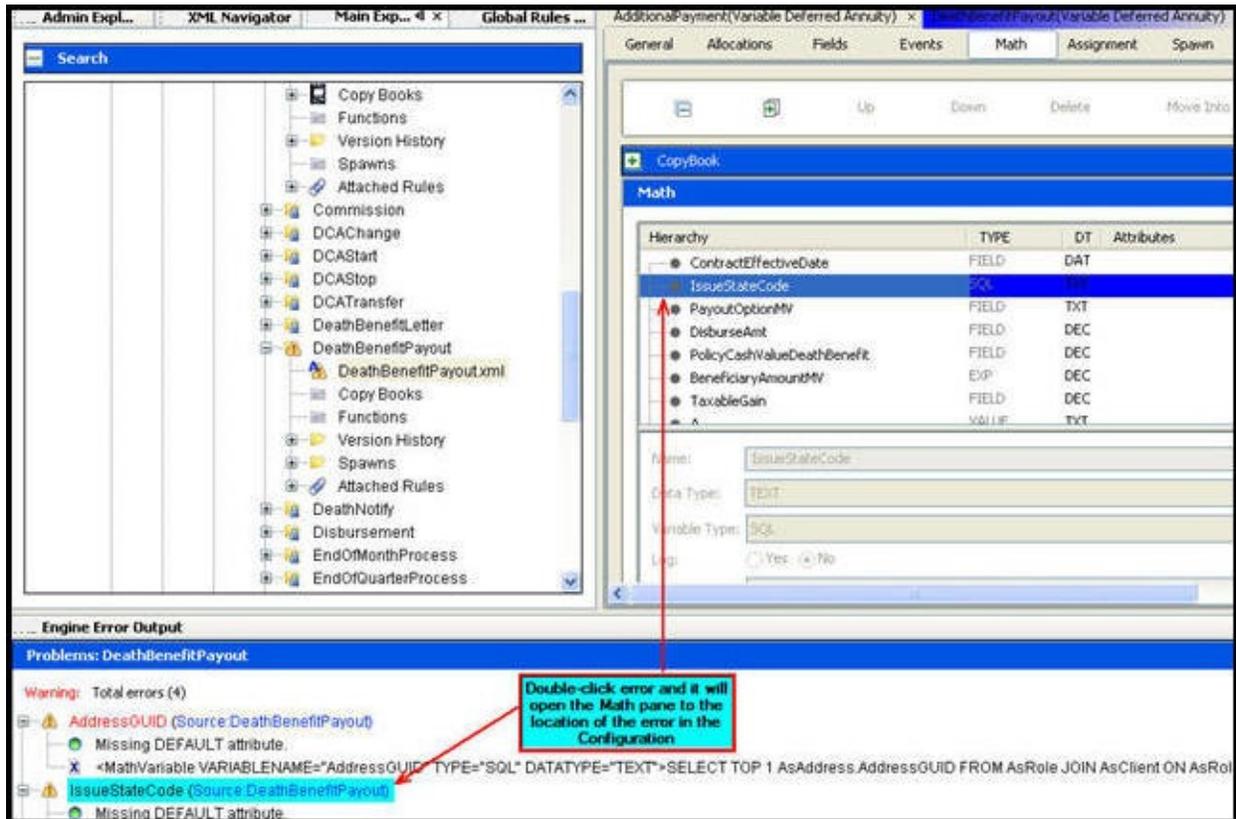
The [XML Navigator](#) window will automatically open when a transaction with errors is checked out. It is **necessary** that this window remain open, as it connects the error output to its location in the configuration.

There are three types of errors that are listed in the **Engine Error Output** window.

1. **Translation Error** is an error that was caused by a syntax error in the math section. All translation errors will be listed at one time.
2. **Compilation Error** is an error with the configuration in the math section. In this situation, the compile feature stops compiling the configuration of a transaction after it locates this type of error. Only one compilation error at a time is handled. After resolving the compilation error and selecting the **Save** button  , the compiler runs again through the configuration and locates the next

compilation error. This process repeats until all errors have been resolved.

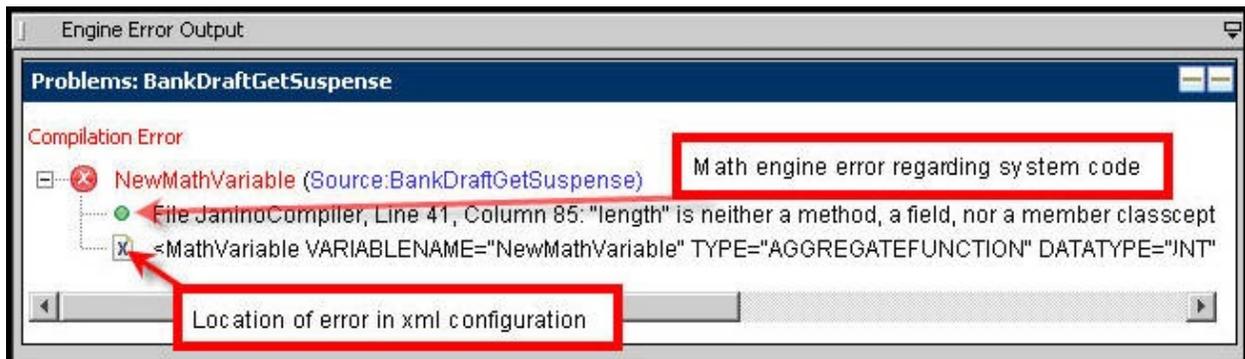
3. **Runtime Error** is a system code error that was caused by configuration.



Errors Shown in Engine Error Output Window

Steps to Compile all Transactions or all Segments Simultaneously

1. Right-click on the Transaction or Segment folder.
2. Select **Compile Transactions**.
3. Navigate in folders with the  icon until the transaction or segment's XML file is located.
4. **Check-out** the XML file.
5. Review the error in the **Engine Error Output** window.
6. Double-click on the configuration error message  to be brought to the error in the configuration.
7. Resolve the error.
8. Select the **Save** button  .
9. Repeat steps 5 through 8 until no errors are reported.



Compilation Error Shown in Engine Error Output Window

Steps to Compile an Individual Transaction or Segment

1. **Check-out** the transaction or segment's XML file.
2. Review the error in the **Engine Error Output** window.
3. Double-click on the configuration error message  to be brought to the error in the configuration.
4. Resolve error.
5. Select the **Save** button  .
6. Repeat steps 2 through 5 until no errors are reported. Please see the [Debug Pane](#) for correcting errors that do not pertain to compiling.



If the **Engine Error Output** window is closed, it can be opened from **Window** on the Main Menu by selecting  .

You are here: Configuration > Transactions > Clone a Transaction

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

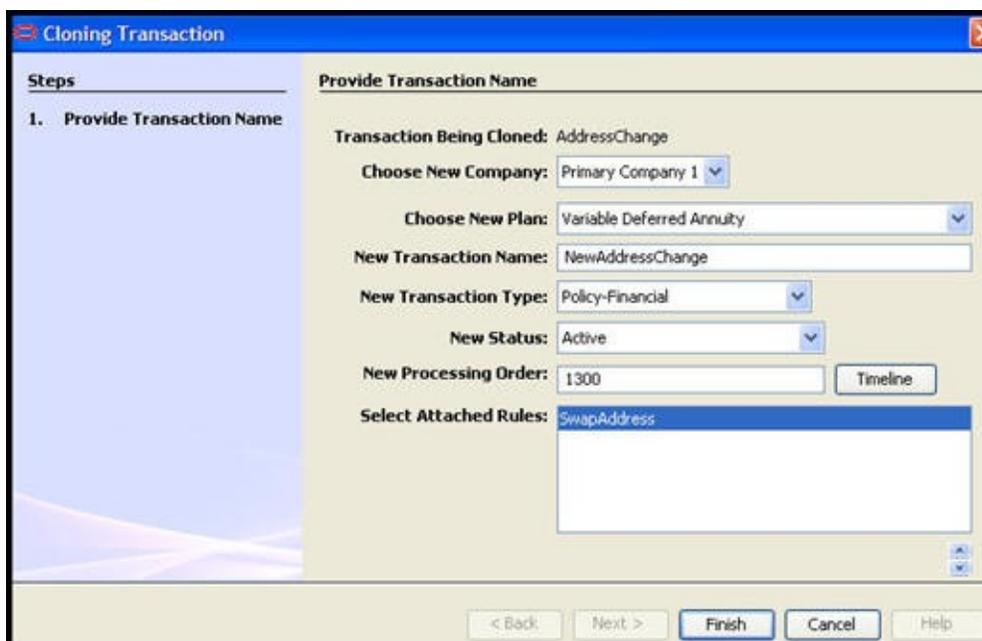
Clone a Transaction

In Rules Palette, a transaction can easily be cloned. There is no need to cut, paste and re-attach business rules in order to clone a transaction to each plan that uses it. Instead, simply clone transactions across companies and plans, as well as into the current plan in which the transaction exists. This eliminates errors and saves time. All business rules, including [TransactionBusinessRulePacket and Transaction Cosmetics](#) , will also be attached to any transaction cloned.

Steps to Clone a Transaction

1. Expand the transaction folder.
2. Right-click on the transaction.
3. Select **Clone Transaction**. (This option is only available when the transaction is not checked-out.)
4. In the Cloning Transaction Wizard select:
 - a. New company
 - b. New plan
 - c. Name of the cloned transaction
 - d. [Transaction Type](#) if applicable
 - e. [Status](#) as active
 - f. [Processing Order](#) for the transaction if applicable
 - g. Deselect any [attached rules](#) that should not be cloned with the transaction.
 - h. Select **Finish**.

After the transaction is successfully cloned, navigate to its location. Configure the transaction as needed.



The screenshot shows the 'Cloning Transaction' wizard dialog box. The title bar reads 'Cloning Transaction'. On the left, a 'Steps' pane shows '1. Provide Transaction Name'. The main area is titled 'Provide Transaction Name' and contains the following fields:

- Transaction Being Cloned: AddressChange
- Choose New Company: Primary Company 1 (dropdown)
- Choose New Plan: Variable Deferred Annuity (dropdown)
- New Transaction Name: NewAddressChange (text input)
- New Transaction Type: Policy-Financial (dropdown)
- New Status: Active (dropdown)
- New Processing Order: 1300 (text input) with a 'Timeline' button
- Select Attached Rules: SwapAddress (list box)

At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Wizard to Clone a Transaction

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Transactions > Delete a Transaction

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Delete a Transaction

There are several things to consider before deleting a transaction. There are two situations when a transaction cannot be deleted.

1. If the transaction has been applied to a policy in OIPA as an activity, regardless of status, it will not be deleted.
2. If the transaction is specified in the Spawn section of another transaction it will not be deleted.

If Release Management functionality is turned on, that will also affect whether or not a transaction can be deleted. In Release Management, when a transaction is deleted it must be added to a Configuration Package. Once the package is promoted and deployed into the target environment, the transaction can be deleted as long as activities in OIPA are not associated with it. If there are pending or active activities associated with the transaction, an error message will display listing activities associated with that transaction and the configuration package will not be deployed. Delete the activities that are associated with the transaction and then re-deploy the configuration package.

Make sure when a transaction is deleted that it is removed from all possible locations where it could have been attached. Check the following locations:

- [EligibleTransactionsByPolicyStatus](#) business rule. If it was added to this business rule then remove it.
- [Data Dictionary](#). Remove entries in the Data Dictionary that refer to the transaction. This will keep the Data Dictionary clean and current.
- [OIPA Security](#). Remove entries from Security that refer to the transaction. Check-out and then check-in each user under **OIPA Security | Users**. The system will only pull current transactions so ones that have been deleted will not appear. This is not mandatory, but will keep the security tables up-to-date.
- [Global Rules Explorer](#). Check the Transactions folder. Type the transaction name in the Search field to pull up the transaction and

any overrides. Make sure to delete them all.

Steps to Delete a Transaction

1. Navigate to the Global Rules Explorer and type the transaction name into the Search field at the top of the screen. The search results will show all the overrides that exist for that transaction. Make a note of the overrides so that they are all deleted. .



Search for Transaction on Main Explorer Tab

2. Navigate to the Main Explorer and open the **Company | Plan** folder where the transaction resides.
3. Scroll through the list of transactions to find the one needed.
4. Double-click the transaction. This will open the folder and reveal the transaction XML file.
5. Right-click the transaction XML file and select **Delete Rule**.

Note: If Release Management is turned on, a window will appear asking for the deleted transaction to be added to a Configuration Package. Select a Configuration Package and click **OK**.

6. Click **Yes** when the delete confirmation message appears. The transaction will be deleted.



Delete Confirmation Message

7. Navigate to the Business Rule folder for the Company and Plan where the deleted transaction resided.
8. Double-click the Plan Rules folder and double-click the [EligibleTransactionsByPolicyStatus](#) business rule.
9. Click the EligibleTxns tab and scroll through the list of transactions to the one deleted.
10. Click the transaction. It will appear highlighted in blue.
11. Click **Remove**.
12. **Check-in** the file to save the changes to the database.

Note: If there are other plan overrides for the transaction, make sure to open each plan folder and delete the transaction. Make sure of the location of all plan overrides by performing a search in the Global Explorer for the transaction. All plan overrides will come back in the search results.

Delete Data Dictionary Entry

Once a transaction is deleted, click the Data Dictionary icon on the tool bar. When the Data Dictionary opens, search for the transaction. Highlight the transaction in the search result window and click Delete to remove it from the Data Dictionary.

Update Security

Open the Admin Explorer tab and navigate to **Security | OIPA Security | Users**. Check-out each user file and then check it back in. This will update security and remove the deleted transaction information.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Transactions > Edit Transaction General

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Edit Transaction General

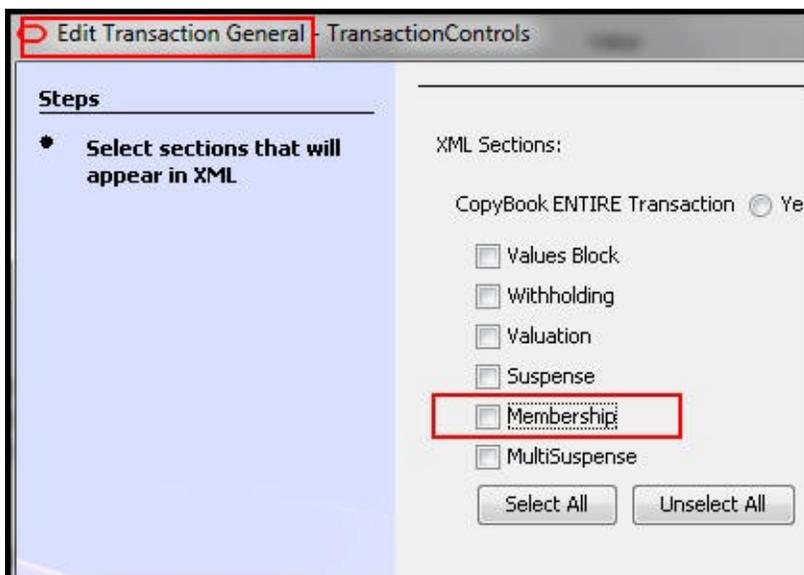
Once a transaction is created, the only way to remove a section from the General Pane is to use the right-click option **Edit Transaction General**. This option is only available if the transaction is checked-out.

Refer to [Transaction General Pane](#) for complete information on all General Pane sections.



Edit Transaction General Option in Main Explorer

When the Edit Transaction General option is selected, a window opens with a list of all Transaction General sections. If a section is selected, it will be removed from the General pane and all information in that section will be removed from the database.



Edit Transaction General Window with Membership Section

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Transactions > MultiField Configuration

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

MultiField Configuration

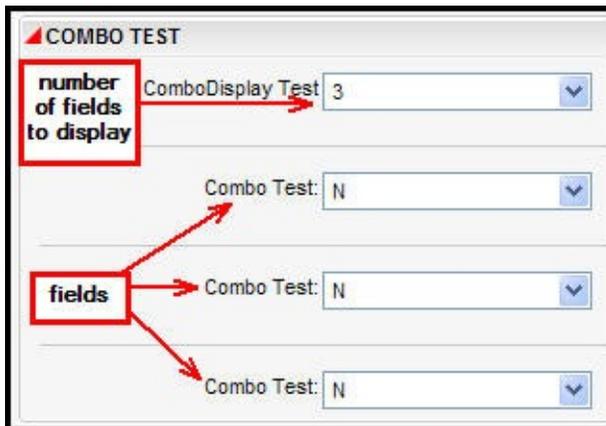
Multifields configuration allows users to select a number from a configured drop down box on the screen, which determines the number of subsequent fields that should be displayed.

Steps to Configure Multi Fields

1. In the Global Explorer, right-click on the System Node and select New Global Rule for category System Rule.
2. For best practice, name the rule starting with "MultiField". This also indicates to the system to put the initial <MultiField> elements in the rule.
3. Select **System** from the TypeCode drop-down box.
4. Configure in the XML Pane of the new rule. You can find the syntax in the XML Configuration Guide. This business rule is like a copybook, where it can be called from various transactions or screens and there may be many MultiField business rules.
5. In a transaction or screen, you can then configure MultiField elements that reference a MultiField business rule.

Prototype Scenario

In OIPA, the user navigates to a screen that has configured Multifields. When the activity window opens, the first combo box allows the user to select the number of fields to display. The user selects three as the number of fields to display, and then three separate fields appear below. Sets of fields can also be configured to display multiple times.



Multifields in OIPA

Prototype Configuration

In order to understand this prototype, you need to look at the following two pieces of configuration:

- The Multifields business rule called MultiField-MultiFiedPrototype. Notice the name of the business rule started with Multifields.
- The MultiFieldPrototype transaction, which has a MultiField element that calls the specific MultiField business rule called MultiField-MultiFiedPrototype. Also the value of the MultiField element must be set to yes.

```
<MultiFields RULE="MultiField-MultiFieldPrototype">Yes</MultiFields>
```

Multifields Element in a Transaction

Locating the Prototype Samples

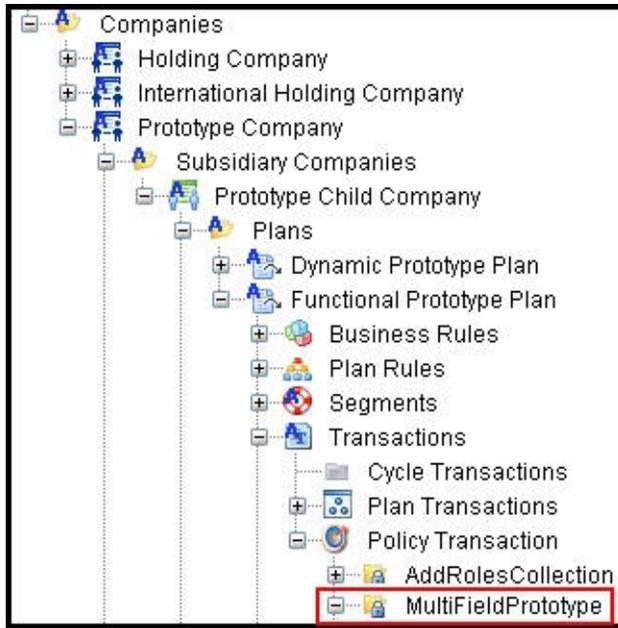
The Prototype Company in the Rules Palette contains two sample configurations for Multifields.

- Multifields business rule: This business rule defines the main combo box where the number of multifields is determined. Navigate through the following folders in the Global Rules Explorer to locate the configuration sample: **Business Rules | Screen | MultiField-MultiFieldPrototype**.



Multifields Business Rule in Global Explorer

- Multifields transaction: This transaction demonstrates how to call a multifield business rule. Navigate through the following folders in the Main Explorer to locate the configuration sample: **Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Functional Prototype Plan | Transactions | Policy Transactions | MultiFieldPrototype**.



Multifields Transaction in Main Explorer

You are here: Configuration > Transactions > Attach Rules to a Transaction

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Attach Rules to a Transaction

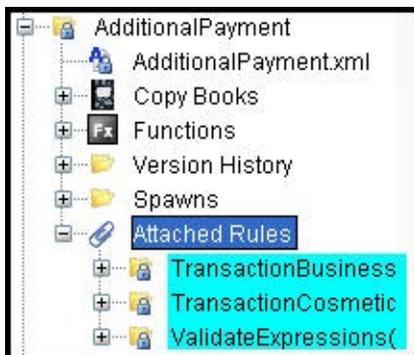
Rules can be attached to transactions to enhance and support processing. Global processing rules can be used by [overriding](#) and attaching them to a transaction. The attached rule is an override at the transaction level of the global rule and the configuration is a copy of what is in the global rule. The attached overridden rule can be edited in order to make the functionality specific to the transaction's requirements.

Once the overridden version of the rule is complete, it is attached to the transaction to support processing. Attaching rules adds them to the transaction's **Attached Rules** folder and also automatically adds the rules to the **TransactionBusinessRulePacket**. The order of the attached business rules in the **TransactionBusinessRulePacket** should be determined by the user according to configuration standards.

Each attached rule's folder can be expanded  to check-out its XML file and configure the override appropriately.

The Rules Palette always attaches the [TransactionCosmetics](#) and [TransactionBusinessRulePacket](#) business rules for each transaction. They do not need to be manually attached.

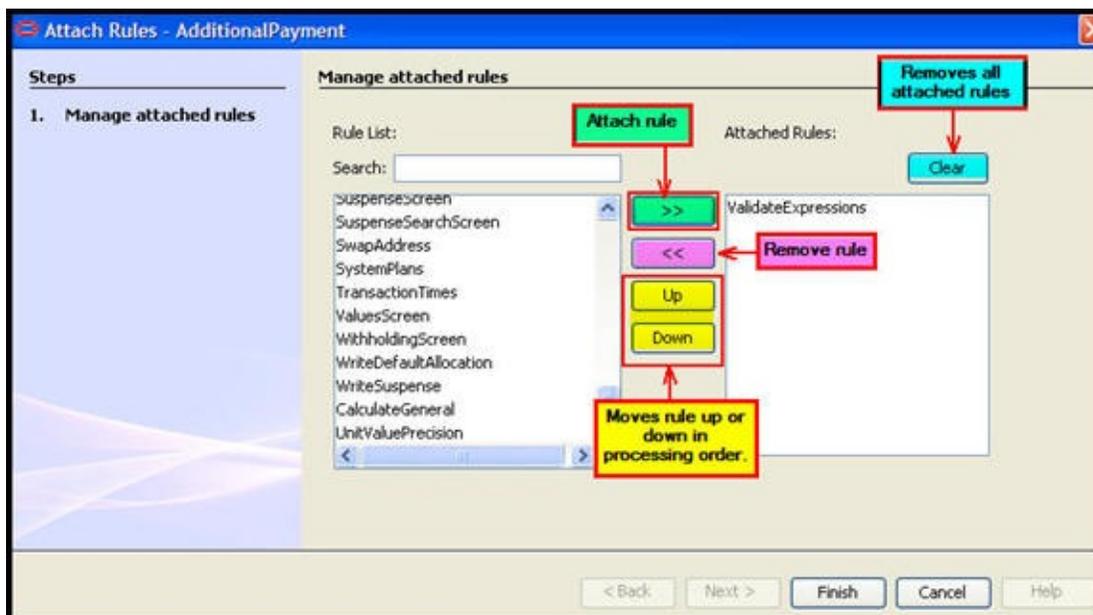
Attached rules can be [removed](#) from a transaction by right-clicking on the Attached rules node and selecting the option **Edit Attached Rules**.



Attached Rules Listed in Main Explorer

Steps to Attach a Rule

1. Expand  the associated transaction folder in the **Main Explorer**.
2. Right-click on the transaction's XML file.
3. Select **Edit Attached Rules**. The rule must be checked-in to see this option.
4. Select the rule to attach from the Rule List.
5. Use the  button to attach the rule.
6. Repeat as necessary.
7. Select **Finish**.



Attach Rules Window

Steps to Edit Attached Rules

1. Expand  the associated transaction folder in the **Main Explorer**.
2. Right-click on the transaction's XML file.
3. Select **Edit Attached Rules**. The rule must be checked-in to see this option.
4. Select the rule to remove from the Rule List.
5. Use the  button to remove a rule.
6. Repeat as necessary.
7. Select **Finish**. The Attached Rules node may disappear from the navigation tree. If this happens, logout and log back into the Rules Palette to refresh the navigation tree.

You are here: Configuration > Transactions > Transactions That Process Projections

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transactions That Process Projections

Occasionally insurance companies need to run projections that mimic a policy's behavior as if it were processed on a day-to-day basis using normal processing. When projections are run, the logic typically uses a subset of the same calculations, retrievals, business processing logic, and updates as those used during normal lifecycle processing.

Valuation requires some additional consideration. When valuation is needed, a starting value based on the funds is used, and then further crediting is handled through a fixed method (either a single field or a string of values that is pre-determined.) The same rules will be processed for both the projection and regular processing. In this way, the system ensures that the end results are the same for calculations, costs and updates when the input parameters are the same.

Typical situations where projections would be used include annual statement reporting and Illustrations, including new business and proposals.

Scenario

An annual statement for a Life Insurance policy needs to project from the current date to the maturity date of the policy. All current policy information is used. If there are any loans on the policy, these carry forward (without additional loans), and reflect riders/benefits dropping off of the policy at the specified times. This projection uses all of the calculations and processing as with normal "in force" policy processing. There are two projections made: one for "with premiums paid" as reflected in the "anticipated annual premiums" on the policy information, and the second is with "no premiums" paid. The values that are returned are the Projected End Dates for both with and without premium.

How Projections Work in OIPA

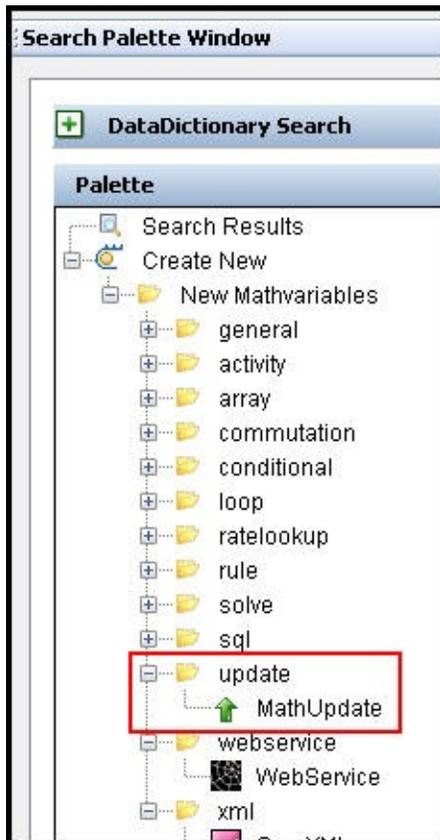
OIPA sets into memory the beginning values for each Instance Data Element, which reflects the current information in the Database. This can be performed multiple times using a single activity. Fields that are updated throughout the projection processing that also need to retain their original (current) values are loaded via math variables or a collection before the rest of the math commences.

OIPA checks to see if a value for a field exists in memory before looking to the database to retrieve it. The **Table name : Field name** (e.g., Policy:FieldName) syntax does this as well as the POLICYFIELD and SEGMENTFIELD math variable types. OIPA always returns the latest value in memory first.

Only the values in memory are updated: the actual database table value is not changed during the projection processing. It is possible for a field's value in memory to change any number of times throughout the activity processing.

Overview of Configuration

The Math pane of a transaction is used to configure projection processing. The Palette window contains a category named **Update**. The **MathUpdate** element can be dragged and dropped into the Math pane. This element is used to hold the math variable (or literal numeric value) that is copied into the specific Table field in memory. The **FIELDNAME** attribute holds the name of the Table field to update in memory. The **TYPE** attribute identifies the source table of the record type to update. SegmentFieldCollection will update the named Segment field records with the collection values. Both of these attributes are required. An optional **SEGMENTGUID** attribute is required only if TYPE equals SegmentField. It identifies the segment record to update.



MathUpdate Element in Palette Window

You are here: Configuration > Transactions > [Activity Processing](#) > Activity Processing Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Activity Processing Overview

Activity processing is performed on the [Activity screen](#) in OIPA. Some activity processing can occur on other screens, but typically the Activity screen houses the activity processing functionality. There are several ways to process an activity. An activity can be manually processed by selecting the lightning bolt ⚡ icon next to an activity. Activities can also be generated by the system, [reversed](#) or [deleted](#). Activities can be processed at the plan or policy level.

There are two main components of an activity: [activity type](#) and [activity status](#). Activity type defines how the activity was created (generated by the user or by the system). The activity status defines the current state of the activity (active, pending, etc.).

OIPA controls activity processing through a [timestamp](#). This prevents users from accessing an activity that is being processed.

Automatic Processing in OIPA

To use the Auto Process option in OIPA, the AutoProcess business rule must be configured to **Yes**. This business rule dictates whether activities are to be processed as soon as they are added or when manually processed by the user.

If set to **Yes**, then the AutoProcess button will be checked on the Activity screen. The rule is overridden to affect the various Activity screens. The Policy Activity screen is affected by overriding the rule at the plan level. The Plan Activity screen is affected by overriding the rule with the plan that holds plan level transactions. The Client Activity screen is affected by overriding the rule with the primary company.

You are here: Configuration > Transactions > [Activity Processing](#) > Activity Screen in OIPA

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

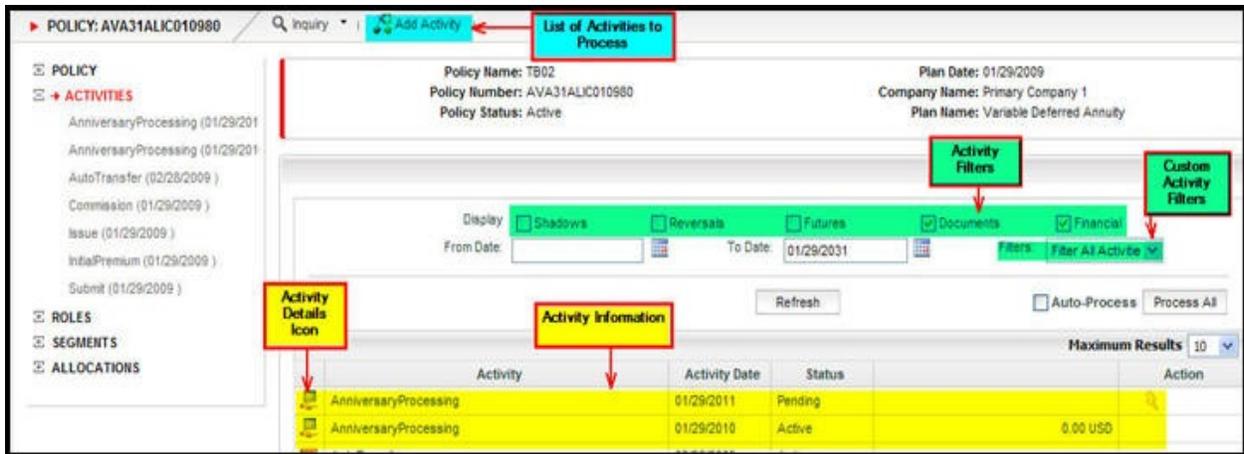
Activity Screen in OIPA

The Activity screen is located in the Oracle Insurance Policy Administration (OIPA) system. The Activity screen provides a list of activities and their associated data. A filter option displays activities based on their [type](#) and/or [status](#). This screen also contains the processing options that are available for processing activities. Activities can be processed at the company, client, plan or policy level.

Activities can be manually added on the screen by selecting the **Add Activity** button on the secondary menu. A window will open with an Activity drop-down box and an Activity Details section. Select the activity to process from the drop-down box and then enter the relevant activity details. Select the **OK** button to add the activity to the Activity screen. A lightning bolt icon will display to the right of the activity. Click the lightning bolt to process the activity.

The [EligibleTransactionsByPolicyStatus](#) rule defines the activities that are applicable to the policy based on policy status. The Add Activity drop down list will present only those activities defined for that status.

The Activities screen displays activities added to the policy. The view of the activities can be modified by activity status and date filters. If the [ActivityShading feature](#) is enabled at the company level, then the activity list will display each activity with colored text that matches the activity's status.



Activity Screen in OIPA

Activity Levels

Activities can occur at different levels in the OIPA system. Company, client, plan and policy level activities are supported.

Access company level activities from the Company option on the Main Menu. The **Add Activity** link on the Company Secondary menu displays company level activities.



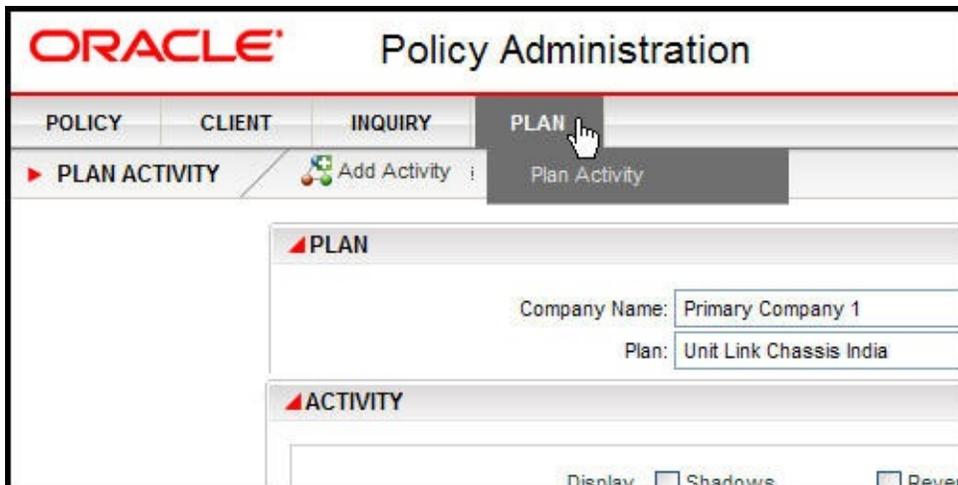
Company Menu in OIPA

Access client level activities from the Client option on the Main Menu. First search for the client record. When the record is open on the screen, an Add Activity link will display on the Secondary menu.



Client Add Activity Link in OIPA

Access plan level activities from the Plan option on the Main Menu. The **Add Activity** link on the Plan Secondary menu displays plan level activities.



Plan Link on Main Menu in OIPA

Access policy level activities from the Policy Secondary menu. Open a policy and then the **Add Activity** link will appear on the Secondary menu. In order for an activity to be added, it must first be configured in the Rules Palette.



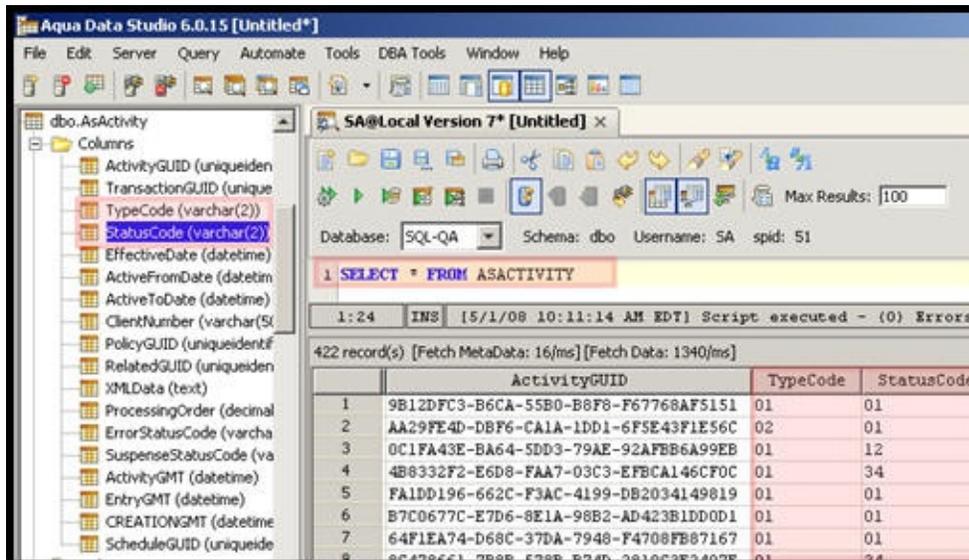
Add Activity Link on Policy Secondary Menu in OIPA

You are here: Configuration > Transactions > [Activity Processing](#) > Activity Records

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Activity Records

Activity records are written to the AsActivity table. The [type](#) of activity is tracked in the TypeCode column and the [status](#) in the StatusCode column.



Activity Records in the Database

There are two instances of historical data that are stored with processed activities:

- The Client Number, which is the ID of the user who processed the activity last.
- Actual financials and money exchange, if any.

You are here: Configuration > Transactions > [Activity Processing](#) > Activity Types

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Activity Types

The types of activities performed in OIPA are identified in AsCodeActivityType. These types should not be confused with the status of an activity, but used in conjunction with them, to understand how an activity was generated and what status the activity is currently in. Activities can be generated by an end user or the system may automatically generate activities when a dependent activity's data is changed.

To verify an activity's type run a query against the AsActivity table. The TypeCode column of this table tracks an activity's type.

Activity Type	TypeCode	Definition
Natural	01	An activity entered manually by a user or an activity that was spawned for the first time from a natural activity. A spawned activity can be considered a natural activity even if it was system generated because the user manually processed the activity that spawned it.
Reversal	02	Reversal activity that was created by an end user either manually deleting or recycling an activity. A spawned activity that was reversed because the originating activity was manually reversed.
Undo	03	This is a system generated reversal that was automatically performed by the system because an activity it depended on was changed. A Redo activity usually is generated with an Undo.
Redo	04	This is a system generated activity that was automatically created due to the generation of an Undo activity.
Deleted	05	Should not be in a production environment.

You are here: Configuration > Transactions > [Activity Processing](#) > Activity Statuses

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Activity Statuses

The following terms define the most common statuses an activity can have in OIPA. They indicate at the activity level the status of that activity record. The activity status, with the date stamp in current and history records, identifies the significant point of processing and provides internal control for activities. A complete list of statuses can be found by querying the AsCode table for AsCodeStatus.

Activity Status	Status Code (AsCodeStatus Table)	Processing Detail
Active	01	Current data that has completed activity processing and math calculation. This includes the following XML activities: <ul style="list-style-type: none"> ▪ Develops XML process ▪ Changes to XML and table changes ▪ Inserts XML and writes to table
Pending	02	Information entered but not processed. Pending data requires action before it can be applied or calculated. All required data must be entered and the lightning bolt selected to change the status from pending to active.
Shadowed	12	An activity that has been reversed. In OIPA, the term reverse refers to an activity that has been changed to shadowed by a reversal activity. All updated data is set back to the original state before the shadowed activity was run. A reversal activity could be performed either by recycling or deleting an activity record.
Pending Shadowed	34	An activity with data that was entered but never processed and then deleted.
NUV Pending	13	No Unit Value for the system date but there is a Unit Value for the effective date of a transaction. Processes all activity values except for the gain amount. Active - NUVs needed to calculate units and gain/loss.
Gain/Loss Pending	14	Active - NUVs needed to calculate gain/loss.
Queued	58	Activity waits to process until all prior activities that share allocation funds with the activity move out of NUV Pending status.

Activity Status Colors with Activity Shading Enabled

The Activity Shading feature enables the display of activity status in the [Activity screen](#) in color. If the company is configured with the [ActivityShading](#) feature enabled (in a company override of the CompanyCosmetics rule), then the activity list will display each activity with colored text. The color of the text in the activity listing matches the activity's status.

- **Black** — Active, Active (Backdated activity)
- **Green** — Pending, Pending Ready, Requirement Pending, Queued, NUV Pending, NUV Pending (Backdated activity), Gain/Loss Pending
- **Red** — **(With Error)** Pending, Pending Ready, Requirement Pending, Queued, NUV Pending, Gain/Loss Pending
- **Gray** — Shadow, Pending Shadow, Active (Backward processing)
- **Purple** — Asynchronous Processing Wait, Asynchronous Processing Stop

Active Status

An activity is in active status when the math calculations process successfully. Any resulting data is stored in the database. The activity also displays an active status on the Activity screen in OIPA. Activities can be manually processed when a user selects the lightning bolt icon next to the activity. They can also be put in an active status by using the process button or by checking Auto Process. The system will automatically process any pending activity by date through cycle or batch processing.

Example:

The lightning bolt was selected for the CoverageCalculation activity. The activity processed and the values were displayed for the activity on the Activity Details screen. The activity is in active status because the lightning bolt was selected. The activity record has TypeCode 01 because it was manually entered. The StatusCode has changed from 02 (pending status) to 01(active status).

```
4 SELECT * FROM AsActivity where policyguid='CC53B2A5-960D-E4A8-CEB1-AE1BDDCB42E0'
```

1 record(s) [Fetch MetaData: 0/ms] [Fetch Data: 0/ms]

	ActivityGUID	TypeCode	StatusCode	Effectived
1	3CA7E66B-404C-56A0-1F49-F3616B239D94	01	01	5/1/2008 12:00

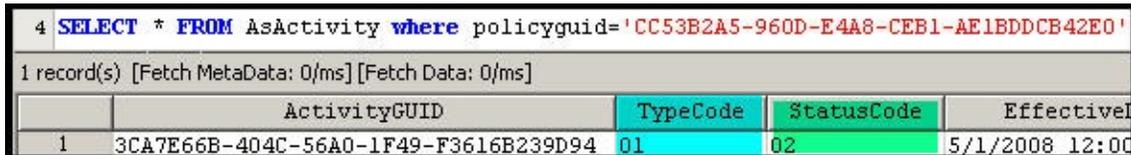
Activity in Database in Active Status

Pending Status

An activity is in pending status when information for an activity is entered but not processed. No calculations or math were run for this activity yet. An activity will be in a pending status because it was manually added or spawned but not processed. Verifying the TypeCode indicates how the activity was processed. Either way the activity record will remain in pending status until it is processed. Once processed, the StatusCode will be updated in the database and the status listed in OIPA will reflect the new status.

Example:

The CoverageCalculation activity was entered manually by selecting it from the activity drop-down box and then selecting the **OK** button. The activity displays on the Activity screen in a pending status. The activity record in AsActivity table stored the TypeCode 01 because it was manually entered and the StatusCode 02 because it is pending. It is important to note that if the activity was spawned it will have type code 01 as well.



The screenshot shows a SQL query result in a table. The query is: `4 SELECT * FROM AsActivity where policyguid='CC53B2A5-960D-E4A8-CEB1-AE1BDDCB42E0'`. The result shows 1 record(s) with the following columns: ActivityGUID, TypeCode, StatusCode, and EffectiveDate. The data row contains: 1, 3CA7E66B-404C-56A0-1F49-F3616B239D94, 01, 02, and 5/1/2008 12:00.

	ActivityGUID	TypeCode	StatusCode	EffectiveDate
1	3CA7E66B-404C-56A0-1F49-F3616B239D94	01	02	5/1/2008 12:00

Activity in Database in Pending Status

Shadowed Status

An activity is in shadowed status when it has been [recycled](#) or [deleted](#).

Recycle Example

Selecting the Recycle button associated with an activity creates two additional activities to be processed. The first is a reversal activity that has green reversal arrows for its Activity Detail icon. The second activity replaces the original activity. The reversal activity must be processed prior to the new activity that will replace the original.

Step 1: Select the recycle button for an activity.

Two pending activities are generated in OIPA for the reversal and new activity.

Activity	Activity Date	Status	Action
CoverageCalculation	01/29/2031	Pending	
CoverageCalculation	01/29/2031	Active	
CoverageCalculation	01/29/2031	Pending	

Two Pending Activities Generated For Recycle

Two additional records are written to the database to track the reversal and new activity.

```
4 SELECT * FROM AsActivity where policyguid='CC53B2A5-960D-E4A8-CEB1-AE1BDDCB42E0'
```

3:1 INS [5/1/08 12:43:35 PM EDT] Script executed - (0) Errors (Time: 110ms)

3 record(s) [Fetch Metadata: 0/ms] [Fetch Data: 0/ms]

	ActivityGUID	TypeCode	StatusCode	Effectiv
1	1921A837-9224-3C59-9026-03C033AA5D10	01	02	4/20/2008 12:
2	3CA7E66B-404C-56A0-1F49-F3616B239D94	01	01	5/1/2008 12:0
3	AC8C51AF-809D-ECC8-A093-F4FB70590ED8	02	02	5/1/2008 12:0

Reversal Records Written to Database

Step 2: Select the Process button for the reversal activity.

The status of the original activity is updated to shadow and the reversal activity status is updated to active.

Activity	Activity Date	Status	Action
CoverageCalculation	01/29/2031	Pending	
CoverageCalculation	01/29/2031	Shadow	
CoverageCalculation	01/29/2031	Active	

Active Reversal Activity

The status of the original activity's record changes from 01(active) to 12(shadowed). The reversal activity changes from 02(pending) to 01(active).

```
4 SELECT * FROM AsActivity where policyguid='CC53B2A5-960D-E4A8-CEB1-AE1BDDCB42E0'
```

	ActivityGUID	TypeCode	StatusCode	EffectiveDate
1	1921A837-9224-3C59-9026-03C033AA5D18	01	02	
2	AC8C51AF-809D-ECC8-A093-F4FB70590ED8	02	01	
3	3CA7E66B-404C-56A0-1F49-F3616B239D94	01	12	

Reversal Activity in the Database

Step 3: Select the pending activity that has the correct information. The activity's status changes from pending to active. The new activity's status changed from 02(pending) to 01(active).

```
4 SELECT * FROM AsActivity where policyguid='CC53B2A5-960D-E4A8-CEB1-AE1BDDCB42E0'
```

	ActivityGUID	TypeCode	StatusCode	EffectiveDate
1	1921A837-9224-3C59-9026-03C033AA5D18	01	01	4/20/2008 12:00:00
2	AC8C51AF-809D-ECC8-A093-F4FB70590ED8	02	01	5/1/2008 12:00:00
3	3CA7E66B-404C-56A0-1F49-F3616B239D94	01	12	5/1/2008 12:00:00

Explanation of Reversal Records in Database

Delete Example

Selecting the delete button associated with an activity generates a reversal activity. The reversal activity is in pending status and must be processed in order to delete the activity and place it in a shadowed status.

Step 1: Select the delete button associated with an activity. A reversal activity is generated with a pending status.

Activity	Activity Date	Status	Action
CoverageCalculation	01/29/2031	Active	
CoverageCalculation	01/29/2031	Pending	

Reversal Activity in Pending Status

There are now two records, one for the original activity and one for the reversal activity.

4 SELECT * FROM AsActivity where policyguid='B8015595-3B51-4BCC-C173-ADD71D00FD16'					
1:11	OVR	[5/1/08 5:08:54 PM EDT] Script executed - (0) Errors [Time: 109ms]			
2 record(s) [Fetch MetaData: 0/ms] [Fetch Data: 16/ms]					
	Ac	The original active activity	TypeCode	StatusCode	EffectiveD
1		54842A45-C54D-DA97-3B8B-9DAB1808959D	01	01	5/1/2008 12:00
2		The reversed activity in pending status	02	02	5/1/2008 12:00

Two Reversal Activities in Database

Step 2: Select the Process button associated with the reversal activity. The original activity is then updated to shadowed status and the reversal activity is updated to active status. The database record for the original activity is updated to 12(shadowed) and the reversal activity is updated to 01(active)status.



The delete display filter does not control the viewing of deleted activities. An activity that is deleted will have a shadowed status. Select the filter for shadowed activities to see those activities that were deleted. This filter displays the ability to delete activities via the trash can icon. The delete option should not be available in a production environment.

Queued Status

If an activity is in Queued status and the NUV Pending activity before it is deleted (shadowed) then the activity will still process with the current system date as its effective date. When ADVANCETOSYSTEMDATE is set to “Yes”, the activity cannot process with an effective date prior to the current system date.

If an activity in Queued status is deleted, it will behave as any pending activity and move to status “34” (Pending Shadow).

If an activity is deleted/inserted with a prior effective date, then the activity in Queued status is not incidentally deleted and remains in a Queued status until it is processed again.

When a Queued activity moves out of Queued status it will cause undo/redo on all activities after it with a newer effective date (considering Processing Order for same date activities).

You are here: Configuration > Transactions > [Activity Processing](#) > Activity Processing Dates

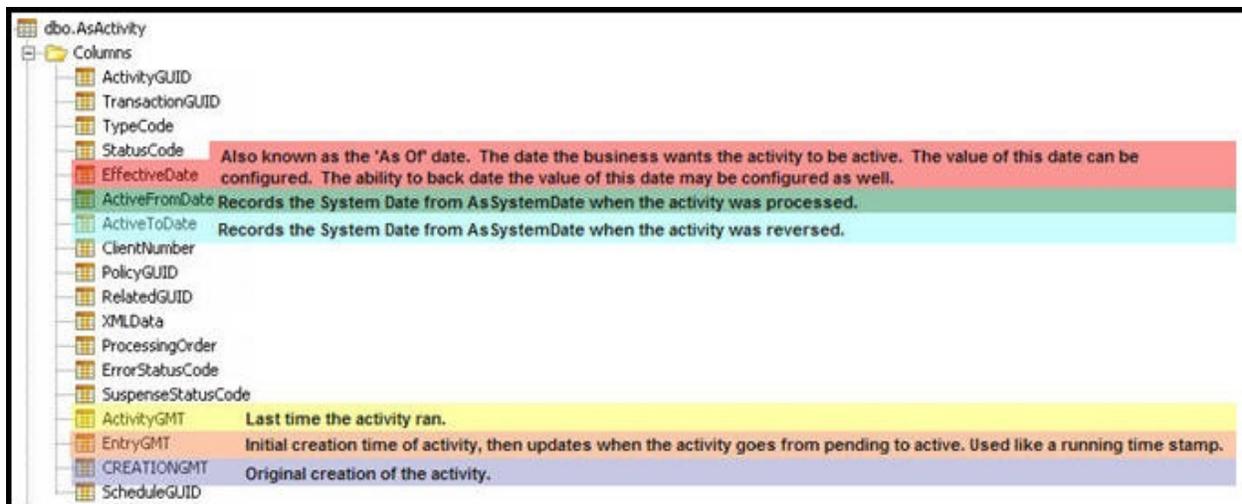
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Activity Processing Dates

The OIPA system controls activity processing through a timestamp in the table AsPolicy. This column always records time as Greenwich Mean Time (GMT).

When an activity is first processed the timestamp is compared against the current time. If the timestamp is less than the current time, then the timestamp will be updated and the activity will process. If the timestamp is greater than the current time, then the asp page will refresh and display a dialog box stating, "**Policy being modified by another user.**" Because the page is refreshed the current data will be displayed.

The look-up and update of the timestamp are performed within a transaction. This transaction will prevent another process from accessing the record until the transaction is complete. This prevents the same activity from being processed by two users.



Column Name	Description
ActivityGUID	
TransactionGUID	
TypeCode	
StatusCode	
EffectiveDate	Also known as the 'As Of' date. The date the business wants the activity to be active. The value of this date can be configured. The ability to back date the value of this date may be configured as well.
ActiveFromDate	Records the System Date from AsSystemDate when the activity was processed.
ActiveToDate	Records the System Date from AsSystemDate when the activity was reversed.
ClientNumber	
PolicyGUID	
RelatedGUID	
XMLData	
ProcessingOrder	
ErrorStatusCode	
SuspenseStatusCode	
ActivityGMT	Last time the activity ran.
EntryGMT	Initial creation time of activity, then updates when the activity goes from pending to active. Used like a running time stamp.
CREATIONGMT	Original creation of the activity.
ScheduleGUID	

Database Date and Time Tables

Activity Processed on 1/5/05					
Activity	TypeCode	StatusCode	EffectiveDate	ActiveFromDate	ActiveToDate
A1 Original activity	01 Natural	01 Active	1/5/05	1/5/05	NULL
Activity A1 Recycle on 1/6/05					
Activity	TypeCode	StatusCode	EffectiveDate	ActiveFromDate	ActiveToDate
A1 Original activity	01 Natural	01 Active	1/5/05	1/5/05	1/6/05
A2 Reversal activity	02 Reversal	01 Active	1/5/05	1/6/05	NULL
A3 New activity	04 Redo	01 Active	1/5/05	1/6/05	NULL

Activity Database Information

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Transactions > [Activity Processing](#) > Recycle Activities

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Recycle Activities

The reversal process in OIPA consists of an activity that is created to modify a processed activity. The reversal procedure allows changes to activities that could affect other processed activities. Using this functionality causes the system to respond by reversing all activities affected. The details will be maintained within the OIPA system.

- Identify the activity to reverse.
- Reverse the activity by selecting either the [delete](#) or [recycle](#) button.

Undo/Redo

Reversing any activity will cause most activities with a date LATER than the current activity to go through Undo/Redo. Undo and Redo are system generated activities that are created to correct any activities that were affected because they were dependent on a reversed activity's data.



Make sure to process the delete activity before the original activity is actually deleted.

Clear On Recycle

If an activity has the [field property](#) **Clear On Recycle** set to Yes, then when the activity is recycled the value captured by the field will be deleted and the field will contain no value.

Recycle and Reversal Exceptions

Some activities are configured to prevent a user from reversing or recycling it once it is processed. These activities are called **Nonreversible-Nonreversing** activities and they will not have a delete or recycle icon in the Action column on the Activity screen. They are also not affected by other activities that are reversed or recycled around them. A **Delete checkbox** is available on the Activity screen to reverse these activities in the special case that this action is needed. Only users with the [proper security privileges will have access to this checkbox](#). Clicking the Delete checkbox applies a trash can and recycle activity to the Activity column for the activity.

There is also another category of activities that have special reversal circumstances. **Reversible-Nonreversing** activities can be reversed, but they will not cause the activities around them to be reversed or recycled. These activities will have a trash can and recycle icon in the Action column on the Activity screen.

Deleting

Reversing an activity can be performed using two different methods depending on the desired outcome. If an activity needs to be reversed but not replaced with the same type of activity, then use the Delete icon. Deleting creates a new activity that is a reversal activity. This activity will change the status of the original activity to a **Shadow** [Shadow](#) status. All data that was updated by the original activity is updated with the original value.

1 2		Maximum Results 10			
	Activity	Activity Date	Status	Action	
	CoverageCalculation ← Original Activity	01/29/2031	Shadow		
	CoverageCalculation	01/29/2031	Active		

Deleted Activity in Shadow Status

Recycling

If the information stored in the activity needs to be updated, then use the Recycle icon. Recycling reverses a current activity and allows the original information to be updated. This eliminates the step of having to re-enter the same activity with the corrected information. Recycling works identically to deleting with the exception of scheduling a new activity to reprocess.

When clicked, the Recycle icon recycles an activity to change certain information that was originally entered. As a result, any activity that is spawned by this activity will be removed and regenerated upon reprocessing. The outcome of the recycle option is that a new and a reversal transaction are generated.

Activity	Activity Date	Status	Action
CoverageCalculation	01/29/2031	Pending	
CoverageCalculation	01/29/2031	Active	
CoverageCalculation	01/29/2031	Pending	

Three Activities Involved in Recycle Process

You are here: Configuration > Segments > Segments Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Segments Overview

Segments contain information about the base policy as well as additional riders or features of the policy. Each segment has its own business rule that is used to configure the segment information. These SegmentName business rules are named to reflect the segment they represent. The Base Coverage SegmentName business rule would be BaseCoverage. All of these rules can be found in the [Global Explorer](#) under the **Segments** node.

Segments are [created](#) from the Main Explorer from the **Segment** node inside a plan folder.

Segment Override Levels

Segments can be overridden at the following levels, listed from most to least specific:

- Plan
- Customer
- Child Product
- Product

Segment Screen Business Rule

There is also a SegmentScreen business rule, which controls the Segment screen in OIPA. The Segment screen can display multiple segments that the user can select from a drop-down box. When the user selects a specific segment, the Segment screen populates with the associated data. Segments are configured similar to business screen rules but allow for additional validations.

Segments and Calculate Rules

Segments can invoke [Calculate](#) business rules. In the Rules Palette each segment has its own calculate folder, which stores these business rules.



Use the Segment screen business rule to set-up the screen layout for any segments needed for the policy. Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Plan Rules | Segment Screen** .

You are here: Configuration > Segments > Create a New Segment

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create a New Segment

Steps to Create a New Segment

1. Open the **Main Explorer** tab.
2. Navigate to the **Company | Plan** folder where the segment will be created.
3. Right-click on the segment folder.
4. Select **New Segment**.
5. Enter the segment name and select **Next**.
6. Select the **TypeCode** for the segment and then select **Next**. (Please see [Codes Names](#) for further detail.)
7. Select **Finish**.

The segment will then appear in the segment folder in blue font. This indicates that it has been created, but has not yet been saved to the database. Check-in the segment in order for it to be saved.

Data Inheritance

Segments overridden at the Product, Child Product and Plan levels are able to utilize [data inheritance](#). Configuration inherited from a higher override level will display in grey, indicating that it is unable to be edited. However, the configuration may still be able to be altered by overriding that piece of configuration entirely.

The following pieces of segment configuration are able to utilize data inheritance:

- Fields
- Events
- ScreenMath
- Actions
- Roles
- RoleViews

General Pane

View the segment properties, such as version number and TYPE, from the General pane.

Fields Pane

This pane is used to create fields that will display on the Segment's screen. The functionality is exactly the same as the Fields pane described in the rules section of this guide. Please see the [Fields](#) pane for more information.

Events Pane

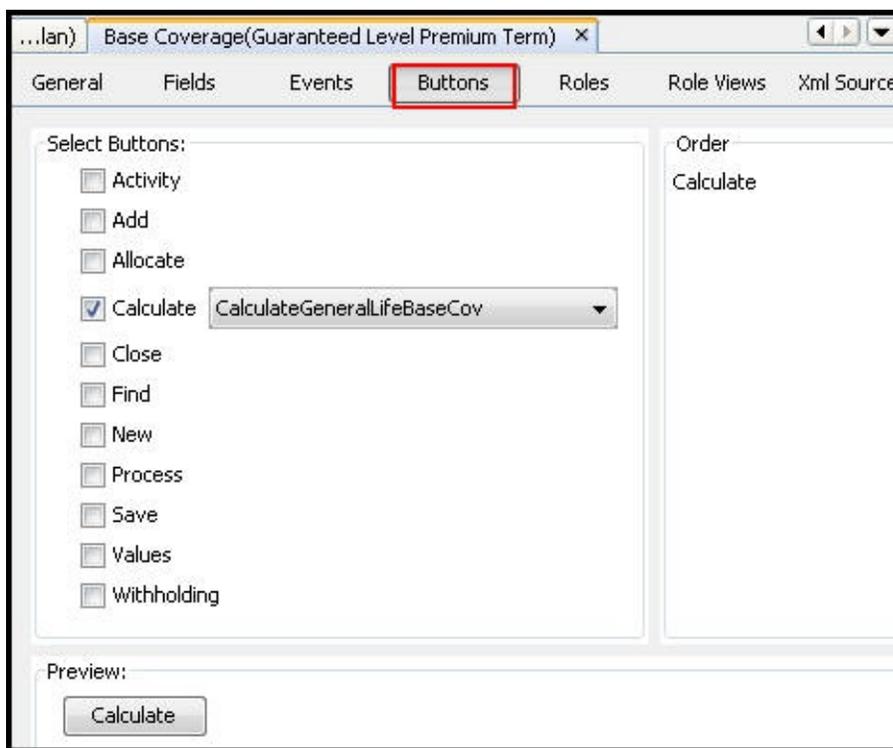
This pane is used to create validations that will display on the Segment's screen. The functionality is exactly the same as described in the configuration section. Please see the [Events](#) Pane for more information.

Buttons Pane

The Buttons pane allows for the selection of the type of button to be displayed on the Segment screen. Checking the box next to the button adds the button name to the right in the Order column. The Up and Down buttons can be used to change the order of the buttons on the screen. There is a preview of the buttons located in the Preview box at the bottom of the [Configuration Area](#).

Calculate Button has a drop-down box where a Calculate business rule can be selected for the segment. When a segment is calculated, the math configured in a Calculate business rule takes into account various aspects of the policy and arrives at the values, which are critical to the policy. [Calculate business rules](#) will all start with CalculateGeneral as a standard.

The Calculate rule selected is then stored in the Segments Calculate Rule folder in the Main Explorer tab for that plan. The Calculate rule can be edited from this folder by double-clicking on its XML file.



Buttons Pane for a Segment

Button Security

Security is applied to the buttons on the Segment screen in the Admin Explorer. Open the Security folder to reveal the options for applying security.

Steps to Add Security to Buttons

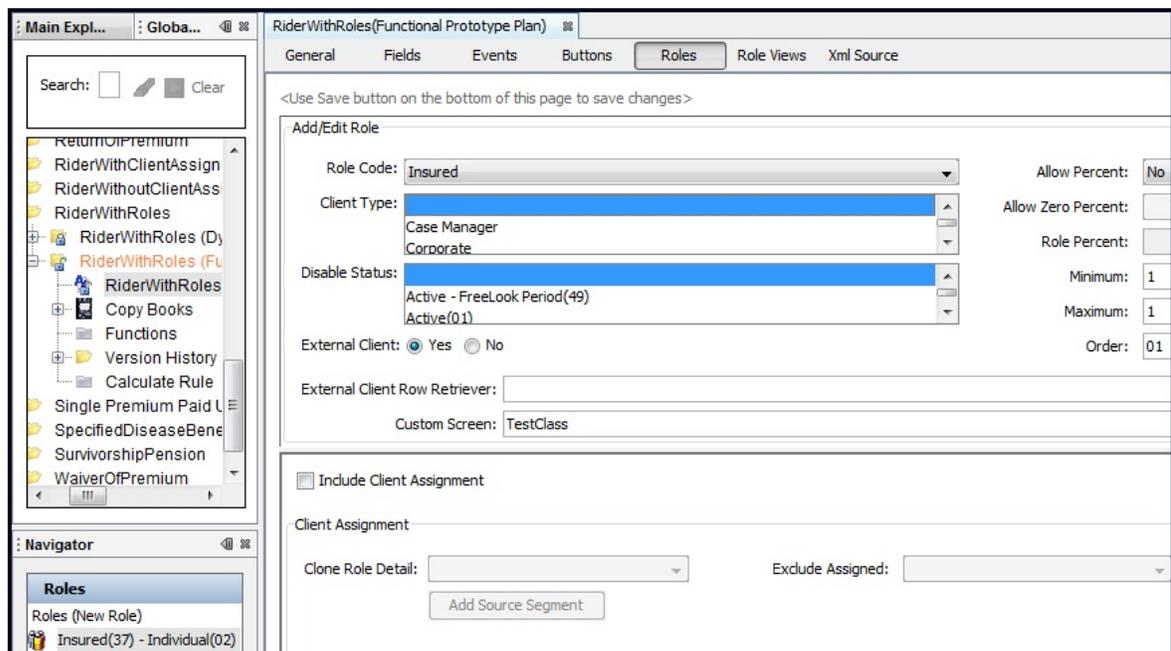
1. Open **Admin Explorer | Security | Application Security | Security Groups**.
2. Select the Security Group that has access to the Segment screen.
3. Open **Plan Security | *Name of the Company* | Plan Pages**.
4. Right-click the PolicySegment node and select **Check-out** to open it in the Configuration Area.
5. Check the box next to each button the users in the security group will have access to in OIPA.
6. Check-in the file to save the changes to the database.

Roles Pane

The Roles pane is used to add, edit, or remove roles that are associated with a segment.

Steps to Add a New Role

1. Select the **Roles** pane.
2. Right-click on the Roles node in the Navigator menu and select **New Role**.



Roles Node in Navigator Window

3. Select the desired role name from the available list in the **Role Code** drop-down box.



Role options are populated by AsCodeRole, which can be updated through the [Codes Names](#) folder in Admin Explorer.

4. Select **Yes** or **No** in the Allow Percent drop down box. If Yes is selected, then the AllowZeroPercent drop down box will be enabled.
5. Select **Yes** or **No** in the AllowZeroPercent drop down box.

6. If applicable, select a **Client Type**. This selection defines the client types that can be assigned to the role. Only these client types will be displayed in the client type drop-down box on the Client screen for the specific role. For example, an insured may need to be an Individual client type for a particular plan.
7. Enter a role percent in the **Role Percent** field. Percentages can range from 0% to 100%, or an asterisk ("*") can be used to indicate that the total percent for the role can exceed 100%. Each individual record will still have a maximum of 100%, but the total for multiple records on the same Policy Role will be able to exceed 100%.

Using an asterisk in the Role Percent field will generate a warning message at the bottom of the Roles pane.

4. Select a status from the **Disable Status** list. This tells OIPA to disable role fields based on policy status. The list is populated by AsCodeStatus.
5. Specify whether the role is an external client by selecting the **Yes** or **No** radio button in the External Client field. An external client is a client whose data is stored in an external database that cannot be modified within OIPA.
 - **Yes:** Indicates that client data is stored in an external database. If this option is selected, the EXTERNAL attribute with a value of "Yes" is added to the <Role> element in the XML source. Additionally, if the CustomScreen field is populated, then the Client Type field can be left blank. If the CustomScreen field is not populated and no Client Type is chosen, then the system will generate an error message upon saving the role. These conditions apply to both visual and XML source configuration.
 - **No:** The default selection—indicates that client data is not stored in an external database, and can be modified within OIPA. The Custom Screen and External Client Row Retriever fields will be disabled, and a client type must be chosen, or the system will generate an error message upon saving the role. This condition applies to both visual and XML source configuration.

6. If the External Client radio button is set to Yes, type a class name in the External Client Row Retriever field. This should be the name of a class that has been created to implement the external client search.
7. If the External Client radio button is set to Yes, and if applicable, type a class in the Custom Screen field. This should be the name of a class that has been created to implement a customized display of external client search results. Note: Refer to the Extensibility document on OTN in the library for this release to access additional information on extending OIPA to include external clients.
8. Type a minimum in the Minimum box. This is the minimum number of this type of role that can exist on the segment.
9. Type a maximum in the Maximum box. This is the maximum number of this type of role that can exist on the segment.
10. Type an order in the Order box. This is the order in which the role will appear on the screen.
11. Select **Yes** or **No** from the **Final** drop-down box. Selecting **Yes** will prevent a lower level override of the segment from altering the role configuration.
12. Select **Yes** or **No** from the **Hidden** drop-down box. Selecting Yes will hide the role on the Segment Role screen.
13. Check or uncheck the **Include Client Assignment** checkbox.
 - **Checked:** The Include Client Assignment check box will enable the following source segment fields: Add Source Segment button, Clone Role Detail combo box, Exclude Assigned combo box. The selection of the Include Client Assignment check box will not place any tags into the XML Source until the Save button on the bottom of the editor is clicked.
 - **Unchecked:** If the Include Client Assignment check box is unselected then all of these actions will be done: all Source Segments defined for the role will be removed. The ClientAssignment element will be removed from the XML Source after the Save button is clicked.
14. Select **Yes** or **No** to Clone Role Detail.
15. Select **Yes** or **No** to Excluded Assigned.

16. Click the **Add Source Segment** button to add a Source Segment and Segment Role Code fields. The maximum number of additions will be equal to the number of segments for the plan minus one.
17. Select the **Save** button at the bottom of the Roles pane when finished.

Note: To add multiple new roles, you need to select **Add Role** from the Navigator window before each new role is entered. This clears the fields. CTRL + click deselects the options if they are already selected in a drop down list.

Steps to Remove a Role

1. Click the **Roles** pane.
2. Click the role to be removed from the Roles section of the Navigator window. The role will appear highlighted in blue.
3. Right-click the role and select **Delete**.

Steps to Edit a Role

1. Select the **Roles** pane.
2. Select a role from the Navigator window. The role will appear in the Configuration Area.
3. Make the necessary changes.
4. Click **Save** at the bottom of the Roles pane.

Role Views

The Role Views pane controls two aspects of roles: Role Views and Display Status.

- **Display Status:** determines the filter status for the role views. Multiple statuses can be selected.
- **Role Views:** controls the roles that are visible to the user. Multiple roles can be selected.

Steps to Set Display Status

1. Check-out a segment.
2. Click the **Role Views** pane. This will open the Navigator window in the bottom right corner of the screen.
3. Click **Display Status** in the Navigator window. This will open the Display status options in the Configuration Area.
4. Select the status when the Role Views may be used. Hold the CTRL key down and click to select multiple statuses.
5. Click **Save** when finished.

Steps to Create Role Views

1. Check-out a segment.
2. Click the **Role Views** pane. This will open the Navigator window in the bottom right corner of the screen.
3. Right-click the Role Views node in the Navigator window and select **New Role View**. The Role View information will open in the Configuration Area.
4. Enter a name for the view in the **View Name** box.
5. Select the roles that will be available to display for that view. Hold the **CTRL** key down and click to select multiple roles or use the **All Roles** checkbox.
6. Click **Save** when finished.

Steps to Remove a Role View

1. Right-click on the Role View in the Navigator window and select **Remove Role View**.
2. Click **OK** when the Confirmation Window appears.

XML Source Pane

This pane allows individuals who prefer to configure in XML to do so. Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Plan Rules | Segment Screen**.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Programs > Programs Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Programs Overview

A program consists of multiple activities associated together to complete a recurring business process. A program can be set up to run on a policy or specific segment for a predetermined period of time. It includes attributes that define the program instance and as a result, activities can be run over time according to a specific schedule. Examples of such programs include asset rebalancing, automatic investment plans, systematic withdrawals and cost of living adjustments.

Users can inquire and change program data and scheduling information through the use of a unified programs view, instead of trying to piece together these facts by looking at activities and activity history.

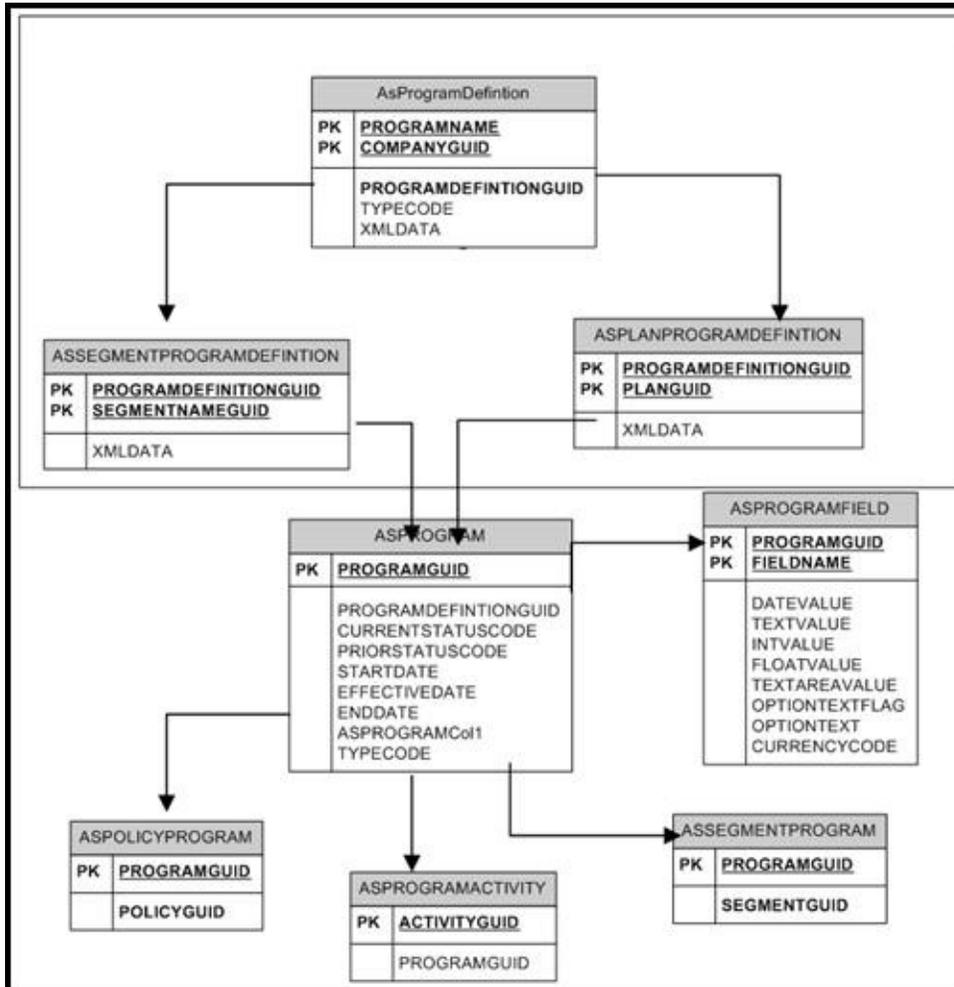
Several [prototype programs](#) have been configured to demonstrate the configuration requirements for this functionality.

Business Rules

There are several business rules that drive program functionality. Refer to the [Program Business Rule](#) page for a complete explanation of these rules.

Database Tables

The following database diagram shows the relationship between the tables that store program data.



Program Database Tables

System Codes

There is one system code and one user defined code that store program information: AsCodeProgramStatus and AsCodeProgramType. Each of these can be edited using the [Code Names](#) editor in the Admin Explorer.

- **AsCodeProgramStatus:** this system code identifies program statuses. Program statuses are referenced in the <ProgramPriorStatusCode> element in the [ProgramDefinition](#) rule, when identifying a program's status prior to suspension.
- **AsCodeProgramType:** this user defined code identifies the various types of programs. It is referenced in the <ProgramType> element in the [ProgramScreen](#) rule.

Security

Security is applied to the Program screen in OIPA using the Plan Security node for [Program](#). If the main **ProgramScreen** checkbox is not checked, then the Program node will not be visible in OIPA.

There are also two business rules that control the display of the Add and Save buttons on the Program screen in OIPA. The PolicyProgram and SegmentProgram rules have an <Eligibility> section that identifies the policy statuses that allow programs to be added and updated. If security is applied, but a policy is in a status not defined in this section of the rule, then the Add and Save buttons will not be available. Refer to the [PolicyProgram](#) and [SegmentProgram](#) rules for additional information.

You are here: Configuration > Programs > Programs Business Rules

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Programs Rules

There are several rules that drive program functionality. Each rule is explained below and an image of the corresponding section of OIPA controlled by the rule is provided.



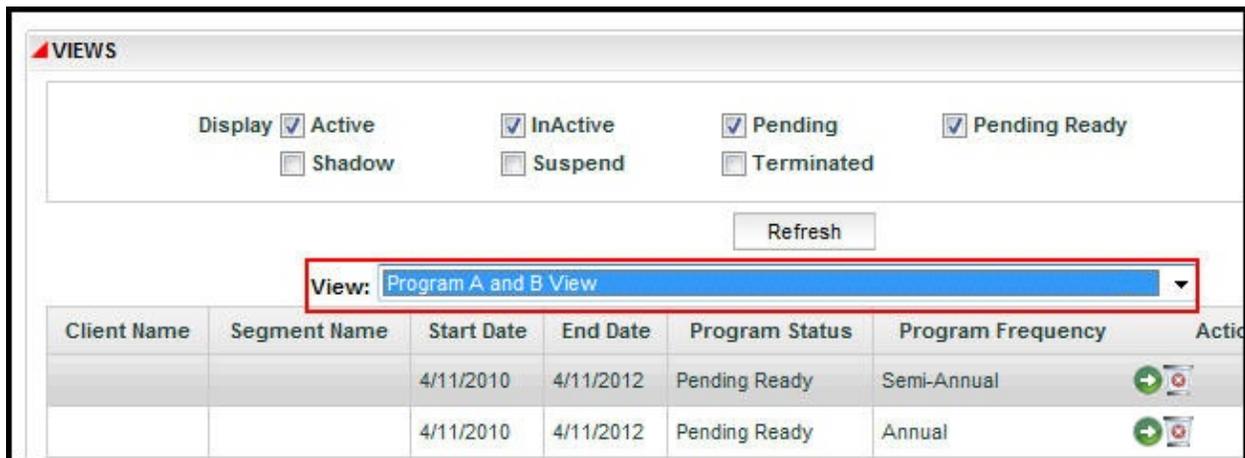
A complete explanation of all elements and attributes available in these rules can be found in the XML Configuration Guide. This guide is available from the **Help** option in the Rules Palette main menu.

Programs Rules and the OIPA Application

OIPA supports many different types of rules. In the case of programs, there are rules that drive screen configuration and rules that drive activity processing. The rules are divided into categories below according to the main purpose of the rule.

Screen Configuration Rules

ProgramScreen: This rule is used to define the table views that are available to a user from the **Views** section of the Program screen in OIPA. A view can be created and associated with multiple program types. When a particular view is selected, the table headings change to reflect the view requirements defined in this rule and the information is displayed for the program types associated with the view.



Program View Drop Down in OIPA

ProgramDefinition: This rule defines the fields and validations that occur when a new program is created. The section of the Program screen in OIPA controlled by this rule is the **Program Detail** section. This rule also defines the actions that can be performed on the program. Available actions are Start, Run, Terminate and Reinstatement. Only actions identified in the <ProgramAction> element will be supported on a program.

PROGRAM

Program Detail | Audit History | Activity Summary

PROGRAM DETAIL

Program Status: Pending Ready

Effective Date: 4/11/2010

ClientName:

Program Amount: 800.00 USD

Start Date: 4/11/2010

End Date: 4/11/2012

Program Frequency: Annual

Next Scheduled Date:

Save

Program Details on Program Screen in OIPA

PolicyPrograms: This rule controls the availability of the **Add** and **Save** buttons on the Program screen for policy programs. The policy statuses defined in the rule determine when the buttons are visible to the user. When a policy is in one of the statuses identified in this rule, the buttons will be available.

SegmentPrograms: This rule controls the availability of the **Add** and **Save** buttons on the Program screen for segment programs. The policy statuses defined in the rule determine when the buttons are visible to the user. When a policy is in one of the statuses identified in this rule, the buttons will be available.

ADD PROGRAM

Policy Segment

Programs: Select Program

Add

PROGRAMA

Program Detail : Audit History : Activity Summary

PROGRAM DETAIL

Program Status: Pending Ready

Effective Date: 4/11/2010

ClientName:

Program Amount: 800.00 USD

Start Date: 4/11/2010

End Date: 4/11/2012

Program Frequency: Annual

Next Scheduled Date:

Save

Program Add and Save Buttons in OIPA

Activity Processing Rules

CopyToProgramFields: This rule allows one or more math variables to be copied from an activity to one or more program fields, upon processing the activity with this rule attached. As a best practice, the ProgramGUID should be referenced in the configuration so that the GUID can be used as an identifier. An example of this is shown below:

```
<MathVariable VARIABLENAME="ProgramGUID" TYPE="EXPRESSION" D
```

ReinstateProgram: This attached rule creates the appropriate reinstate activity as defined in ProgramDefinition. Programs can be reinstated from a status of Suspend.

SuspendProgram: This is an attached rule that will lock down the program and prevent any updates to the program information.

TerminateProgram: This is an attached rule that will end the program.

ResetProgram: This rule, attached to the SetUpProgram activity, sets the program status to PendingReady when the SetUpProgram activity is reversed.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Programs > Create Programs

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create Programs

Programs are created from the Global Explorer **Programs** node. The Global Explorer is the only tab where programs can be added or edited. Once a program is created, it can be linked to a segment or plan. A program can only be linked to a segment or plan once.

Add a New Program

Programs are added from the Global Explorer tab. A ProgramDefinition.xml file is created for each program to hold all the program characteristics. There are five panes available in the Configuration Area for programs.

- **General pane:** this pane provides additional general information for the program, such as the version number.
- **Fields:** this pane is used to configure the fields that are available to a user in OIPA for capturing program details.
- **Program Actions:** this pane is used to provide a simple method for configuring program actions. A list of available actions is available to the left of the Configuration Area. Clicking on an action will add the opening tag to the Configuration Area. The additional information can be added directly under the tag. When an action has saved configuration, clicking on the action will display that action's XML in the Configuration Area. This gives a close up view of the specific action configuration. If the complete configuration needs to be viewed, click the XML Source tab.
- **Events:** this pane is used to provide a simple method for configuring events and actions. A list of available events and actions is available to the left of the Configuration Area. Clicking on a node will add the opening tag to the Configuration Area. The additional information can be added directly under the tag. When an event or action has saved configuration, clicking on the action will display that action's XML in the Configuration Area. This gives a close up view of the specific even or action configuration. If the complete configuration needs to be viewed, then click the XML Source tab.
- **XML Source:** this pane provides a way to configure directly in the XML without using the visual configuration tools.

Steps to Add a New Program

1. Navigate to the **Global Explorer** tab.
2. Open the environment folder and right-click the Programs node. Select **New Program**. The New Program wizard will display.
3. Type the name of the new program in the New Program Name field.

4. Select a program type from the Program Type drop down box. Types are defined in AsCodeProgramType.
5. Select a primary company to associate to the program.
6. Click **Finish**. The new program will be listed under the Program node.
7. Double-click the ProgramDefinition.xml file for the new program to open it in the Configuration Area.
8. Configure the [ProgramDefinition](#). Refer to the **XML Configuration Guide** in the Help menu for a complete explanation of elements and attributes.
 - configure the [Fields](#) that will display when the program is added to the Program screen in OIPA.
 - configure the [Events and Actions](#) needed for the program.
 - configure the program actions. The action can be visually configured in the Program Actions tab or straight in the XML using the <ProgramAction> element. The four options are Start, Run, Reinstate and Terminate. From the Program Action tab, click the action in the Program Actions Navigator pane that needs to be added. The opening tag will appear in the Configuration Area. Identify the transaction that kicks off the action in the <Transaction> element. Fields are defined if needed. A special <ReinstateTransaction> element is needed to further define the reinstate process. The [program prototype](#) example demonstrates all the configuration options for program actions.
9. Check-in the ProgramDefintion.xml file to save the configuration changes.

Link a Program to a Plan or Segment

After a program is created and the ProgramDefinition is configured, it can be [linked to plans](#) or [linked to segments](#).

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Funds > Index Funds > Index Fund Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Index Fund Overview

OIPA contains the architecture required to administer Equity Indexed Life (IUL) and Annuity products (EIA). These types of policies give companies the option of allowing policyholders to use a Stock Market Index rather than individual funds as a basis for account value growth. Many customers prefer this strategy as it reduces exposure to market risk while at the same time provides limited stock market participation. The performance is based on an underlying index such as the S&P 500 or Russell 2000. Other factors that impact the calculation of the credit to the policy, called the Index Credit, are variables such as the Cap, Floor and Participation rate.

The Index Prototype Company in the Prototype Company of the Rules Palette demonstrates some common design features that are characteristic of index funds.

Common Index Fund Characteristics

Sweep Date

Many index funds use a **Sweep Date**. Sweep dates, in this context, restrict the addition of money into the Indexed tracking funds to certain times, such as a particular day of the month. In order to accommodate that restriction the concept of an Interim or Holding Fund is introduced. Money received prior to a given date will be held in the interim fund and credited with a fixed rate of interest. When the Sweep date arrives that deposit plus the interest earned is added to the Indexed Fund. In OIPA, a combination of plan data can be configured to set a standard Sweep Date and then math configuration for calculating any needed logic to figure out the actual business day the sweep date can occur is reused to support this portion of Index fund design.

Index Segment Period

Indexed funds also use an Index Segment Period, which is the length of time money allocated to an indexed fund must remain in the fund in order to receive the full value of the Index Credit. This is also known as the Index Period. Through transaction configuration, premiums are swept into indexed funds according to allocations. A configurable feature called **buckets** provide a means of tracking multiple premiums to ensure index segment period requirements are met.

There are several common methods that can be used to credit the policy account. The three common types are listed and summarized below:

- **Annual Point to Point:** this is where the beginning equity index value is recorded and compared to the ending equity index value at the end of the index period. If the ending index value is higher, then interest is credited annually subject to the participation rate and growth cap or spread. If the ending index value is lower, then no interest is credited.
- **Averaging Method:** this takes the average daily indexed value over the entire index period and compares this average with the beginning index

value at the first day of the index segment. If the average indexed value over the entire index period is greater than the beginning index value, then interest is credited. (While stated as daily, can also be monthly.)

- **High Water Mark:** this uses the difference between the starting value of the index and the highest value achieved by the index during the crediting period.

For each of these methods, OIPA rate tables are used to store index daily rate values, which can be easily called from configuration. Configuration logic can figure out if an ending index value is higher, compare averages or figure out a difference, which is all supported by the OIPA math syntax.

Valuation for index funds is not done in unit values, but rather an Index or Index Value is stored in AsRate and updated. A range of these values is then used to determine the percentage that will be used as the main component for the Index Credit or credit.

Some other product design items use the Rate table for information that isn't technically a rate, but because it is easy to call the Rate table from configuration, it is stored there. These values are applied to the Index Credit to arrive at the final Index Credit that is used to update the policy's value. The product's specification will determine which of these should be used.

- **Participation Rate:** this determines how much of the index's gain the policy will receive.
- **Growth Cap:** this limits the interest rate to a certain percentage.
- **Growth Floor:** this sets the minimum interest rate that could be credited (usually 0%).
- **Spread** or (sometimes) **Asset Fee:** this is a fee.
- **Minimum Guarantee:** this sets a minimum return value for the policy, most of the time a percentage of the total premium plus an interest credit.

Buckets

Buckets are used to uniquely identify, value and track one or more deposits into an indexed fund that must be valued separately from the fund as a whole. In OIPA, a bucket is represented by an integer value in the AsBucket database table. The bucket links a bucket and its Effective Date with a policy and the actual values of premium and the associated bucket in the AsValuation table, using the BUCKET column.

The maximum number of buckets for an indexed fund is dependent on how often a new Index Segment can begin and the duration of the Index Segment, which can then be configured in the Plan Data page of the Rules Palette. For example, if a new Index Segment can begin once a month and the duration is one year, there could be a maximum of 12 buckets. If the duration is 5 years, then there could be up to 60 buckets.

The bucket value is updated when deposits are transferred (“swept”) into an indexed fund. It is common to use the calendar month as the integer value for twelve month tracking periods. For example, a deposit may be received on February 15. If the Sweep day is the 28th of each month, then on February 28 the net deposit is transferred to the user allocation. The bucket is updated to **2** if any money is allocated to an indexed fund in that transfer.

In **assignment processing**, there are two attributes in the Assignment tag that relate to buckets.

- **BUCKET**: identifies the bucket number that money will be assigned to or removed from when money movement occurs.
- **BUCKETEFFECTIVEDATE**: Specifies the effective date when money can be assigned to a bucket for the following assignment types: Apply, ApplyByFund, Transfer (To Side), Switch (To Side).

When working with **arrays**, the following math operations relate to buckets.

- There is a math operation for TYPE="INTEGERARRAY" called OPERATION="FILLBY-BUCKETLIST. This accepts a FundGUID as the MathVariable element’s text and assigns to the MathVariable an array of bucket numbers for the given fund.

When pulling values into buckets, keep the following information in mind.

- There is a math variable TYPE="FIELD" with the following syntax
Valuation:Fund:MyFundGUID:Bucket:MyBucketNumber:CashValue

High Level Steps to Configure Indexed Funds

1. Create a plan.
2. Create and configure a FundScreen override.
3. Create interim funds to hold premiums until the Sweep date.
4. Create index funds.
5. Configure allocations to allow policy owners the option to select index funds via the OIPA interface.
6. Configure the InterestRateCalculation business rule to calculate the interest that should be credited to the premium in interim funds.
7. Create Rate Groups and enter initial rates for the index funds.
8. Create Rate Groups and load initial rates for the Growth Cap, Growth Floor, Participation Rate and Spread.
9. Configure static values.
10. Configure reusable configuration.
11. Configure transactions to take in money, move money into the interim account, move money from interim account into index funds, apply index credits and other design features according to the product design.
12. Configure the ValueScreen business rule.

For detailed configuration steps involved in setting up index funds, refer to the [Configure Index Fund](#) section.

Related Topics

[Configure Funds](#), [Fund Database Tables](#), [Child Funds](#), [Currency Codes for Funds](#), [Fund Screens in OIPA](#), [Allocation Models](#)

You are here: Configuration > Funds > Index Funds > Configure Index Funds

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Configure Index Funds

Configuring index funds is a multi-step process. A complete explanation of the steps involved in the process are provided below in the order in which they should be performed.

Step 1: Create the Plan

The first step is to create the plan that will use index funds.

1. In the Main Explorer of the Rules Palette, navigate to **Companies | Name of Company | Subsidiary Companies | Name of Subsidiary Company | Products | Name of Product | Plans**.
2. Right-click on the **Plans** folder and select **New Plan**. Skip this step if the plan already exists.



Plan Menu in Main Explorer

3. In the New Plan wizard enter the following:
 - New Plan Name: The name of the product being configured.
 - Currency Code: Select the default currency for the plan.
 - Market Maker: If converting currencies for funds, select the market maker.
 - Plan Effective Date: Enter the date when the plan will be available for enrollment.
 - Plan Expiration: Enter for plans that are only available until a specific date.
 - Point In Time Valuation: Select **No**. This is not available for index funds.
 - Plan Allocation Method: Use Default or Plan Allocation Method. Both methods use parent level funds.

Step 2: Configure the Fund Screen

Configure fund fields that will allow funds to be properly identified. For example, this is where fields are created for the Fund Index, such as S&P 500 or NASDAQ. Other fields could be the Index Calculation Method to use such as Cap, Spread, Floor and Participation.

Refer to the [Create Parent Level Funds](#) section for the necessary steps to configure the FundScreen rule. Only configure the <ParentField> element when creating fund fields. Index funds must use parent funds.

Step 3: Create Interim Fund

Because of the design of Index products, funds must be created to hold premium until the Sweep date is reached. These interim funds are fixed assets and use the InterestRateCalculation business rule to value the interest rate that will be credited.

Refer to the [Create Parent Level Funds](#) section for the necessary steps to configure funds.

Step 4: Create Index Funds

Create the index funds and select the fund fields configured in Step 2 from the section above. These funds are variable based and use daily values of the index for calculating the Index Credit. When selecting the index design characteristics, such as fund index, index calculation method, etc., the actual logic is not yet added. This just identifies that the elements are used. Configuration occurs later in the transactions, copybooks and other rules. Index funds will eventually be mapped to the plan.

Refer to the [Create Parent Level Funds](#) section for the necessary steps to configure funds.

Step 5: Configure Allocations

After the funds are set up, the allocation structure that allows OIPA end users to select the actual funds they want to allocate their premiums to must be configured. Use either the Default method or the Allocation Model method.

Refer to the [Default method](#) or [Allocation Model](#) section for additional information.

Step 6: Configure the InterestRateCalculation Rule

The InterestRateCalculation business rule is the rule that should be configured for fixed funds. Configure static interest rates that are commonly used for interim funds. There are various interest rate calculation elements that can be configured to customize how the interest is calculated. This rule is overridden and attached at the fund level.

Refer to the [InterestRateCalculation](#) section for additional information.

Step 7: Create Rate Groups and Enter Initial Rates For Indexed Funds

Index funds need separate rate tables to store index values. After the tables are created, RATE and RATEARRAY math variables can be used to pull data into configuration logic for index calculation methods.

Refer to the [Rate Overview](#) section for additional information on creating rate groups and entering rates.

Step 8: Create Rate Groups and Enter Initial Rates For Growth Cap, Growth Floor, Participation Rate and Spread

Index fund product design uses various percentages that can change. Rate groups can be created to support this. These rate groups can hold values for design elements such as Growth Cap, Growth Floor, Participation Rate and Spread. After the initial rates are loaded into the rate structure, you may decide how additional values are loaded in the future, whether it be manually through the Rules Palette or through AsFile.

Refer to the [Rate Overview](#) section for additional information on creating rate groups and entering rates.

Step 9: Set Up Static Values

A typical product design will require static values to be set that are called by configuration. The Product Data node can be configured for values such as Maximum buckets and Sweep Dates. These fields should be configured in the PlanScreen rule. The values can be entered in the Plan Data node.

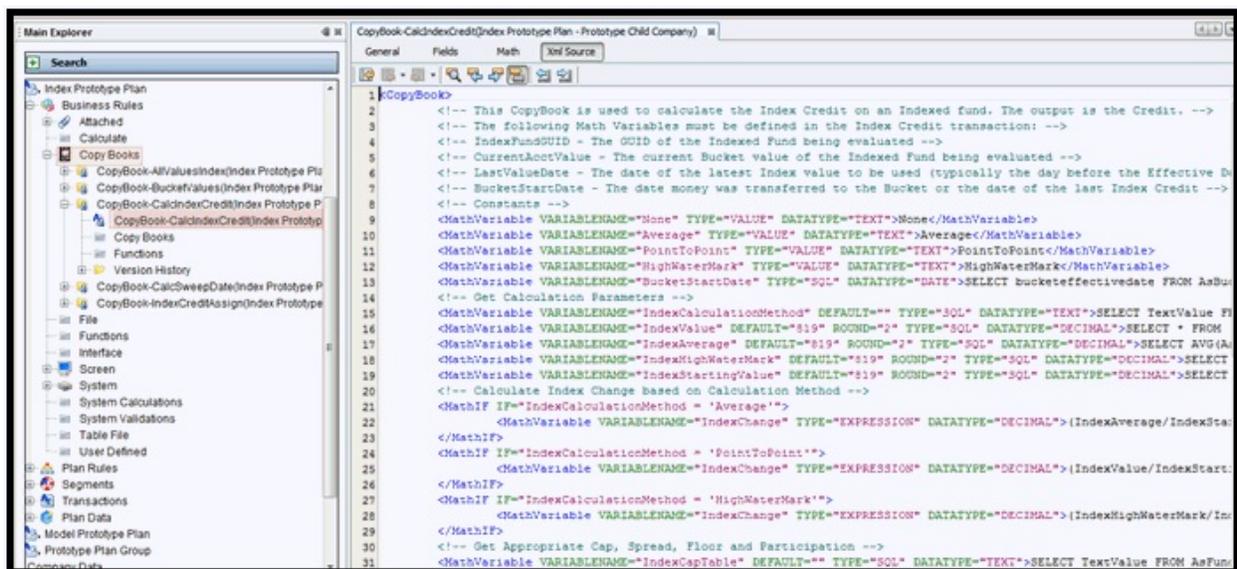
1. Navigate to the [PlanScreen](#) rule in the appropriate plan and check out the rule.
2. Configure the fields that will store the static values required then check in the rule.
3. Navigate to the **Plan** node and check out **Plan Data**.
4. Enter the values in the fields that were configured in the PlanScreen rule.
5. Check-in the Plan Data.

Step 10: Configure Reusable Configuration

Index funds products will reuse logic to calculate items such as the index credit or the actual sweep date. CopyBooks or Function rules can be used to configure this reusable logic. Transactions and other rules will call CopyBooks and Functions for these commonly used configurations.

The image below is taken from the Index Prototype Plan that is provided with the release. The prototype uses CopyBooks to calculate the credit for the index. Math variables are set up for the methods of crediting. Then a SQL is used to retrieve the actual index method set for the fund, which was set when the fund was created (from step 4). Other values are then queried to retrieve values from the rate tables that were created. The IF statements are used to figure out the index method and then run custom calculations.

Refer to the CopyBook prototype examples for specific CopyBook configuration. Navigate to **Main Explorer | Companies | Prototype | Subsidiary Companies | Prototype Individual Child Company | Products | Individual Prototype Product | Plans | Index Prototype Plan | Business Rules | CopyBooks.**



```
1 <CopyBook>
2 <!-- This CopyBook is used to calculate the Index Credit on an Indexed fund. The output is the Credit. -->
3 <!-- The following Math Variables must be defined in the Index Credit transaction: -->
4 <!-- IndexFundGUID - The GUID of the Indexed Fund being evaluated -->
5 <!-- CurrentAcctValue - The current Bucket value of the Indexed Fund being evaluated -->
6 <!-- LastValueDate - The date of the latest Index value to be used (typically the day before the Effective D
7 <!-- BucketStartDate - The date money was transferred to the Bucket or the date of the last Index Credit -->
8 <!-- Constants -->
9 <MathVariable VARIABLENAME="None" TYPE="VALUE" DATATYPE="TEXT">None</MathVariable>
10 <MathVariable VARIABLENAME="Average" TYPE="VALUE" DATATYPE="TEXT">Average</MathVariable>
11 <MathVariable VARIABLENAME="PointToPoint" TYPE="VALUE" DATATYPE="TEXT">PointToPoint</MathVariable>
12 <MathVariable VARIABLENAME="HighWaterMark" TYPE="VALUE" DATATYPE="TEXT">HighWaterMark</MathVariable>
13 <MathVariable VARIABLENAME="BucketStartDate" TYPE="SQL" DATATYPE="DATE">SELECT bucketeffectivedate FROM AsBu
14 <!-- Get Calculation Parameters -->
15 <MathVariable VARIABLENAME="IndexCalculationMethod" DEFAULT="" TYPE="SQL" DATATYPE="TEXT">SELECT TextValue F
16 <MathVariable VARIABLENAME="IndexValue" DEFAULT="0" ROUND="2" TYPE="SQL" DATATYPE="DECIMAL">SELECT * FROM
17 <MathVariable VARIABLENAME="IndexAverage" DEFAULT="0" ROUND="2" TYPE="SQL" DATATYPE="DECIMAL">SELECT AVG(A
18 <MathVariable VARIABLENAME="IndexHighWaterMark" DEFAULT="0" ROUND="2" TYPE="SQL" DATATYPE="DECIMAL">SELECT
19 <MathVariable VARIABLENAME="IndexStartingValue" DEFAULT="0" ROUND="2" TYPE="SQL" DATATYPE="DECIMAL">SELECT
20 <!-- Calculate Index Change Based on Calculation Method -->
21 <MathIF IF="IndexCalculationMethod = 'Average'">
22 <MathVariable VARIABLENAME="IndexChange" TYPE="EXPRESSION" DATATYPE="DECIMAL">(IndexAverage/IndexSta
23 </MathIF>
24 <MathIF IF="IndexCalculationMethod = 'PointToPoint'">
25 <MathVariable VARIABLENAME="IndexChange" TYPE="EXPRESSION" DATATYPE="DECIMAL">(IndexValue/IndexStart
26 </MathIF>
27 <MathIF IF="IndexCalculationMethod = 'HighWaterMark'">
28 <MathVariable VARIABLENAME="IndexChange" TYPE="EXPRESSION" DATATYPE="DECIMAL">(IndexHighWaterMark/In
29 </MathIF>
30 <!-- Get Appropriate Cap, Spread, Floor and Participation -->
31 <MathVariable VARIABLENAME="IndexCapTable" DEFAULT="" TYPE="SQL" DATATYPE="TEXT">SELECT TextValue FROM AsFund
```

CopyBook Used for Index Funds

Step 11: Configure Transactions

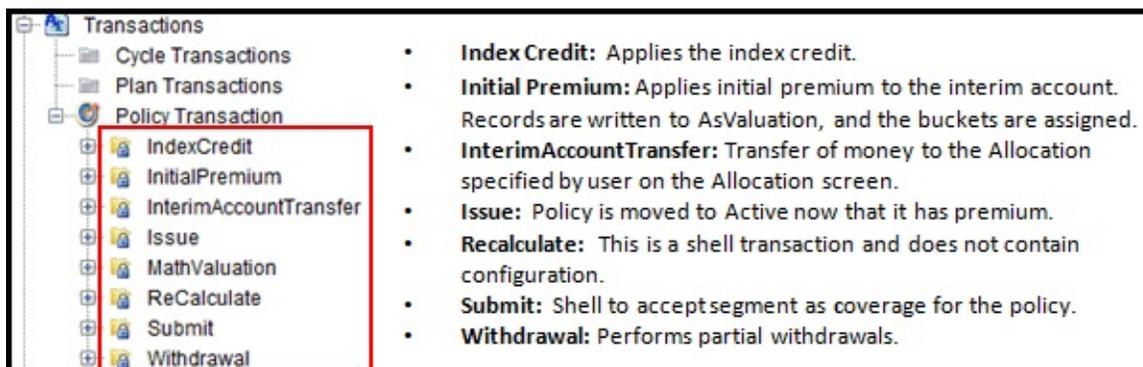
Transactions must be configured to move the policy through its lifecycle. Some examples of events that are controlled by transactions are issuing the policy, applying money in and transferring money from the interim account into index funds on the Sweep Date.

For a complete example of the basic transactions required, review the transactions in the Index Prototype Plan, which demonstrate transactions that support the design of an Index product. A base segment is also needed to support the full product lifecycle. Navigate to **Main Explorer | Companies | Prototype Company | Subsidiary Companies | Prototype Child Company | Plans | Index Prototype Plan | Transactions | Policy Transactions**.

When working with the Withdrawal transaction, notice that as money is moved out according to a removal precedence and removal method that was established during fund creation. PureLIFO and ProRataLIFO are the methods that must be used with index funds.

- **PureLIFO:** last in/first out removal method for index funds. Applies across all indexed funds with the same removal precedence removing from the newest bucket first.
- **ProRataLIFP:** last in/first out removal method for index funds. Pro-rate across indexed funds first and then removes from the newest bucket first in each fund.

Bucket removal is based on the BucketEffectiveDate attribute of Assignment. This information is held in the AsBucket table.



The screenshot shows a tree view of transactions in the Rules Palette. The 'Policy Transaction' folder is expanded, showing a list of transactions: IndexCredit, InitialPremium, InterimAccountTransfer, Issue, MathValuation, ReCalculate, Submit, and Withdrawal. To the right of the list, a bulleted list provides descriptions for each transaction.

- **Index Credit:** Applies the index credit.
- **Initial Premium:** Applies initial premium to the interim account. Records are written to AsValuation, and the buckets are assigned.
- **InterimAccountTransfer:** Transfer of money to the Allocation specified by user on the Allocation screen.
- **Issue:** Policy is moved to Active now that it has premium.
- **Recalculate:** This is a shell transaction and does not contain configuration.
- **Submit:** Shell to accept segment as coverage for the policy.
- **Withdrawal:** Performs partial withdrawals.

Transactions in Index Prototype Plan in Rules Palette

Step 12: Configure the ValueScreen Business Rule

The ValueScreen business rule must be configured for index funds. Refer to the [Value Screen](#) section for configuration information.

Related Topics

[Configure Funds](#), [Fund Database Tables](#), [Child Funds](#), [Currency Codes for Funds](#), [Fund Screens in OIPA](#), [Allocation Models](#)

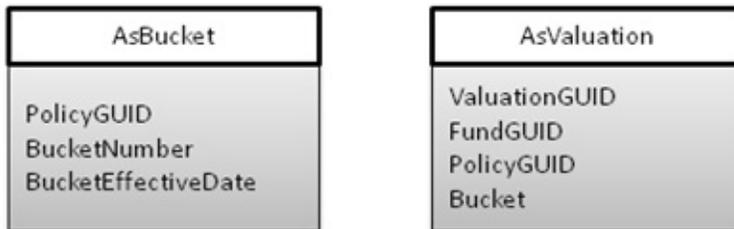
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Funds > Index Funds > Index Fund Database Tables

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Index Fund Database Tables

There are two database tables are important for indexed funds: AsBucket and AsValuation. AsBucket holds the number of premiums that are held, along with the policy and effective date information. The AsValuation table tracks money in and out using the bucket column along with FundGuid and other critical data.



Index Fund Database Tables

You are here: Configuration > Funds > Funds Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Funds Overview

Insurance products can be configured to use funds as an investment tool for the policy owner. Both variable and fixed funds are supported in OIPA. Funds must be defined and allocation structures must be configured to allow a user to select the way money is divided and applied to the available funds.

Different types of funds accrue money at various rates. A business rule can be configured to define fixed rates, or a separate system can be used to bring in the fund values for variable types of funds. Configuration is also needed to define how funds are valued for a policy.

Fund Types

Funds vary in the way they are valued. OIPA classifies fund types and defines the algorithm applied when performing valuation on the various types of funds. An explanation of the standard calculations performed on each fund types is provided below, along with the type code for the fund.

- **Fixed Funds (01):** Interest is calculated and added to the current amount invested in the fund to determine its value. This type of fund requires an attached interest rate business rule.
- **Variable Funds (02):** Fund value is established by taking the current total number of units and multiplying it by the current unit price.
- **Fixed Benefit Funds (03):** No interest is accrued on these funds, as they are used for fixed annuity benefit split activities and are excluded from the value of the policy. These funds are treated similarly to Non-Invest funds. An attached interest rate business rule is required.
- **Unitized Funds (04):** These are fixed funds that are valued using prices stored on the AsNetAssetValue table. These funds are allocated to policies using a single parent fund but individual deposits and withdrawals are made using sub-funds that are available based on activity effective date. When back dating activities with this fund type, the system will not calculate gain/loss amounts for the funds. If type 02 funds exist in the same activity valuation, then their gain or loss will be calculated and written only for the type 02 funds. Accounting and valuation records are associated to the parent funds while the underlying amounts are calculated using the invested lateral fund prices.
- **Non-Invest Funds (05):** These funds are treated as fixed funds in OIPA. They are not included when determining a policy's value.
- **Index Universal Life (09):** These funds are treated like fixed funds but the interest is calculated outside of the valuation (by the InterestRateCalculation and InterestRateCode rules) usually in transaction math or a function. The funds are given a zero percent interest rate in the above mentioned rules then the interest is calculated during the crediting period within a transaction at the end of the crediting period and deposited back into the funds to get the increase in value.
- **Unit Linked Funds (10):** Fund value is established by taking the current total number of units and multiplying it by the current unit price.



Fund types are defined in `AsCodeFundType` and should never be changed. Changing codes could adversely affect the valuation process.

Fund Asset Classes

OIPA offers the ability to organize funds into Fund Asset Classes. Examples of Fund Asset Classes are Fixed Income and Equities. Funds can then be associated to these classes and defined in an allocation model.

Benefit Funds

OIPA offers the ability to configure Benefit Split functionality. This allows a periodic payment from a variable annuity policy to pay out to more than one beneficiary. The underlying location of these payments is made from funds linked to a policy's current investment. To support this, benefit funds must be setup. Refer to the [Benefit Split](#) section for more information.

Valuation of Funds

Valuation is done at the parent level if no child funds are present. If child funds exist, then valuation is done at the child fund level. The OIPA Variable Annuity product template demonstrates how to configure valuation using parent funds. The OIPA Unit Linked product template demonstrates how to configure valuation using child funds.

The value of a fund is taken from the AsNetAssetValue table for variable funds. For fixed funds, the value is found in the InterestRateCalculation business rule.

Benefit funds are used for payout annuities or the annutization of a product with child funds. The benefit fund can be configured to use the unit fund value but then will subtract the assumed interest rate (AIR) charge that is configured via the FundScreen rule. So although the benefit funds are technically children of the child fund, their value is different than that of the child fund and can be used as the value that will be paid out.

Chart of Accounts and Funds

In OIPA you can setup a Chart of Accounts (CoA), which tracks in and out money movements. Tracking money in and out can be performed according to funds. Please see the [Chart of Accounts](#) section for setting up CoA records that will generate fund processing.

Redemption Fees and Funds

[Redemption fees](#) can be applied to a withdrawal based on redemption criteria. The FundScreen business rule must be configured to support two dynamic fields for funds: Redemption Factor and Redemption Duration. Each fund must also have a redemption value assigned. The RedemptionAmountFormula rule governs the calculation process. View the Redemption Fee Prototype for sample configuration.

Related Topics

[Configure Funds](#), [Fund Database Tables](#), [Child Funds](#), [Currency Codes for Funds](#), [Fund Screens in OIPA](#), [Allocation Models](#)

You are here: Configuration > Funds > Fund Configuration Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Fund Configuration Overview

Fund configuration will vary depending on the type of fund and the allocation structure.

High Level Steps to Configure Funds

1. [Configure the individual funds](#). Funds are created using the Funds editor in the Admin Explorer tab. Open **Admin Explorer | Administration | Funds** to view the existing funds by company.
2. Configure the [FundScreen](#) business rule. This business rule is where extra fields are configured for business specific data that must be stored at the fund level, child fund level, lateral fund level or benefit fund level.
3. Configure the rates for funds. This information tells OIPA how to calculate interest and where to look for variable fund values.
 - [InterestRateCalculation](#) business rule: This rule is used to define the type and details used to calculate interest for fixed funds.
 - **AsNetAssetValues** database table: This table stores the values for variable funds. There is no configuration for this, but values can be inserted to the table using [AsFile](#).
4. Configure the allocation structures for the funds. These structures define how money is applied to each fund.
 - **AllocationScreen**: This business rule contains the configuration of the funds allocation structure for a product at the plan, policy and segment level. This is where plan level allocations are configured. For policy and segment level allocations, this should only be used by V7 and V8 clients who want to upgrade or those using the [default method](#).
 - **Transaction**: The <Allocation> element in a transaction defines the fund allocation structures for an activity. This should only be used by V7 and V8 clients who want to upgrade or those using the default method.
 - **PolicyAllocationScreen**: This rule defines the allocation methods, funds and [allocation models](#) available for policy level allocations.
 - **TransactionAllocationScreen**: This rule defines allocation methods, funds and allocation models available for specific activities processed in OIPA.
 - **FundListForAllocation**: This is an attached business rule that allows configuration to control the list of funds in the Fund drop down box on the Activity screen.
 - **WriteDefaultAllocation**: This is an attached business rule that allows

a transaction to use default plan or policy level allocations.

- **ReassignAllocations**: This is an attached business rule where allocations can be configured to move to an assigned fund. For instance, with new policies all allocations might be moved into a low risk account/fund until a waiting period has expired ensuring money.
- **FundAllocation**: This business rule allows you to display the allocations for an activity on the Activity Details screen.

5. Configure Benefit Split for Variable Annuity payouts.

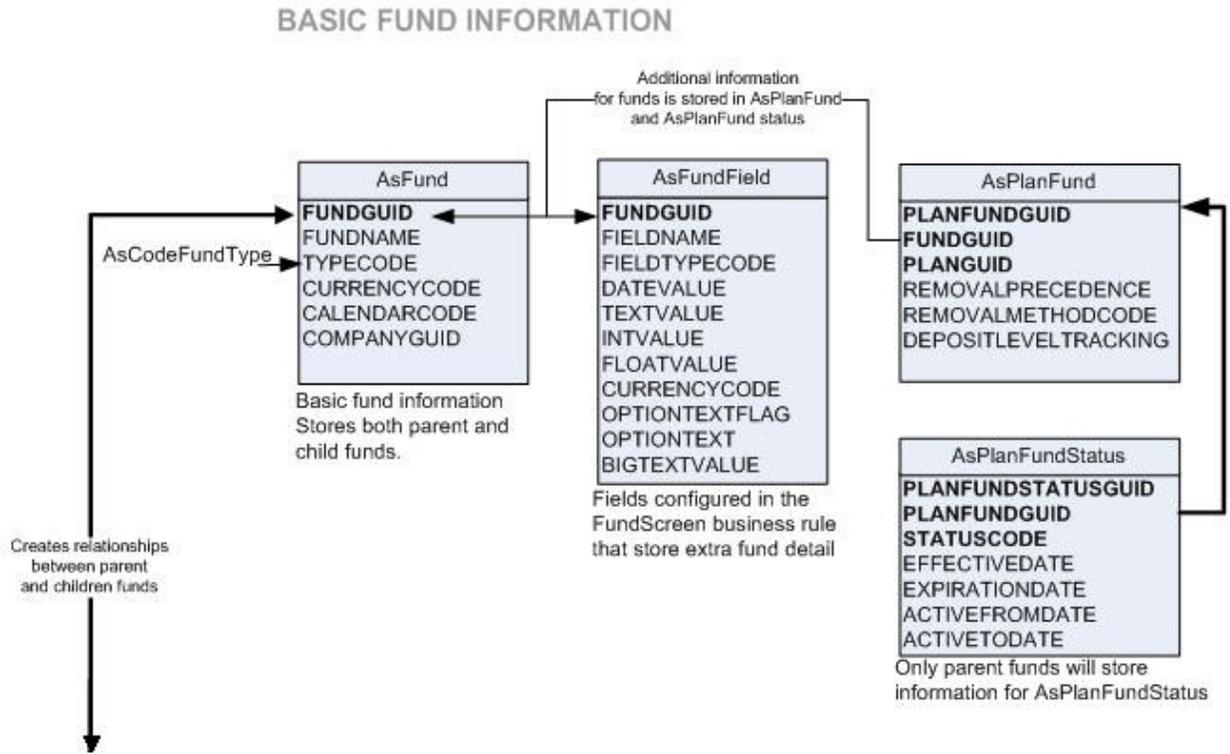
- **CalculateGeneral**: Use this calculate rule to configure benefit split allocations.
- **Transaction**: add the <BenefitSplit> element to the transaction that will perform the split. This element is used to specify how to calculate benefit split records based on specific allocations.
- **DoBenefitSplitChange**: This is an attached rule that can be added to a transaction to change benefit allocations.
- **ReassignAllocations**: This is an attached rule that can be added to a transaction to convert benefit split allocations to parent fund allocations when needed.

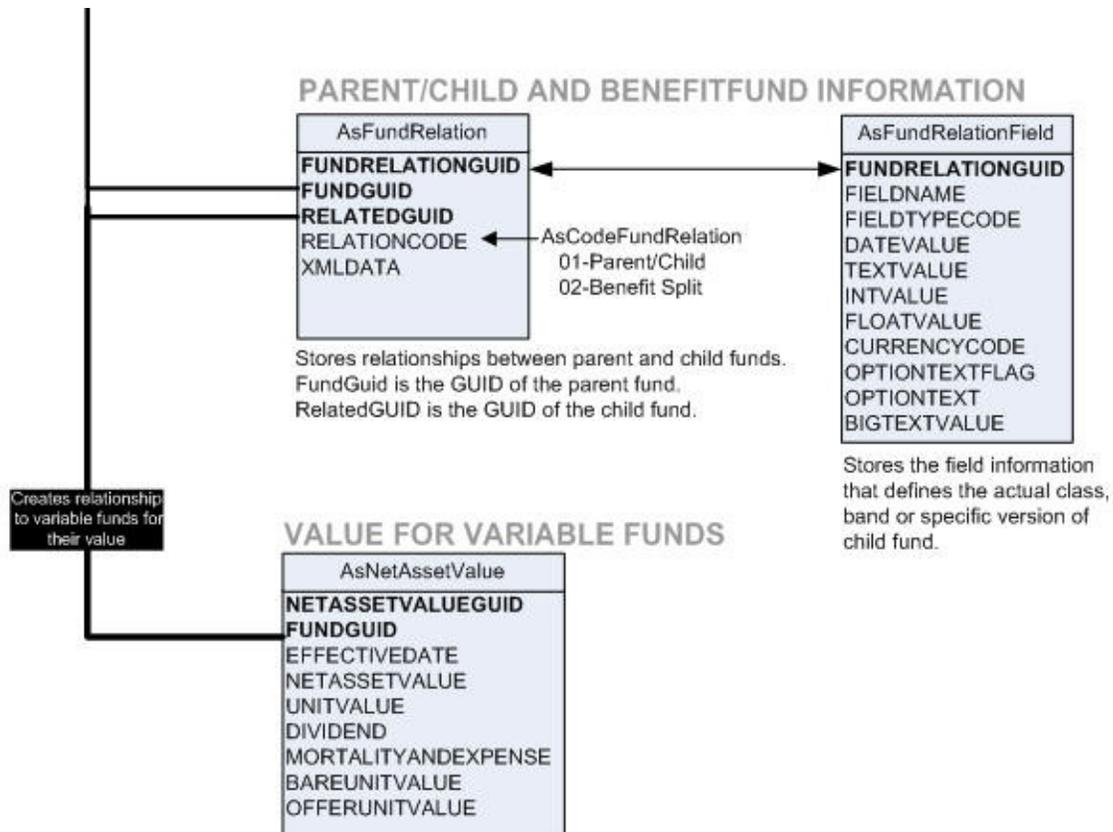
You are here: Configuration > Funds > Fund Database Tables

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Fund Database Tables

The image below shows the relationship of the database tables that contain fund information. The connection between funds and plans is shown as well as the relationship tables that store parent/child and benefit information for funds.





You are here: Configuration > Funds > Create Funds

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create Parent Level Funds

Organizing and Creating Funds

Funds are initially created in the Admin Explorer at the Primary Company level. After they are created at the Primary Company level, they may be associated with specific plans. This must be done prior to setting-up allocation structures, transactions or business rules that use the funds.

All fund administration is handled in the Admin Explorer in the Funds node, where fund details can be edited and funds can be associated to plans. A fund may be associated with multiple plans.

[Child funds](#), [lateral funds](#) and [benefit funds](#) can be created after a parent fund is set-up. Valuation will run on the child, lateral and benefit funds, unless none are created. Valuation only runs on parent funds when no child, lateral or benefit funds exist.

Basic Fund Database Tables

A complete explanation of the database tables involved in storing fund information is provided in the [Fund Database Table](#) section.



Due to the changes in the Fund table for the 9.5.0.0 GA release, fund functionality is not backwards compatible with previous V9 versions or previous V7 and V8 versions. The significant change to funds involved moving fund association from plan to company level. This eliminates the duplication of funds in multiple plans.

High Level Steps to Create Funds

The following list provides a general overview of the major steps involved in setting up funds for use in OIPA.

1. Configure the FundScreen business rule.
2. Create the new fund using the Fund wizard. This wizard is accessed from the Funds node in the Admin Explorer.
3. Enter fund information.
4. Associate funds with plans.
5. Set fund status, if applicable.

Step 1: Configure the FundScreen Business Rule

Before creating funds, the FundScreen business rule must be configured. This rule allows additional fields to be configured for parent level funds. The fields configured are used when entering fund information. They store custom information need by the business in the AsFundField table.

All fields needed by the various fund types must be configured in this rule. There is no way to filter fields by fund type in OIPA so they will all display for the user. If child funds, lateral funds or benefit split funds are needed, they must be defined in this rule as well.

Steps to Configure a FundScreen Business Rule

1. Navigate to the **Global Rules Explorer**.
2. Expand the **Business Rules** folder.
3. Expand the **Screen** folder.
4. Right-click the FundScreen folder and select **New FundScreen Override**.
5. Select **Next** in the first screen of the wizard.
6. Select the appropriate company from the **Company** drop down box.
7. Click **Finish**.
7. Open the new override rule in the Configuration Area and click the **XML Source** pane.
8. Configure the <ParentFundFields> element directly under the opening tag.
9. Add a <Field> element for each field that is needed for parent funds. Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Common Elements | Field Elements**.
10. Right-click on the FundScreen.xml file and select **Check-In**.

```
1 <FundScreen>
2   <ChildFunds ALLOWED="Yes" BENEFITFUNDS="Yes">
3     <Fund TYPE="01">No</Fund>
4     <Fund TYPE="09">No</Fund>
5     <Fund TYPE="02">Yes</Fund>
6   </ChildFunds>
7   <ParentFundFields>
8     <Field>
9       <Name>FundCode</Name>
10      <Display>Fund Code</Display>
11      <DataType>Text</DataType>
12    </Field>
13    <Field>
14      <Name>FundIndex</Name>
15      <Display>Fund Index</Display>
16      <DataType>Combo</DataType>
17      <Query TYPE="FIXED">
18        <Options>
19          <Option>
20            <OptionValue>NULL</OptionValue>
21            <OptionText>N/A</OptionText>
22          </Option>
23          <Option>
24            <OptionValue>SandP500</OptionValue>
25            <OptionText>S and P 500</OptionText>
26          </Option>
27          <Option>
28            <OptionValue>Russell2000</OptionValue>
29            <OptionText>Russell 2000</OptionText>
```

FundScreen Business Rule with ParentFundField Element

Step 2: Create Fund

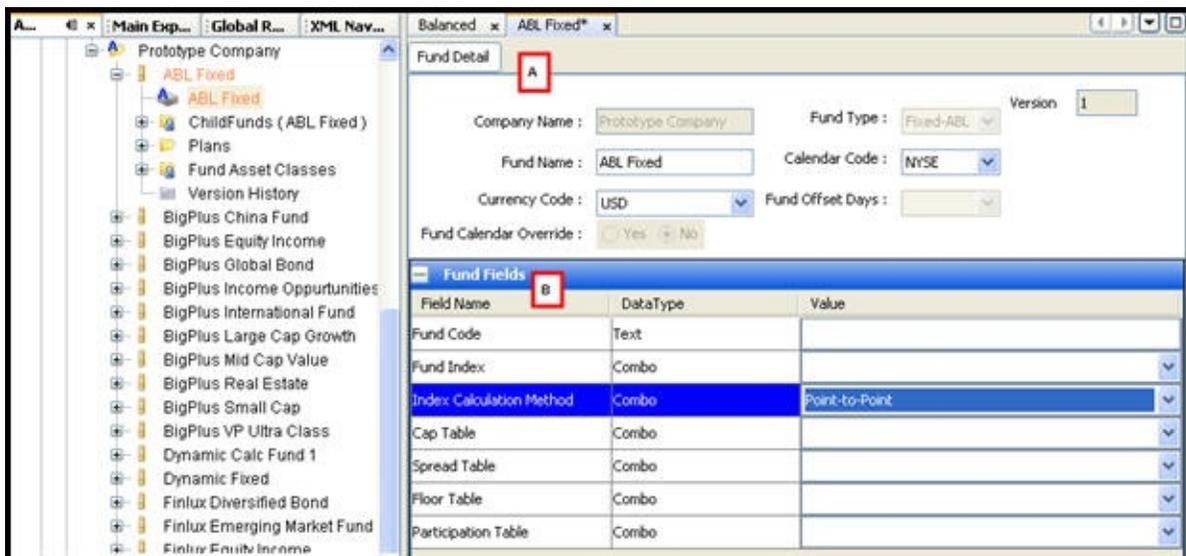
1. Navigate to the **Admin Explorer**.
2. Expand the **Administration** folder.
3. Expand the **Funds** folder.
4. Right-click on the Company level folder where the fund will reside and select **Add New Fund**. The New Fund wizard will display.
5. Enter the fund information. Each field is described below.
 - **Fund Name**: the name of the fund being created. Spaces can be used in the fund name.
 - **Fund Type**: the type of fund being created. Types are pulled from AsCodeFundType. These options can be edited through the Code Names editor in the Admin Explorer.
 - **Currency Code**: This is the currency associated with the fund. The default is the currency associated to the primary company or the system properties. This is populated by AsCurrency, which can be edited through the Currency editor in Admin Explorer.
 - **Calendar Code**: This is the calendar that identifies trading days for the fund. This is populated by AsCodeCalendar.
 - **Override Offset**: This check box is available for unit linked funds. Fund offset days can be defined when this box is checked.
 - **Fund Offset Days**: This drop down box becomes enabled when the fund is a unit linked type and the Override Offset box is checked. Enter the number of days OIPA should delay before processing the fund. When entered an entry is made for the fund in AsPriceOffset, which links to AsFund through a FundGUID.
6. Select **Finish**. If the Finish button is not enabled, required fund information may be missing. When complete, the Fund Detail pane opens for the new fund in the Configuration Area.

Step 3: Enter Fund Information

When the new fund displays in the Configuration Area, additional information can be entered in the fields that were created through the FundScreen business rule in step one.

Fields that do not apply to the fund type being created can be left blank. Since there is no validation supported, the user must determine the fields that apply to each fund type.

The image below shows a fund open in the Configuration Area. **Section A** contains the fund details, which are stored in AsFund and captured through the Fund wizard when the fund is first created. **Section B** shows the fund fields that were configured in the FundScreen business rule. The user enters information on the screen and when saved it is stored in AsFund.



Fund Information from Admin Explorer Funds Node

Step 4: Associate Funds With Plans

There are two ways to associate funds with plans. The first method only applies to parent funds, but the second method can be used by child funds, lateral funds, benefit funds or parent funds.

1. Select the fund from the Admin Explorer **Funds** node and open the **Plans | Plan List** folder to associate multiple funds to a plan.
2. Select the **Subsidiary Company** folder under Funds in the Admin Explorer and open the appropriate plan folder. The various fund levels each have their own folder where funds can be associated to the plan.

Regardless of the way the association between fund and plan is created, a database record is inserted into AsPlanFund, with any additional details in AsPlanFundStatus. In previous versions of the system, the information was stored in the AsFund database table.

Association 1: Fund to Plan

1. In the Admin Explorer, expand **Administration | Funds | Primary Company**.
2. Locate and expand the fund to associate with a plan.
3. Expand the **Plans | Plan List** folder.
4. Right-click the PlanList.xml file and select **Check Out**.
5. Use the right-facing arrow button to move the plans that apply into the **Attached Plans** window. To remove a plan, use the left-facing arrow button to move the plan back to the **Available Plans** window.
6. Under Plan Details, select **Yes** in the **Deposit Level Tracking** drop down box to specify that the fund should be tracked at the deposit level. This is typically used for point in time valuation. If **No** is selected, the fund will be tracked at the fund level. Fund level tracking is the default method. This means the policy will keep the values of each fund on the policy. When Deposit Level tracking is set to Yes, then the value of each individual deposit in a fund is tracked.
7. Under Plan Details, the removal method can be set by selecting an option

from the **Removal Method** drop down box. This tells OIPA how to remove money from a fund. The list is populated by AsCodeFundRemoval.

8. Under Plan Details, a removal precedence can be specified. This identifies the order in which money is removed from funds. Removal precedence is set by selecting an integer in the **Removal Precedence** drop down box. A value of zero (0) will not allow the fund to have money removed.
9. Check-in the file when all information has been added.

Association 2: Plan to Fund

1. In the Admin Explorer, expand **Administration | Funds**.
2. Expand the **Primary Company Name | Subsidiary Companies | Subsidiary Company Name | Plans | Plan Name**.
3. Find the folder for the type of fund (Parent, Child, Lateral or Benefit) then expand the folder, right-click on the fund file and select **Check Out**.
4. Use the right-facing arrow button to move the plans that apply into the **Attached Plans** window. To remove a plan, use the left-facing arrow button to move the plan back to the **Available Plans** window.
5. Under Plan Details, select **Yes** in the **Deposit Level Tracking** drop down box to specify that the fund should be tracked at the deposit level. This is typically used for point in time valuation. If **No** is selected, the fund will be tracked at the fund level. Fund level tracking is the default method. This means the policy will keep the values of each fund on the policy. When Deposit Level tracking is set to Yes, then the value of each individual deposit in a fund is tracked.
6. Under Plan Details, the removal method can be set by selecting an option from the **Removal Method** drop down box. This tells OIPA how to remove money from a fund. The list is populated by AsCodeFundRemoval.
7. Under Plan Details, a removal precedence can be specified. This identifies the order in which money is removed from funds. Removal precedence is set by selecting an integer in the **Removal Precedence** drop down box. A value of zero (0) will not allow the

fund to have money removed.

8. Check-in the file when all information has been added.

Step 5: Set Fund Status, If Applicable

When a status is applied to a fund, it controls by date when a fund is available for selection in OIPA. This is an optional feature, as funds don't have to be restricted by date. Plan fund statuses can only be set when the fund file is checked in.

If multiple plan statuses are needed, then there must be a one day difference between the expiration date of the old status and the effective date of the new status.

Steps to Set Fund Status

1. In the Admin Explorer select **Administration | Funds | Primary Company | *Fund name* | Plans | Plan Status List**.
2. Open the appropriate plan folder.
3. Right-click on the plan.xml file and select **Check Out**. The Plan Fund Status will open in the Configuration Area.
4. Click Add to create a plan status record.
5. Type the status information.
 - **Status:** this is the current status of the fund. Populated by AsCodeFundStatus. These options can be updated through the **Code Names** editor in the Admin Explorer.
 - **Effective Date:** this is the date when the fund becomes available for investors.
 - **Expiration Date:** this is the date when the fund expires and is no longer available for investors. This date must be later than the effective date.
 - **Active From Date:** this is a disabled field that displays the system date indicating when the AsPlanFundStatus record was saved and checked into the database.
 - **Active To Date:** this checkbox closes the status and populates the database with the current system date. A new status record can be created one the previous one has been closed.

6. Check-in the plan status file to save the information.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Valuation > Fund Types and OIPA Valuation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Fund Type and OIPA Valuation

The OIPA system performs standard calculations when valuing the following types of funds.

- **Fixed Funds:** Interest is calculated and added to the current amount invested in the fund to determine its value.
- **Variable Funds:** The current total number of units is multiplied by the current unit price to establish a fund's value.
- **Non-Invest Funds:** User may view the amount of a fund, but the value is not included in the policy's worth. These are treated as fixed funds by OIPA.
- **Unit Linked Funds:** The current total number of units is multiplied by the current unit price to establish a fund's value.

These types of funds are listed in AsCodeFundTypes and should never be changed as the system will use these specific typecodes for valuation.

Fund Type	TypeCode
Fixed Funds	01
Variable Funds	02
Non-Invest Funds	05
Unit Linked Variable Funds	10

Table of Fund TypeCodes

You are here: Configuration > Funds > Child Funds

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Child Funds

Child funds are used when the same fund is offered with different versions or bands. The [Parent fund](#) is always created first with the general information for the fund. Then specific child funds are created with the actual version of the fund that a client would elect in OIPA.

[Lateral funds](#) are another type of fund that can be associated with a parent fund. A parent fund can have a child fund or a lateral fund, but not both. Lateral funds and child funds are mutually exclusive.



If [benefit funds](#) are needed, then they are added after child funds as they must be attached to a child fund in a one-to-one relationship.

High Level Steps to Generate Child Funds

1. Configure the [FundScreen](#) business to support child funds and any extra fields needed for the fund information.
2. Make sure [Parent funds](#) were created using the Fund wizard in Admin Explorer.
3. Use the ChildFunds.xml file to create child funds.
4. Associate funds with plans.
5. Set fund status, if applicable.

Step 1: Configure the FundScreen business rule

The FundScreen business rule needs to be configured to indicate to the system that child funds will be allowed for a particular company. A <ChildFund> section is included at the beginning of the rule indicating whether child funds and benefit funds are supported and defining the types of parent funds that can have child and benefit funds. Usually only one field is created for child funds and the field is a combo field that stores the type of child funds that could potentially be created.

The <ParentFundFields> section holds fields for additional information for the fund, which carry over to any child or benefit funds.

1. In the **Global Explorer**, navigate to **Business Rules | Screen | FundScreen**.
2. Locate the **FundScreen** business rule in the Company Overrides folder or create a new override of this rule for the applicable company.
3. Configure the <ChildFunds> element.
 - Include the ALLOW="Yes" attribute.
 - Add a <Fund TYPE="##">Yes</Fund> element for each type of fund that needs to support child funds.
 - Add an <Fund TYPE="##">No</Fund> element for each type of fund that does not support child funds.
4. Configure the <ChildFundFields> element.

- Configure the <Field> elements for any the type of child funds that need to be created.

Step 2: Create Parent funds via fund wizard

Create the parent fund that the child funds will be associated with. This can hold general information, but clients will never be enrolled into this as a fund.

1. Navigate to the **Admin Explorer**.
2. Expand the **Administration** folder.
3. Expand the **Funds** folder.
4. Right-click on the Company Level folder and select **Add New Fund**. The New Fund Wizard will appear.
5. Type the fund information in the fields provided. Each field is described below.
 - **Fund name:** The name of the fund being entered.
 - **Fund Type:** This is the type of fund being created. Populated by AsCodeFundType. Edit these options through the Code editor in the Admin Explorer.
 - **Currency code:** This is the currency of the fund, which defaults to the currency code of the Primary Company or System Properties. Populated by AsCurrency, which can be edited in the Currency editor in the Admin Explorer.
 - **Calendar code:** This is the calendar that the fund is traded on. Populated by AsCodeCalendar. An Override Offset checkbox is available to selected if the fund type is Unit Linking. If this box is checked, Fund Offset Days can be selected directly below.
 - **Fund Offset Days:** This drop-down box is only enabled if the Fund Type is Unit Linked and the Override Offset is checked. Enter the number of days to use as the offset for processing. When entered, an entry is made for the fund in AsPriceOffset, which links to AsFund via a FundGUID.
6. Select **Finish**. If the Finish button is not enabled, required fund information

may be missing. When complete, the Fund Detail pane opens for the new fund in the Configuration Area.

Step 3: Use the ChildFunds.xml file to create Child Funds

The ChildFunds.xml file is used to actually create the child funds. It takes the <ChildFundField> information from the FundScreen configuration, lists all possible permutations of field values and provides a default child fund name. Using the check boxes, select the child funds to create.

1. Navigate to the **Admin Explorer** and open the **Company | Parent Fund** folder.
2. Expand the **ChildFunds** folder.
3. Check out the ChildFunds (name of fund).xml file.
4. In the **Create Funds** pane, default child funds are listed. They are created using the <Fields> information from the <ChildFundFields> in the FundScreen business rule. If needed, rename the default fund name for any child fund that needs to be created by selecting in the box and renaming the fund.
5. Check the select box for any child fund that needs to be created.
6. Right-click on the ChildFunds(Plan Name).xml file and select **Check in**. Child funds are automatically created in the Child Funds node in the Admin Explorer. Notice that the information from the Parent Fund is carried over to the Child Fund, which can't be changed, and the additional Child Fund Relation fields are listed in the selection from the Create Funds pane.

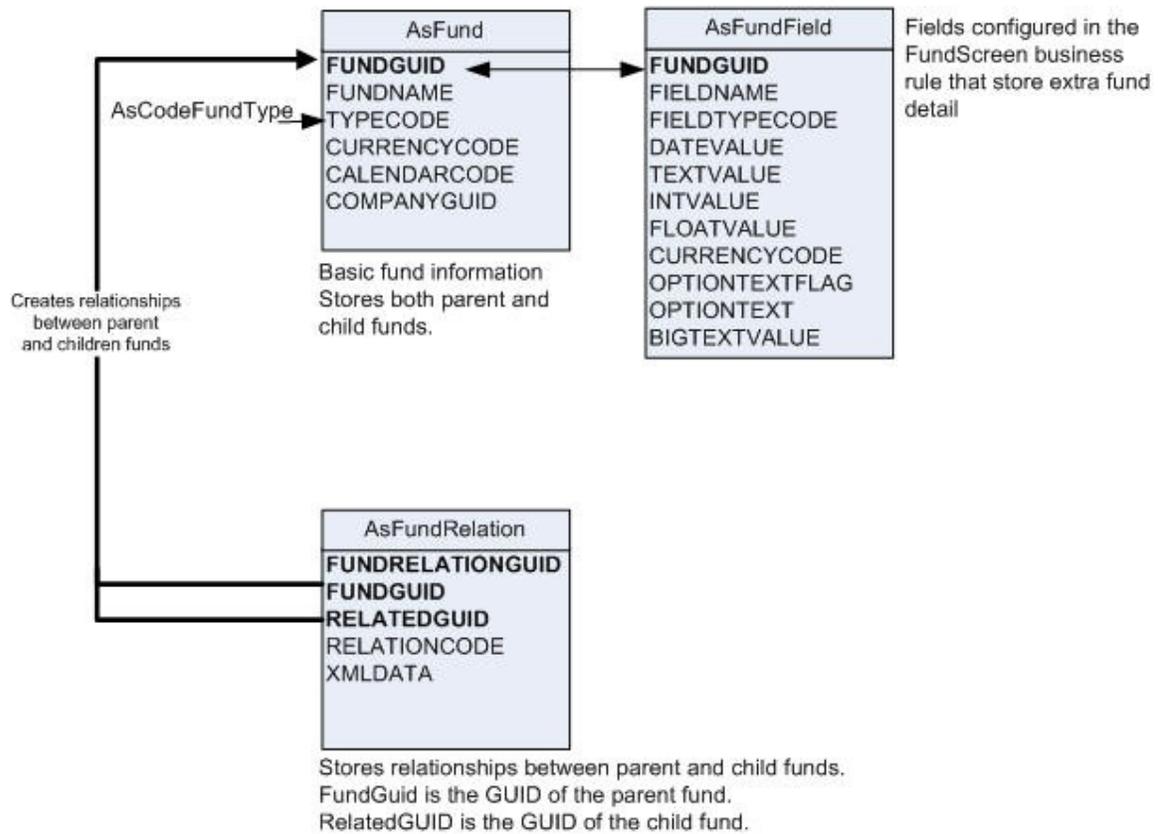
Step 4: Associate Funds with Plans

To associate child funds with a plan, use the Plan to Fund method. The parent fund must first be associated with the plan, before child funds can be associated. Navigate to the plan in the Funds node and make the associations there.

1. In the **Admin Explorer**, expand **Administration | Funds**.
2. Expand the Primary Company Name | Subsidiary Companies | Subsidiary Company Name | Plans | Plan Name.
3. Expand the Parent Funds folder.
4. Check out the Parent Fund xml file.
5. Select the funds from the **Available Funds** box and select the arrow to move them to the **Attached Funds** box.
6. Check in the file.
7. Expand the Child Funds folder.
8. Right-click on the Child Funds xml file and select **Check out**.
9. Select the Child Fund Relation Fields that are listed in the top box under Child Funds that should be available for the plan. Remember the Child Fund Relation Fields are the fields that were configured to define the type of class or band that the fund is going to be.
10. Select the **Apply Relation Filter**, which loads the child funds with matching relation field values selected in the Available Funds box. The list is an alphabetized multi-select list box of available child funds related to the parent funds associated to the plan.
11. Use the arrow buttons between the Available Funds and Attached Funds box to define the child funds that will be associated with the plan.
12. Check in when complete.

Step 5: Set Plan Fund and Plan Fund Status information

Plan fund information such as removal method, removal precedence and deposit level tracking are handled at the parent fund level. Plan fund status records are not created in the AsPlanFund table for child funds. All child funds adhere to the plan fund status information set for the parent fund.



OIPA Fund Database Tables

You are here: Configuration > Funds > Benefit Funds

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create Benefit Funds

Benefit funds are funds specifically used for annuitization and must be attached to a child fund. The [child fund](#) is always created first and there is a one-to-one relationship between the child and benefit fund.

High Level Steps to Generate Benefit Funds

1. Configure the [FundScreen](#) business to support child funds and any extra fields needed for the fund information.
2. Make sure [Parent funds](#) were created using the Fund wizard in Admin Explorer.
3. Make sure the [Child funds](#) were created.
4. Use the BenefitFunds.xml file to create benefit funds.
5. Associate funds with plans.
6. Set fund status, if applicable.

Step 1: Configure the FundScreen business rule

The FundScreen business rule needs to be configured to indicate to the system that child funds will be allowed for a particular company. A <ChildFund> section is included at the beginning of the rule indicating whether child funds and benefit funds are supported and defining the types of parent funds that can have child and benefit funds. Usually only one field is created for child funds and the field is a combo field that stores the type of child funds that could potentially be created.

The <ParentFundFields> section holds fields for additional information for the fund, which carry over to any child or benefit funds.

1. In the **Global Explorer**, navigate to **Business Rules | Screen | FundScreen**.
2. Locate the **FundScreen** business rule in the Company Overrides folder or create a new override of this rule for the applicable company.
3. Configure the <ChildFunds> element to support benefit funds.
 - ALLOWED="Yes" must be present in the <ChildFunds> element.
 - BENEFITFUNDS="Yes" must be present in the <ChildFunds> element.
 - Add a <Fund TYPE="##">Yes</Fund> element for each type of fund that needs to support child or benefit funds.

- Add an <Fund TYPE="###">No</Fund> element for each type of fund that does not support child or benefit funds.
4. Configure the <BenefitFundFields> element.
 - Configure the <Field> elements for any the type of benefit funds that need to be created.

Step 2: Make sure Child Funds are Created

Benefit funds must be associated with a child fund. Make sure the [child funds are created](#) before continuing with this process.

Step 3: Use the BenefitFunds.xml file to create Benefit Funds

The BenefitFunds.xml file is used to actually create the benefit funds. It takes the <BenefitFundField> information from the FundScreen configuration, lists all possible permutations of field values and provides a default benefit fund name. Using the check boxes, select the benefit funds to create.

1. Navigate to the **Admin Explorer** and open the **Company | Parent Fund | Child Fund** folder.
2. Expand the **BenefitFunds** folder.
3. Check out the BenefitFunds (name of fund).xml file.
4. In the **Create Funds** pane, default benefit funds are listed. They are created using the <Fields> information from the <BenefitFundFields> in the FundScreen business rule. If needed, rename the default fund name for any benefit fund that needs to be created by selecting in the box and renaming the fund.
5. Check the select box for any benefit fund that needs to be created.
6. Right-click on the BenefitFunds(Plan Name).xml file and select **Check in**. Benefit funds are automatically created in the Benefit Funds node in the Admin Explorer.

Step 4: Associate Funds with Plans

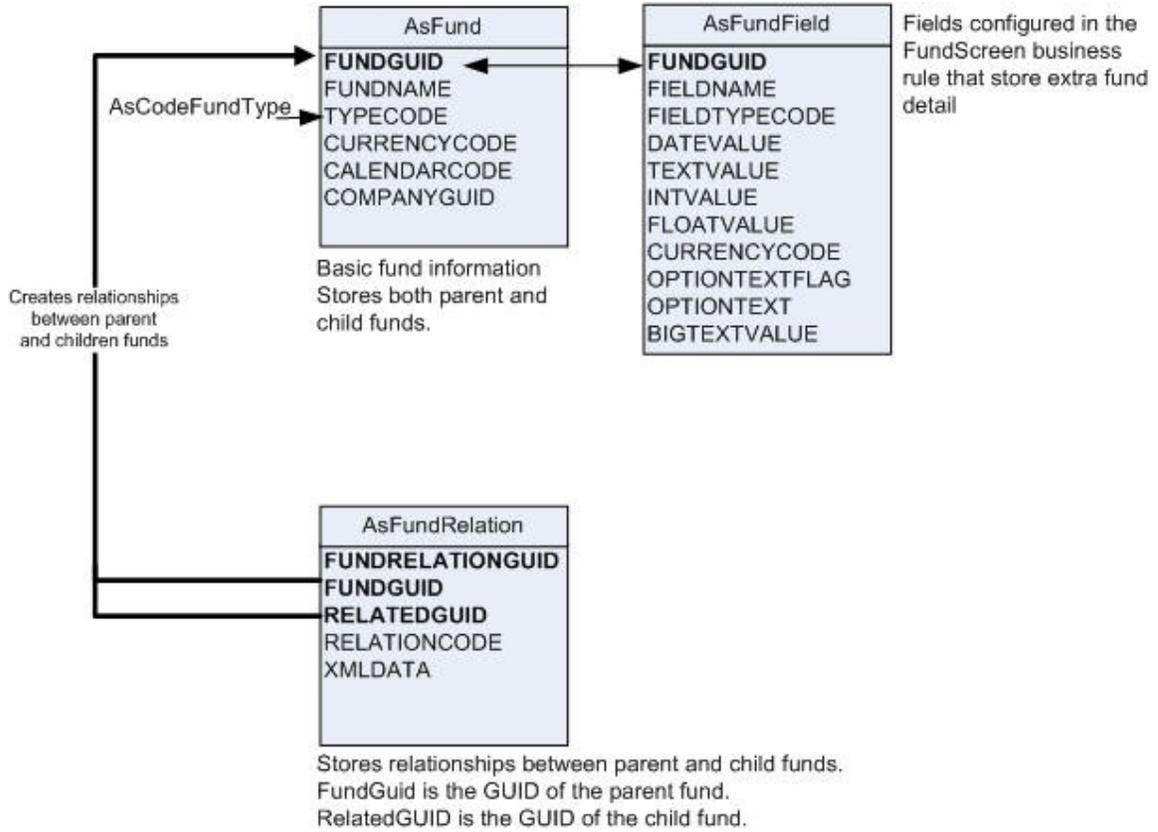
To associate benefit funds with a plan, use the Plan to Fund method. The parent fund must first be associated with the plan, before benefit funds can be associated. Navigate to the plan in the Funds node and make the associations there.

1. In the **Admin Explorer**, expand **Administration | Funds**.
2. Expand the Primary Company Name | Subsidiary Companies | Subsidiary Company Name | Plans | Plan Name.
3. Expand the Parent Funds folder.
4. Check out the Parent Fund xml file.
5. Select the funds from the **Available Funds** box and select the arrow to move them to the **Attached Funds** box.
6. Check in the file.
7. Expand the Benefit Funds folder.
8. Right-click on the Benefit Funds xml file and select **Check out**.
9. Select the Benefit Fund Relation Fields that are listed in the top box under Benefit Funds that should be available for the plan. Remember the Benefit Fund Relation Fields are the fields that were configured to define the type of class or band that the fund is going to be.
10. Select the **Apply Relation Filter**, which loads the benefit funds with matching relation field values selected in the Available Funds box. The list is an alphabetized multi-select list box of available benefit funds related to the parent funds associated to the plan.
11. Use the arrow buttons between the Available Funds and Attached Funds box to define the benefit funds that will be associated with the plan.
12. Check in when complete.

Step 5: Set Plan Fund and Plan Fund Status information

Plan fund information such as removal method, removal precedence and deposit level tracking are handled at the parent fund level. Plan fund status records are not created in the AsPlanFund table for benefit funds.

All benefit funds adhere to the plan fund status information set for the parent fund.



OIPA Fund Database Tables

You are here: Internationalization > Unit Linked Product > Market Calendars

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Market Calendars

Market Calendars can be managed through the Rules Palette. The [Code Names](#) editor in Admin Explorer is used to make the initial entries. When information is entered through this editor it is saved in the database in two locations: AsCodeCalendar and AsPriceOffset.

Steps to Add a New Calendar

1. Navigate to **Admin Explorer | Administration | Code Names | AsCodeCalendar**.
2. Click **Add**. A new row will appear highlighted in blue.
3. Click the Code Value field and type a code value.
4. Click the Short Description field and type a description.
5. Click the Long Description field and type a description.
6. Select a the number of days for the Business Date Offset.
7. Click the System Indicator box if this calendar is the default calendar. Only one calendar can have the System Indicator.
8. Click the Translate field to open the Translation window.
9. Type a language transaction for the calendar name. At least one translation must be entered for each row listed in the Translation window.
10. Click **OK** to close the Translation window.
11. Right-click on the AsCodeCalendar file and select **Check-in** to save the changes to the database.

If a calendar entry needs to be edited, check-out the AsCodeCalendar file and click in any field to make updates. The row will appear highlighted in green and the Edit checkbox will contain a checkmark to indicate that the information has changed. Check-in the file to save the changes.

If a calendar entry needs to be deleted, check-out the AsCodeCalendar file and click the Delete checkbox in the row that should be deleted. The row will appear highlighted in red. Check-in the file to save the information to the database. After the file is checked-in, the row will be removed. Removing information through this editor updates AsCodeCalendar and AsPriceOffset in the database.

AsCode

Code Name: Version:

Codes:

Code Value	Short Descript...	Long Description	Business Date Offset	System Indicator	Translate	Edit	Delete
01	NYSE	New York Stock Exch	0	<input checked="" type="checkbox"/>	<input type="button" value="..."/>	<input type="checkbox"/>	<input type="checkbox"/>
02	LXM	Luxembourg	0	<input type="checkbox"/>	<input type="button" value="..."/>	<input type="checkbox"/>	<input type="checkbox"/>
03	Korea	Korean Exchange	0	<input type="checkbox"/>	<input type="button" value="..."/>	<input type="checkbox"/>	<input type="checkbox"/>
04	Thailand	Thailand Exchange	0	<input type="checkbox"/>	<input type="button" value="..."/>	<input type="checkbox"/>	<input type="checkbox"/>
05	India	India Exchange	0	<input type="checkbox"/>	<input type="button" value="..."/>	<input type="checkbox"/>	<input type="checkbox"/>

AsCodeCalendar in Admin Explorer

You are here: Internationalization > Unit Linked Product > Fund Guarantee Price Offsets

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Fund Guarantee Price Offsets

In addition to a calendar offset, which applies to all funds associated with that particular calendar, there is also a way to add an offset to a particular fund. When the new fund is created, an option for Fund Calendar Override allows the user to specify additional offset days that apply to that particular fund.

In the New Fund wizard, there are four fields that control the fund price offsets.

1. **Fund Type:** UnitLinkedVariable must be selected in the Fund Type drop down box. This enables the Calendar Code, Fund Calendar Override and Fund Offset Days fields.
2. **Calendar Code:** Calendar codes exist in AsCodeCalendar. Each calendar code may have a designated business date offset, which would apply to any fund using that particular calendar. When a calendar code is selected from the drop down box, any existing offset associated with the calendar will automatically appear in the Fund Offset Days fields.
3. **Fund Calendar Override:** Yes enables the Fund Offset Days field. No leaves the field disabled.
4. **Fund Offset Days:** When the drop down becomes enabled, any existing business date offset that applies to the Calendar Code selected will display. Additional days can be added for the new fund. Additional offsets are saved to the AsPriceOffset table.

Activity with Calendar and a Fund Calendar Override

1. Activity is entered with effective date 1/9/2010 (Saturday)
2. The MarketMaker has a calendar (NYSE) with an offset of 1 business day.
3. The activity effective date is not a business day so the system looks to the next business day for NYSE and finds 1/11/2010 (Monday)
4. The calendar offset day is applied to Monday and the system returns 1/12/2010 (Tuesday).
5. Tuesday is the activity exchange rate date and the system will look for rates on AsExchangeRate for that date.
6. The fund attached to the activity has a calendar (HKEX) with an offset of 2 business days.
7. 1/12/2010 is a business day for HKEX so the system adds 2 day to get to 1/14/2010 (Thursday)
8. Thursday is the funds guaranteed price date and the system will look for the next AsNetAssetValue price equal to or greater than that date.

Activity with a Calendar That Has Additional Offset Days Added for the Fund

1. Activity is entered with effective date 1/9/2010 (Saturday)
2. The MarketMaker has a calendar (NYSE) with an offset of 1 business day.
3. The activity effective date is not a business day so the system looks to the next business day for NYSE and finds 1/11/2010 (Monday)
4. The calendar offset day is applied to Monday and the system returns 1/12/2010 (Tuesday).
5. Tuesday is the activity exchange rate date and the system will look for rates on AsExchangeRate for that date.
6. The fund attached to the activity has a calendar (NYSE) with an offset fund override of 2 business days.
7. 1/12/2010 is a business day for NYSE so the system adds 2 day to

get to 1/14/2010 (Thursday)

8. Thursday is the funds guaranteed price date and the system will look for the next AsNetAssetValue price equal to or greater than that date. AsPriceOffset will have two entries for calendar NYSE in this scenario. One where the FundGUID is blank, and one where it is populated.

The screenshot shows two side-by-side windows. The left window is titled "New Fund Wizard" and contains various input fields. The right window is titled "AsCodeCalendar" and displays a table of codes.

New Fund Wizard Fields:

- Company: International Child Company
- Plan: Unit Linked Template
- Fund Name: Silver
- Fund Type: UnitLinkedVariable
- Effective Date: [Empty]
- Fund Status: [Empty]
- Removal Precedence: [Empty]
- Removal Method: [Empty]
- Currency Code: [Empty]
- Calendar Code: NYSE
- Fund Calendar Override: Yes No
- Fund Offset Days: 1
- Deposit Level Tracking: Yes No

AsCodeCalendar Table:

Code Value	Short Description	Long Description	Business Date Offset
01	NYSE	New York Stock Exc	1
02	LXM	Luxembourg	0
03	Korea	Korean Exchange	0
04	Thailand	Thailand Exchange	0
05	India	India Exchange	0

Annotations:

- A red box highlights "New Fund Wizard" in the top left.
- A red box highlights "AsCodeCalendar" in the top right.
- Red boxes highlight "NYSE" in the "Calendar Code" field and "NYSE" in the "Short Description" column of the table.
- A red box highlights "1" in the "Fund Offset Days" field and "1" in the "Business Date Offset" column of the table.
- A red box contains the text: "Business offset is prepopulated in Fund Offset Days for Calendar Code selected".
- Red arrows point from the "Business offset" text box to the "Fund Offset Days" field and the "Business Date Offset" column.

New Fund Wizard and AsCodeCalendar with Calendar Fields Highlighted

You are here: Configuration > Allocations > Allocations and Assignments

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Allocations and Assignments

Allocations, Assignments and Valuation work together to move money between funds associated with a policy and track the value of the funds. Assignment allows money to be deposited, withdrawn or transferred from the funds. Allocation provides a request as to how to affect each fund, such as a deposit, withdrawal or transfer. Valuation is the result after money movements are completed.

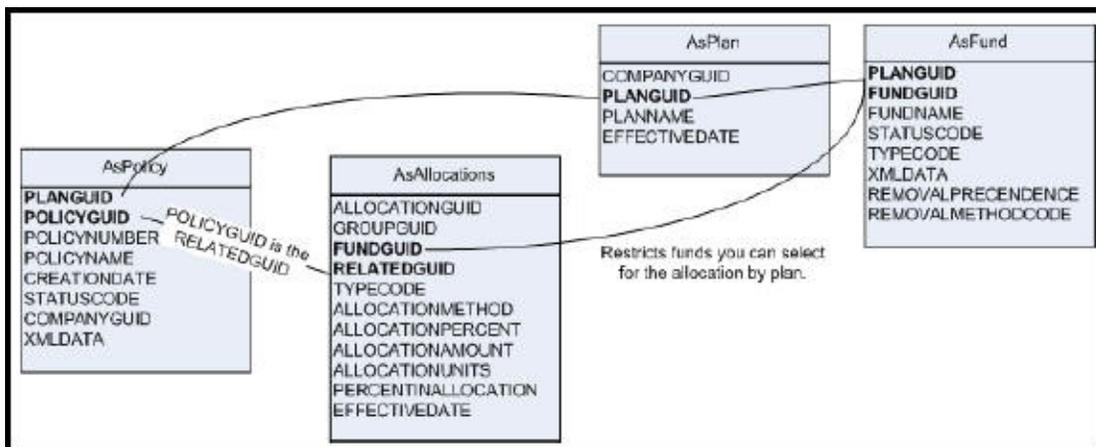
Allocations

Allocations support the movement of money into or out of funds. When payments or withdrawals are made, the configuration of allocations allows for the introduction of logic to move the money into or out of a fund.

Plan-level Allocations are allocations made at the plan level. For example, during the Freelook period all deposits are invested in Fund A until the end of the Freelook period is reached where it will be distributed to the funds specified by the client. The allocation to Fund A during the Freelook period is an example of a plan allocation. The plan-level allocations are specific to a plan, not a client or policy. Plan-level allocations can be established using the AllocationScreen business rule.

If Allocation data needs to be displayed, the AllocationScreen business rule must be configured.

Allocation data is stored in the AsAllocation database table. The AsAllocations table is linked to the AsPolicy, AsPlan and AsActivity tables through the RelatedGUID column.



Allocation Table and Related Tables

Assignments

Assignments are used in conjunction with allocations when the allocated values are to be applied, transferred, or removed from a policy's value. Assignment values can be in the form of a monetary amount, percentages, number of units or pro-rata distribution of the allocation. There are several types of assignments that can be used, namely:

- **APPLYBYFUNDTO**: used to apply money/units to a fund for a specific policy.
- **GROSSWITHDRAW**: used to remove money/units from a fund for a specific policy.
- **TRANSFER**: used to move money/units from one a fund to another fund for a specific policy.
- **SWITCH**: used specifically for unit linked products. This is similiar to the **TRANSFER** assignment, in that it moves money/units from one fund to another on a policy.
- **WITHDRAWAL**: supports the removal of amounts or percents from unit linked funds.

Additional assignment types can be found in the XML Configuration Guide.

Completion of Assignment processing always adds new records to AsValuation; the record of the money movement. The Assignment process will update existing allocation records on AsAllocation when it is executing a pro-rata distribution of the money movement. In all situations, if the business rule [ReassignAllocations](#) is attached to the activity, Assignment processing will add new records to AsAllocation based on the allocation results from ReassignAllocations.

Assignment Processing for Child and Lateral Funds

When Assignment looks up the necessary sub-funds (child and lateral) for valuation purposes, the criteria under the Child funds will only apply the funds with an AsFundRelation type of 01 and the criteria under the

Lateral funds will only apply the funds with an AsFundRelation type of 03. The Assignment will further filter the lateral funds by the activity effective date relative to the lateral fund relationship effective and expiration dates. Both fund relation dates are inclusive so that the relation is effective through the expiration date, not to it.

Assignments for child funds will use the fund that matches the fund relation criteria specified in the child fund level.

Deposit Assignments for lateral funds will use the fund that matches the fund relation criteria specified in the lateral fund level and whose effective date is less than or equal to the activity effective date and whose expiration date is null or greater than or equal to the activity effective date. If the EffectiveDateNUVMustExist element is set to **Yes** then all lateral funds for the parent matching the criteria (regardless of effective and expiration dates) must have prices for the activity effective date.

If NUVs are not found, then a business error is displayed saying, “NUVs for Effective Date Missing.”

The SystemDateNUVMustExist element will be ignored regardless if set to **Yes**, as no gain or loss will be calculated.

Removal Assignments will remove amounts allocated to type 04 parent funds from the individual lateral fund deposit amounts based on the parent fund's removal method (LIFO or FIFO) and removal precedence. The larger the removal precedence value, the later the fund will be used in money removal. Deposits of funds with a removal precedence of 1 will be exhausted before deposits on the same date for funds with a higher removal precedence value. This feature will work for all fund types.

Activity Valuation for Child and Lateral Funds

Activity valuation will calculate individual lateral fund deposit values and write the records using the parent fund GUID. The FundGUID of the lateral fund will be captured on the valuation tables in order for the system to map back to the correct fund prices during subsequent

valuations. (AsValuation, AsDepositValue). The EffectiveDate of the valuation record will be stored on the valuation table in order for the system to map back to the correct fund prices during subsequent valuations. (AsDepositValue). Only the parent fund name will display in the UI with an aggregated sum of all the lateral fund deposit valuations. Unit values will not display for the funds.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Funds > Redemption Fees for Withdrawals

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Redemption Fees for Withdrawals

Redemption fees can be applied to a withdrawal when the withdrawal occurs before the deposit has matured. OIPA can validate that the redemption fees apply to a withdrawal and can determine the amount to apply.

A prototype configuration has been created to demonstrate this feature. [View prototype.](#)

Configuration Required

The following configuration must be in place to support redemption fees.

- The **FundScreen** rule must contain dynamic fields for the redemption factor and the redemption duration. The name of these fields is important and must be referenced exactly in the RedemptionAmountFormula rule.
 - The **Redemption Factor** is used by the RedemptionAmountFormula rule to calculate redemption fees. This should be entered as a decimal value that indicates a percent. For example, 0.01 would be 1%. The default value is 0.00
 - The **RedemptionFee** is an existing dynamic field, which by default is set to 01. This indicates that OIPA should look at fund details to calculate fees.
 - The **RedemptionDuration** identifies the number of days the deposit must sit in the policy account before a withdrawal can occur. If the age of the deposit is less than the date when the withdrawal occurs, then the withdrawal is subject to redemption fees.
- The **RedemptionAmountFormula** rule defines the calculations that must be performed to determine the redemption fee. It must reference the Fund screen redemption factor and redemption duration field names exactly. The redemption factor and redemption duration are retrieved from the FundScreen rule. A math variable is identified that pulls the values using the prefix **Fund:** and then the name of the dynamic field from the FundScreen rule. For example, **Fund:RedemptionFactor** pulls the value from the Redemption Factor field on the Fund screen.
- A transaction must be configured with the assignment types that contain the redemption attributes REDEMPTIONFEE and REDEMPTIONMONEYTYPE.
- Funds that support the use of redemption fees must have Fund fields identified for Redemption Fee, Redemption Factor and Redemption Duration.

You are here: Configuration > Funds > OIPA Screens Supporting Funds

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

OIPA Screens Supporting Funds

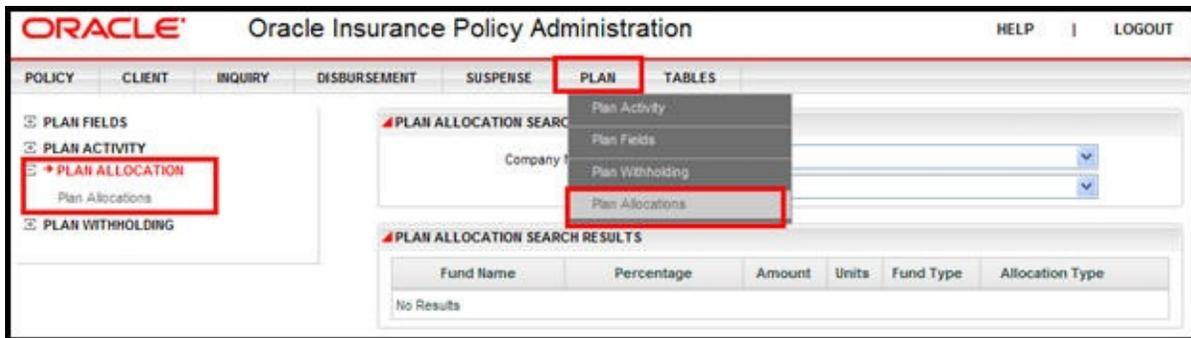
Configuration controls how fund information is made available in OIPA. Screens can be configured to present fund information to the user and to capture fund information when activities are processed against policies. Several examples of OIPA screens that display fund information are shown below.

In OIPA, the **Tables** option on the Main menu has a drop down option for **Unit Value**. Selecting this option reveals a table of net asset values for funds at various points in time. The Unit Value option will be available if security privileges for the **Net Asset Value** page in **Company Pages** are granted to the user.

Date	Bid	Bare Unit Value	Offer Unit Value
01/29/2031	14.14 USD	0.00 USD	0.00 USD

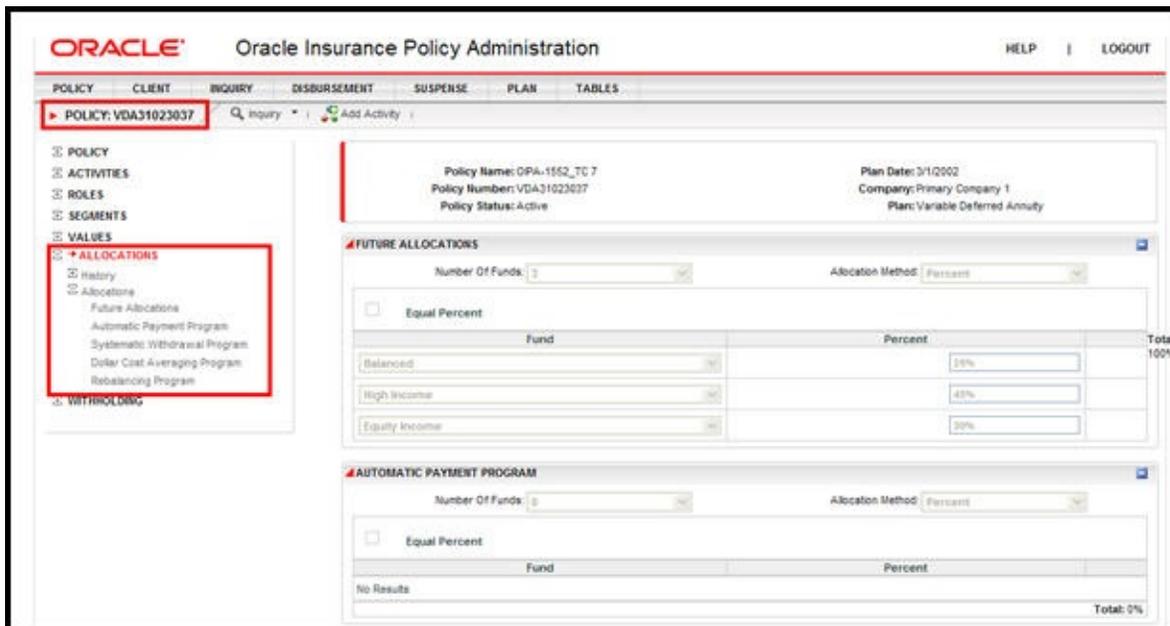
Tables Main Menu in OIPA with Unit Value Option Selected

The **Plan** option on the Main menu has a drop down option for **Plan Allocations**. Selecting this option provides search fields that allow a user to enter plan information and view the allocations for the plan. The AllocationScreen business rule is used to configure this screen.



Plan Allocation Option on Plan Main Menu

When a policy is open in OIPA, the Left Navigation menu has an **Allocations** link. When this link is clicked, the allocation options for the policy are displayed. The AllocationScreen business rule is used to define what displays when the Allocations link is clicked.



Allocation Link on Policy Screen

When an activity that is configured to capture allocations is added to a policy, the Add Activity window provides fields to capture allocation information. This is configured in the transaction using the <Allocation> element.

ORACLE Oracle Insurance Policy Administration HELP |

POLICY CLIENT INQUIRY DISBURSEMENT SUSPENSE PLAN TABLES

POLICY: VDA31024243 Inquiry Add Activity

POLICY

Policy Name: Plan Date: 1/29/2011
 Policy Number: VDA31024243 Company Name: Primary Company 1
 Policy Status: Active Plan Name: Variable Deferred Annuity

FUNDTRANSFER - VDA31024243

Details **Allocations**

ALLOCATIONS	
Fund Name	Cash Value
Equity Income (Band160)	7,070.00 USD
Balanced (Band125)	7,070.00 USD

ACTIVITY

FROM FUNDS

Number Of Funds: 0 Allocation Method: Amount

Fund	Number Of Funds	Amount
No Results		
		Total: 0.00 USD

TO FUNDS

Number Of Funds: 0 Allocation Method: Amount

Fund	Number Of Funds	Amount
No Results		
		Total: 0.00 USD

Allocation Link in Add Activity Window in OIPA

You are here: Configuration > Annuity with Overflow > Annuity with Overflow

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Annuitization with Overflow

Annuities are retirement products that may be used to help an individual increase savings or generate a guaranteed stream of income. Annuities that are used to increase savings are called deferred annuities. Annuities that generate a stream of income are called income annuities. Many times both types of annuities are used to create a single product that has an accumulation phase (deferred annuity) and a payout phase (income annuity).

When both products are used to create a single product, **annuitization** must occur to change the product's behavior from that of a deferred annuity to that of an income annuity.

The Annuitant can choose to receive the payments for life or for a specified period. In addition to choosing the length of time for the payments, the annuitant can choose the amount. The payments are driven by the types of underlying funds associated with the annuity. The funds associated with annuities can be either variable or fixed. Variable funds will be subject to market conditions and can vary during the accrual phase and the payout phase. Fixed funds on the other hand will increase by a specified percentage during the accrual phase and have a specific amount paid throughout the payout phase. If the annuitant outlives the period specified in the annuity contract (i.e. 20 Year Certain) payments will continue for the remainder of the annuitant's life.

In OIPA, annuity products use Benefit Split functionality to generate a policy's guaranteed payment stream upon annuitization. Funds may be variable, fixed or a combination of variable and fixed. This functionality as well as an enhanced Fund Allocation structure facilitates a seamless transition from the accumulation phase to the payout phase of a Variable Annuity product. The withdrawal transactions can also accommodate the situation when the annuitant outlives the contract through an overflow process.

Benefit Split records can be created for scheduled or nonscheduled

benefit payments or for annual benefit leveling. Activities can also be used to emulate these behaviors as needed. XML tags and attached business rules support this functionality



If Benefit Split functionality is used, then all funds on the policy must be set up for Benefit Split.

Components of Benefit Split Processing

There are three major components of the Benefit Split functionality:

- **creating (adding) Benefit Split records:** Benefit Split records are created in the global Calculate General rule on the [Benefit Split tab](#) in the visual editor.
- **reading (calculating) the value of Benefit Split records:** the [GetBenefitSplit Math Statement](#) is used to retrieve the benefit split records.
- **updating (changing) existing Benefit Split records:** the [DoBenefitSplitChange rule](#) is configured on the transaction initiating the change (typically Transfers, Purchases and Withdrawals). DoBenefitSplitChange supports variable to variable transfers as well as variable to fixed, and ABL purchases. In addition to changing the Benefit Split, it can also be reassigned from one fund to another.

Date	Type
1/1/2012	Deferred Annuity
5/4/2012	Deferred Annuity

BenefitSplit

Segment Amount: USD

Variable Benefit Amount: USD

Fixed Benefit Amount: USD

Allocation Type:

Merge Percent:

Calculate

OIPA Benefit Split on Segment Screen in OIPA

Code Names Needed for Benefit Split

The following code names and values must be created to support Benefit Split.

- **AsCodeFundType 03**: this code is for FixedBenefit. It is a zero interest, fixed annuity payout fund.
- **AsCodeBenefitSplitType 05**: this code is for Current Benefit Segment. It is used for active benefit split records that are paying out.
- **AsCodeBenefitSplitType 51**: this code is for Deferred Benefit Segment. It is used for deferred life contingent benefit split records.
- **AsCodeBenefitSplitStatus 01**: this code is used to indicate Active status. It is used for active benefit split records.
- **AsCodeBenefitSplitStatus 99**: this code is used to indicate Deleted status. It is used for segments that were deactivated due to recalculations and update activities.

Preliminary Configuration Steps

There are [four steps that must be taken before configuring Benefit Split](#).

1. The [Fund Allocation structure](#) must be established before using the Benefit Split functionality. Since child funds typically offer a choice of benefit funds it is important for this to be configured correctly.
2. Configure the [FundScreen](#).
3. [Parent funds](#) and [Child funds](#) must be configured. Benefit funds are always attached to a child fund in a one to one relationship.
4. The [Fund Screen must be associated with the plan and the funds](#).

High Level View of Benefit Split Configuration Steps

Once the preliminary configuration steps are complete and the business rules and code names have been created, the actual benefit split configuration can take place. The significant steps involved in this process are shown below. Follow the links for additional information on any step in the process.

Refer to the [Benefit Split Prototype](#) to see an example of all configuration described below.

1. Create the [Calculate General](#) business rule with the supporting Benefit Split fields.
2. Create the [SegmentName](#) rule for Benefit Split. This segment will point to the Calculate General rule that was configured in the previous step and include any dynamic fields that may be required by business processes.
3. Create [transactions](#). Benefit Split functionality can be accomplished using transactions (activities). Configure the Assignment Types that will be needed inside the transaction XML. There should be at least two transactions created; a payout transaction and a spawned disbursement. Additional transactions can be created to support other business needs for annuitization. Transaction elements are discussed in more detail in the section below.
4. Add Attached [business rules](#).

You are here: Configuration > Annuitization with Overflow > Preliminary Benefit Split Configuration Steps

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Preliminary Benefit Split Configuration Steps

There are four steps that must be taken before configuring Benefit Split.

1. The [Fund Allocation structure](#) must be established before using the Benefit Split functionality. Since child funds typically offer a choice of benefit funds it is important for this to be configured correctly.
2. [Parent funds](#) and [Child funds](#) must be configured.
3. Create the FundScreen. Refer to **Configure Fund Screen** below for additional information on this step.
4. The Fund Screen must be associated with the plan and the funds. Refer to **Fund Screen Association to Plan and Fund** below for additional information on this step.

Configure Fund Screen

The following list provides an explanation of the important elements needed in the Fund Screen configuration to support Benefit Split. A complete explanation of the elements and attributes are provided from the Rules Palette Help menu in the **XML Configuration Guide**.

- **ChildFunds**: this is an optional element that holds the attributes ALLOWED and BENEFITFUNDS. The value of the ChildFunds element will be a repeatable Fund sub-element.
 - **ALLOWED**: this is a required attribute that will accept a literal **Yes** or **No** with the default value being No. This attribute will indicate if child funds should be created from parent funds.
 - **BENEFITFUNDS**: this is an optional attribute that will accept a literal **Yes** or **No** with the default value being No. This attribute will indicate if benefit funds should be created from child funds.



The ALLOWED attribute must be set to **Yes** if the BENEFITFUNDS attribute is set to **Yes**. This will always be set to **Yes** for Benefit Split.

- **Benefit Fund Fields:** this element contains repeatable Field sub-elements. Field will use the existing field syntax and data types with the exception of Line, Label, Comment, Blank, Client, Address, Identifier and TextArea. Sub-Elements that are not supported include Expanded, Hidden and Disabled. BenefitFundFields is required if the ChildFunds element exists and the BENEFITFUNDS attribute is set to **Yes** (ALWAYS for Benefit Split). The Benefit Fund Field(s) is the filter that will be used to identify the correct benefit fund for a given child fund. The Child Fund Field(s) in the previous section of the Fund screen similarly identifies the appropriate parent fund descendent. The Benefit Fund Field is what is used as the FundRelation later in the configuration.

XML Example

```

<FundScreen>
  <ChildFunds ALLOWED="Yes" BENEFITFUNDS="Yes">
-
-(Parent and child fund field omitted for clarity)-
-
    <BenefitFundFields>
      <Field>
        <Name>InvestmentRate</Name>
        <Display>Investment Rate</Display>
        <DataType>Combo</DataType>
        <Query TYPE="SQL">Your Selection</Query>
      </Field>
    </BenefitFundFields>
  </FundScreen>

```

Fund Screen Association to Plan and Fund

After the Fund screen has been configured, the plan and fund are associated to the screen. The Rules Palette provides a visual editing interface to facilitate this task. From the **Admin Explorer** tab, open **Administration** and **Funds**.

Each company has a unique folder containing all funds available for use

by policies associated to that company. The funds are listed in alphabetical order. At the bottom of the list of funds is a **Subsidiary Company** folder, which contains a folder for each subsidiary company as well as folders for parent funds, child funds and benefit funds. A **Benefit Funds** node is created for each parent fund if the configuration in the FundScreen rule allows benefit funds for the parent fund type. Checking out the Benefit Funds node will open the **Plan Funds** editor.

Steps to Associate Fund Screen Rule to Plan and Fund

1. Navigate to the Admin Explorer tab and open **Administration | Funds | Primary Company Name | Subsidiary Companies | Subsidiary Company Name | Plan Name | Benefit Funds**.
2. Right-click the Benefit Funds file and select **Check-out**. This will open the Plan Funds editor.
3. In the top section of the editor, select a value from the options provided. These are populated from the <Field> section in the <**BenefitFundFields**> section of the FundScreen business rule.
4. Click the **Apply Relation Filter** to pull all available funds that meet the criteria selected in the first section.
5. Move Available Funds to the Attached Funds box to associate the fund with the plan and the FundScreen rule.
6. Right-click on the Benefit Funds file and select **Check-in** to save the changes. New records will be created on AsFund and AsFundField that replicate the values of the Parent fund.

You are here: Configuration > Annuity with Overflow > Benefit Split Rules

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Benefit Split Rules

The following rules are used to configure benefit split in OIPA. The CalculateGeneral rule should be configured first, followed by the SegmentName rule. The other attached rules are used to support transaction processing.

Required Rules

The following rules must be configured to support benefit split configuration.

- **Calculate General**: this calculate rule is used to calculate the Benefit Split. The override should be created at the plan level.
- **SegmentName rule**: this segment rule generates the Benefit Split record. The override should be created at the plan level.

Rules That Support Benefit Split Transaction Processing

The following rules can be attached to transactions to support benefit split processing in OIPA.

- **ReassignAllocations**: this attached rule writes the Allocation record to AsAllocation. The override should be created at the transaction level.
- **ReassignBenefitSplit** : this attached rule writes to AsBenefitSplit. The override should be created at the transaction level.
- **DoBenefitSplitChange** this attached rule writes to AsBenefitSplit. The override should be created at the transaction level.
- **DoSegmentRecalculations**: this attached rule writes to AsBenefitSplit. The override should be created at the transaction level.

Optional Rules

The following rules are optional and while not specific to benefit split, will support the activity-based creation of a new policy or segment based on events occurring on an existing policy.

- [CreateSegment](#): this attached rule will build the policy segment. The override should be created at the transaction level.
- [CreatePolicy](#): this attached rule will build the new policy. The override should be created at the transaction level.

Math Statement

- [GetBenefitSplit](#): this is the math function that is used to calculate the benefit split record for the activity calculation.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Annuity with Overflow > Benefit SplitTransaction Elements

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Benefit Split Transaction Elements

The following set of transaction elements can be added to a transaction to support Benefit Split. An example of a transaction that might require Benefit Split is Fund Transfer, but other business processes may require many others. The following elements are used in the same way that standard transaction elements are used.

Refer to the [Benefit Split Prototype](#) to see these elements integrated into a complete transaction configuration.

Transaction Elements

- <FundLevel>: optional element that indicates the funds for assignment to use in writing to AsValuation based on relation criteria to the parent fund.
 - LEVEL: required attribute that identifies the valuation relationship. Parent or Child are valid values. The default is Parent.
- <Relation>: this element contains relation keys to link assignment funds to the parent allocations. It is only required if LEVEL attribute above is set to Child.
 - <Criteria>: this is a repeatable sub element that identifies the fund field that should be matched.
 - NAME: this is a required attribute that identifies the Fund Field.
 - DATATYPE: this attribute specifies the data type of the criteria value and Fund Field (Value of the criteria is a
 - Value of the Criteria is a Field Name or literal value for the criteria name to match

Math Variables

- [GetBenefitSplit](#) math statement function reads parameters to calculate the benefit split record for the activity calculation.

Assignment

Assignment allows money to be deposited, withdrawn or transferred from the funds. [Allocation](#) provides a request as to how to affect each fund, such as a deposit, withdrawal or transfer. [Valuation](#) is the result after money movements are completed.

When the annuitant outlives the annuity contract term, Assignment may be used to allow additional withdrawals to be made. The Assignment type of **GrossWithdrawal** with the optional money type of **Fund Overflow** should be used in conjunction with the [GetBenefitSplit Math Statement](#), which calculates the annuity benefit payment and any overflow for a transaction. The Fund Overflow money type can also be used with

Assignment types of **GrossFullWithdrawalWithAllocations** and **GrossFullWithdrawal**. If the Fund Overflow is not configured on the withdrawal transaction and there is not enough money to make a withdrawal, then the system will generate an error.

In all cases, Assignment will use the allocations from the [ReassignAllocations](#) rule and map the [parent funds](#) specified to the valuation fund based on the Fund Level relation (Fund Level is an optional element that indicates the funds for assignment to use in writing to AsValuation based on relation criteria to the parent fund).

If an adjustment (Floor Guarantee) amount is to be added to the amount calculated by the GetBenefitSplit Math Statement, then separate ApplyByFund or RemoveByFund assignment types should be used where an adjustment fund for the amount could be specified outside the Allocations in the ReassignAllocations rule. It is assumed that the RemoveByFund assignment will have access to the values added by the ApplyByFund assignment within the same activity.

- RemoveByFund will first attempt to remove from the existing value of a fund before looking for an ApplyByFund within the activity.
- Assignments are processed in order. The Apply must exist in the configuration before the Remove in order to use this feature.

Fund overflow will occur if the policy value of the fund is less than the requested payment amount (by fund). An offsetting deposit equal to the shortfall of the fund will be made into the fund using the overflow money type.

You are here: Configuration > Annuity with Overflow > Benefit Split Calculations

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Benefit Split Calculations

When Benefit Split is configured, OIPA will calculate the value of the final allocation, which is an allocation element or a merged FinalAllocation element. The final allocation value is written to AsAllocation if it differs from the input specified in the Allocation element.



If existing active Benefit Split records exist when new ones are calculated, then the status of the existing records is changed to Deleted. This is the case anytime a new benefit split record is created (through Calculate General or an attached rule).

Variable Benefit

If the variable benefit is greater than 0 and benefit funds exist for variable parent funds then OIPA calculates each final parent fund (fund type 02) allocation and divides it by the sum of all variable final allocations to determine the **variable benefit fund allocation**.

The variable benefit fund allocation is multiplied by the variable benefit to determine the **variable benefit fund amount**.

The sum of the variable benefit amounts must equal the VariableBenefit element value.

All fund unit values will be as of the specified EffectiveDate. If values do not exist for the specified date, then OIPA will look up the last known unit values. If no unit values exist for the benefit fund, a system error is displayed.

Fixed Benefit

If the fixed benefit is greater than 0 and the fixed benefit fund is configured then OIPA assumes a fixed benefit ration of 100%. The fixed benefit fund allocation is multiplied by the fixed benefit to determine the **fixed benefit fund amount**.

The sum of the fixed benefit amounts must equal the FixedBenefit element value.

AsBenefitSplit records (type 05) are created by the benefit fund based on relation elements, (AIR, M&E, account codes, etc.). The variable benefit fund amount is divided by the benefit fund unit value (as of the BenefitSplit EffectiveDate) to determine the **benefit units**. The fixed benefit fund amount is passed directly to BenefitSplit and no benefit units are calculated.

Benefit Split Error Messages

If benefit funds do not exist for all variable parent funds based on relation logic, then a business error message will display.

If the FixedBenefit is greater than 0 and no FixedBenefitFund element exists or it has a null value, then a business error message will display.

If more than one benefit fund is found for each parent then a system error (stack trace) is displayed.

You are here: Configuration > Suspense > Suspense Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Suspense Overview

Suspense and accounting functionality (Chart of Accounts in Admin Explorer) can work in tandem or separately. Best practice is to implement both and have them work together to provide information on financial activities related to a company.

The Prototype Company provides an example of suspense processing that supports multiple currencies. View the [prototype example](#) for additional information.

How Suspense Works

Money can be posted directly to a policy or entered into a suspense record. Part of the suspense record or the entire record can be attached to a policy.

Accounting processes based on criteria indicators set on the [Chart of Accounts](#) and the optional ChartOfAccountsSpecifications rule. The ChartOfAccountsSpecifications rule allows the use of indicator values on the suspense record that control which suspense account(s) the money is apply to.

If money cannot be applied directly to a policy, then it is usually entered as a suspense record at the company level through the Suspense screen, which may also invoke account processing.

When the entire suspense amount is disbursed, the suspense item is closed. If only a portion of the suspense item is disbursed, then the suspense item remains open. Any disbursed amount is added to the suspense record's attached amount.

If the disbursement activity is reversed, then the amount of the disbursement is returned to suspense and the suspense is be reopened if it had been closed. The amount returned is subtracted from the suspense record's attached amount.

Suspense Accounts

A suspense account is an account that is used to temporarily store money until a decision is made about where the money will be allocated. It identifies where the money came from and allows the money to be used as payment to one or more policies. Suspense accounts are set-up in the Rules Palette from the Admin Explorer tab in the [Chart of Accounts](#) folder.

Suspense Records

Suspense records can be created from the Suspense screen in OIPA, or they can be generated by the GenerateSuspense business rule when it is attached to a transaction that is processed in OIPA. If a suspense record is created on the Suspense screen, it can then be selected on the Suspense Search tab of a transaction's Activity Detail screen for use by that transaction, provided the transaction is configured to display the Suspense Search tab.

A transaction may be both configured to display the Suspense Search tab of the Activity Detail screen and have the GenerateSuspense business rule attached. In this case, if a suspense record is selected from the Suspense Search tab, that record will be attached to the transaction. If a suspense record is not selected from the Suspense Search tab, then the GenerateSuspense business rule will be used to generate a new suspense record, which will then be attached upon the processing of the transaction.

A unique suspense number is generated with each suspense record for identification purposes. Suspense records are written to the suspense account(s) for a company's general ledger. These records temporarily hold money until an activity generates a process to disperse or apply the money. Suspense records are created in OIPA and are associated with existing suspense accounts.

High Level View of Suspense Configuration

The following list provides an overview of the major steps involved in setting up suspense in OIPA. Follow the links to pages for more information on each specific step in the process.

1. Set up suspense accounts in the Admin Explorer using [Chart of Accounts](#).
2. Configure Suspense Screens.
 - Configure the [Suspense Screen](#) and/or [Suspense Screen Overrides](#).
 - Configure the [Suspense Search Screen](#).
3. Configure [Suspense Refund](#)
 - Configure the transaction that will activate the refund. Typically client level activities are used for suspense refunds.
 - Configure the [suspense refund number](#) in the transaction.
 - Configure the suspense section in the transaction.
 - Configure the disbursement section in the transaction.
4. Configure Suspense Accounting.
 - Configure the [suspense or multisuspense element](#) in a transaction that is configured to apply money to policies from a suspense account. OIPA uses the suspense element to capture the suspense number and amount from an activity and apply it to the associated suspense record. The suspense record's Attached Amount is updated by adding the amount entered in the activity to the Attached Amount. The Attached Amount starts at zero, and once the amount equals the suspense amount, the record is closed.
 - Configure the [GenerateSuspense business rule](#) if suspense should automatically be generated for activities that move money into a policy. This rule creates accounting detail for a supplied field amount in order to establish a relationship between the amount and a suspense record.
 - Configure the MaintainSuspense business rule if the user needs to be able to change suspense field values and generate accounting through a collection of multiple suspense tickets. This rule should be attached to a transaction. Refer to the XML Configuration guide for a complete explanation of the elements and attributes for this rule.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Suspense > Suspense in OIPA

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Suspense in OIPA

The Suspense Screen in OIPA provides the means for creating new suspense records and searching for existing suspense records. The Suspense Screen can be accessed from the Main Menu.



Suspense Option on Main Menu

Each suspense record contains information for individual money entries. After selecting the Save button, the information is saved, the record status is open and a suspense number is automatically generated that will be used in activities to apply money to a policy.

Activities and Suspense

Activities that move money into the policy must be associated with a suspense record. When adding an activity such as Additional Payment, enter a suspense number for the funds. This is the number that was automatically generated when the suspense record was created. By entering a suspense number in an activity, the OIPA system verifies the record is still open, and adds that activity amount to whatever is in the attached amount. When the attached amount reaches the Suspense amount the system will close the suspense record.

There are two different suspense elements that can be configured in a transaction, either <Suspense> or <MultiSuspense>.

1. <Suspense> should be used when only one Suspense Number field is needed for processing. This is shown in the image below.
2. <MultiSuspense> should be used when more than one Suspense Number and associated amounts are needed for processing.

If a Suspense element is present in the transaction, then when it processes as an activity OIPA, the attached amount will be added or the user will receive an error if there is not enough suspense amount left to cover the money-in event.

ADDITIONALPAYMENT () - VDA31011742

name of new activity

Details | Allocations | Suspense

SUSPENSE ENTRY

Enter suspense number or search for a suspense number and then enter it

SuspenseNumber: 3474

SUSPENSESEARCH

Suspense Number: Amount

First Name: Last Name

BankName: AttachedAmount

Effective From Date: Effective To Date

Display: Open Close Pending Shadow

Find

Suspense link in New Activity Window

You are here: Configuration > Suspense > Configure Suspense > Overrides for Suspense Screens

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Overrides for Suspense Screens

Suspense Screens are normally configured at the Company Level. Overrides of the Suspense Screen rules can be created for different company needs.



Suspense Screen Override in Global Explorer

Overrides are created from the Global Explorer. Open **Business Rules | Screens** and look for the Suspense Screen. Right-click on the rule and select [New Suspense Screen Override](#). After an override is created, it will appear in the Company Override folder under the Suspense Screen in the Global Explorer. Check-out the override and configure it as needed.

You are here: Configuration > Suspense > Configure Suspense > Suspense Screen business rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Suspense Screen Business Rule

This business rule is used to create and control suspense records. Suspense records are used to track money. This business rule identifies where the money came from and allows for the money to be used as payment to various polices. A suspense record is used as a holding account until the money is applied or refunded. A unique suspense number is generated with the suspense record for identification purposes.

Additional Information

Suspense records are written to the suspense account(s) for the company's general ledger. These records temporarily hold transactions until an activity generates a process to disperse or apply the money. Once the respective activities are processed, the company's actual general ledger is updated. These transactions are specific to the inbound and outbound tracking of pieces of money received and disbursed by the insurance company.

For example, a customer may submit payments such as initial or additional premiums, service charges, etc., to the insurance company, along with the completed application. Within OIPA, processing activities like InitialPremium, AdditionalPayment would apply the money from the suspense account to the respective accounts. Similarly, processing activities like FreeLook -> Disbursement are used to disburse the money back to the customer from the suspense account.

Suspense in OIPA

OIPA automatically displays the Attached Amount, which is the total amount of the suspense record used by policy activities. The Status field is the status of the suspense record. Statuses can be found in AsCodeSuspenseStatus. The suspense number is automatically generated.

The screenshot shows a web form titled "SUSPENSE RECORD(S)". At the top, there is a dropdown menu for "Company" with "Holding Company" selected. Below this, the form is divided into three main sections, each highlighted with a colored box and a callout:

- Fixed Fields (Red box):** This section contains the following fields:
 - Policy Number: 10000003
 - Effective Date: 1/29/2031
 - Amount: 10,000.00 THB
 - First Name: (empty)
 - Last Name: (empty)
 - Account Number: (empty)
 - Check Number: (empty)
 - Bank Name: (empty)
 - Bank Number: (empty)
 - User: stevem3
 - Type: Check - Manual (dropdown)
- Configurable Fields (Blue box):** This section contains:
 - Statutory Company: ALIC (dropdown)
 - From Contract No: (empty)
 - UniqueID: (empty)
- Field data generated by the system after Save is clicked (Green box):** This section contains:
 - Attached Amount: 0.00 THB
 - Status: Open
 - Suspense Number: 3,474 (with a refresh icon)

At the bottom of the form, there are three buttons: "Save", "Find", and "Delete".

Suspense Screen Sections

Configuring the Suspense Screen

The SuspenseScreen business rule is used to configure the Suspense screen in OIPA. An explanation of the various sections of the business rule is provided below.

Opening/Closing Tag

The opening and closing tag of the business rule is shown below.

Element/Tag	Definition	Attribute	Element/Attribute Value and Description
<SuspenseScreen>	The opening and closing tag of the SuspenseScreen Business Rule.		

Suspense Screen XML Opening/Closing Tag

Fixed Fields

As with other screen rules, the display name of the fixed fields can be changed or hidden. The data saved in these fields is saved to the AsSuspense table.

XML Example

```
<FixedFields>
<Field>
<Name>TypeCode</Name>
<Display>Type</Display>
</Field>
</FixedFields>
```

Element/Tag	Definition	Attribute	Element/Attribute Value and Description
<FixedFields>	Optional Element; Allows configuration of fixed fields on a j2ee platform.		

Fixed Fields XML in Suspense Screen

Fields

The data saved in these fields is saved in the AsSupenseField table.

Standard events can also be configured for fields on this screen. Please see the Events section in the XML Configuration Guide for more information.

XML Example

```
<Fields>
<Field>
<Name>StatutoryCompany</Name>
<Display>Statutory Company</Display>
<DataType>Combo</DataType>
<Query TYPE="FIXED">
<Options>
<Option>
<OptionValue>ALIC</OptionValue>
<OptionText>ALIC</OptionText>
</Option>
</Options>
</Query>
</Field>
<Field>
<Name>FromPolicyNumber</Name>
<Display>From Contract No</Display>
<DataType>Text</DataType>
</Field>
</Fields>
<Events>
</Events>
```

Element/Tag	Definition	Attribute	Element/Attribute Value and Description
<Fields>	Required Element; Allows configuration of dynamic fields.		
<Field>	Required/Repeatable Element; The opening and closing tag for each Field being configured. <Name>, <Display>, <DataType>, <Disabled> and <Query> are the allowed elements for dynamic fields.		

Explanation of Field XML for Suspense Screen

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Suspense > Configure Suspense > Configuring Suspense Search

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Configuring Suspense Search

The Suspense Search screen is used to locate suspense records when the Find button is selected from the Suspense screen. The Suspense Search screen consists of two sections: Suspense Search and Suspense Results. The SuspenseSearchScreen rule is used to configure the Suspense Search section. The Results section is non-configurable at this time.

For the Search section, fields configured in the SuspenseScreen rule that are stored in AsSuspense and AsSuspenseField tables can be used as criteria to filter a suspense search. This rule automatically includes the Find and Close buttons and suspense status selections of Open, Closed, Pending and Shadowing that allow the end user to filter search results.

The screenshot displays the 'SUSPENSESEARCH' section with the following fields and controls:

- Suspense Number: 3,474
- Amount: [] USD
- First Name: []
- Last Name: []
- BankName: []
- AttachedAmount: [] USD
- Effective From Date: []
- Effective To Date: []
- Display: Open Close Pending Shadow
- Find button

The 'SUSPENSERESULTS' section shows a table with the following data:

SuspenseNumber	EffectiveDate	LastName	FirstName	Amount	AttachedAmount
3474	1/29/2031			10,000.00 THB	0.00 THB

Maximum Results: 10

Suspense Search Screen

Refer to the XML Configuration Guide for a list of all supported elements, attributes and values that can be used to configure the Suspense Search screen. Select **Help** from the Rules Palette Main Menu, then XML Configuration Guide. When the guide opens, select **Business Rules | Screen Rules | Suspense Search Screen**.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Suspense > Configure Suspense > Configure Transaction to Generate Suspense

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

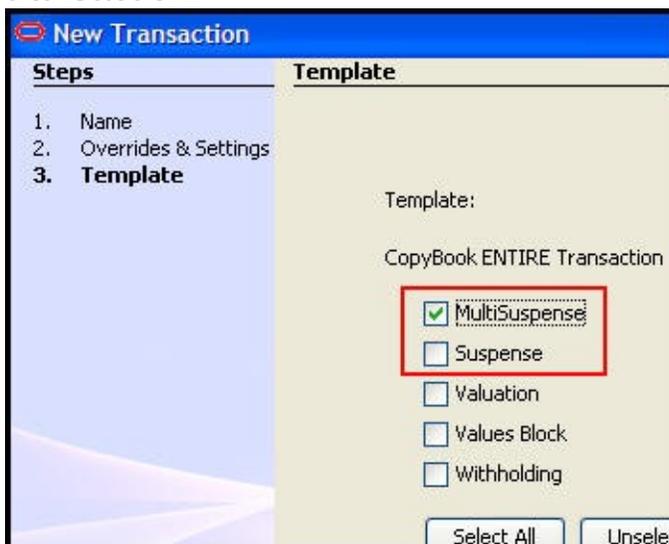
Configure Transaction to Generate Suspense

Transactions can be configured to automatically generate suspense records when they are processed as activities in OIPA. The [GenerateSuspense business rule](#) must be attached to the transaction in order for a suspense record(s) to be automatically generated.

When a transaction is created, suspense or multisuspense sections can be added. This is done by checking the box next to the suspense type needed. When the new transaction opens in the Configuration Area, the General Pane will display a section for the suspense type selected. Select the values for the suspense attributes. The XML Source Pane can be used to view the actual XML for suspense.

Note: If the <Suspense> tag is present in the transaction configuration then the GenerateSuspense attached business rule will not execute. The <Suspense> tag in the transaction takes precedence.

The Rules Palette has two validations in place around multisuspense. The START value must always be less than the STOP value and they both must be defined. A FIELD value, which is the name of the suspense field, must also be defined. The [Engine Error Output window](#) will display error messages if either of these conditions are not met in the new transaction.



The image shows a software interface with a tabbed menu at the top containing 'General', 'Allocations', 'Fields', 'Events', and 'Math'. The 'General' tab is selected. Below the tabs are several expandable sections: 'Effective Date', 'Values Block', 'Withholding', 'Valuation', 'Suspense', and 'MultiSuspense'. The 'MultiSuspense' section is expanded and highlighted with a red box. It contains a table with the following data:

MultiSuspense Attributes	Value
START:	2
STOP:	5
FIELD:	

Suspense Options on New Transaction Wizard MultiSuspense section of Transaction General Pane

You are here: Configuration > Suspense > Configure Suspense > Suspense Elements

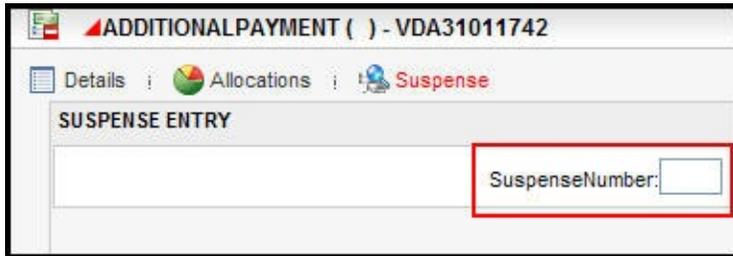
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Suspense Elements

When a transaction is configured with suspense, the <Suspense> or <MultiSuspense> element is used. Suspense should be used when only one Suspense Number field is needed for processing. MultiSuspense should be used when more than one Suspense Number and associated amounts are needed for processing. A transaction may have a <Suspense> element or a <MultiSuspense> element, but not both.

Suspense Element

If a <Suspense> element is present in the transaction, a Suspense tab will be available on the Activity Details screen with a Suspense field and a suspense record will be written in AsSuspense when the activity processes.



Suspense field on Suspense link of the Additional Payment Activity

Element/Tag	Definition	Attribute	Element/Attribute Value and Description
<Suspense>	This is the start and end tag of the element.		Value is a string that identifies the ActivityField. The suspense items must total.
		AUTOENTRY	Yes or No.
		OVERRIDABLE	Yes or No.
		VALUE	Sufficient. The Suspense record(s) must be at least the value of the Activity Field (can be greater).

Suspense Element

Multisuspense Element

If a <Multisuspense> element is present in the transaction, a Suspense tab will be available on the Activity Details screen with a Suspense section and multiple suspense records will be written in AsSuspense when the activity processes.

Multisuspense element



Multisuspense Element

You are here: Configuration > Suspense > Suspense Refunds > Suspense Refunds

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Suspense Refunds

Suspense refund functionality is demonstrated in OIPA using client level activities. When a client level activity is processed in order to create a suspense refund, configuration is present to require that OIPA automatically generate a [suspense refund number](#). The final piece of the process requires that a disbursement activity be processed to return the money to the client.

The [SuspenseRefundDisbursement prototype](#) demonstrates the configuration that can be used to accomplish this refund process.

Suspense

When the <Suspense> element is present in the transaction, a Suspense tab will be available on the Activity Details screen, which will contain a Suspense Number field, along with a Suspense Record search criteria and result sections. When the activity is processed, the suspense record selected to be fully or partially refunded, will be updated to reflect the refunded portion in the Attached column.

Disbursement

The transaction that triggers the refund may also contain a disbursement element.



An alternate configuration is to spawn the disbursement from the activity when it is processed in OIPA.

The optional element <DisbursementClient> was added to the standard Disbursement configuration. The value of this element is the math variable or field that holds the ClientGUID. The <DisbursementClient> element will be ignored if present in the configuration of a policy level transaction. If <DisbursementClient> is not present in the client disbursement transaction configuration then a stack trace is thrown. The <DisbursementRole> element will be ignored if present in the client configuration.

The disbursement requires approval if the disbursement configuration of the transaction uses the Disbursement element attribute APPROVAL. This attribute allows for values of Yes and No. A Yes value indicates that the pending disbursement will be displayed in the Disbursement Approval screen. A value of No indicates that the pending disbursement will not be displayed in the Disbursement Approval screen.

When approval has been stipulated in the transaction disbursement configuration, the disbursement is written to AsDisbursementApproval with a status of Unapproved. Approval of the disbursement will write the Approval Date to this same table. Additionally, the status of the disbursement is changed to reflect Approved. Disapproval of the disbursement will write the Disbursement Disapproval Code to AsDisbursementApproval and change the disbursement status to Disapproved. On the Disbursement Approval screen, security for the OK button is based on security role.

Disbursement recovery for disbursements from a client activity will

process exactly like disbursements from a policy level activity. Unlike recovery processing at a policy level where the shadowed disbursement for a RoleGUID on the policy is summed, for recovery processing on a client, the shadowed disbursements for the ClientGUID will be summed. A summing up of disbursements across different suspense numbers will take place when a client has multiple disbursements from different suspense numbers. This may result in a shadowed disbursement from Suspense 1 recovered from Suspense 2, for example.

High Level Steps to Set Up Suspense Refunds

The following list contains the major steps involved in setting up suspense refunds. Follow the links to find additional information on the steps involved in this process.

1. Configure the transaction that will activate the refund. Refer to the [prototype example](#) for additional information on this transaction. Client level activities are typically used to generate suspense refunds.
2. Configure the [suspense refund number](#) in the transaction.
3. Configure the suspense section in the transaction.
4. Configure the disbursement section in the transaction.

You are here: Configuration > Suspense > Account Records for Transactions with Suspense

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Account Records for Transactions with Suspense

When an activity processes that has suspense configuration, the system will locate the suspense record through the suspense number saved with the activity. The system will compare the amount entered in the activity to the amount left in the associated suspense record.

- If the amount entered in the activity is less than or equal to the remaining suspense amount it will add the amount to the attached amount and leave the suspense records status as open.
- If the attached amount is equal to the original suspense amount, the status of the suspense record is set to closed.
- If the amount entered is more than the suspense amount an error will generate at the activity level indicating there is not enough funds in the suspense record to apply to the policy.

Only one record for suspense is used during the suspense record's life, which is continuously updated with attached amounts until the record is in a closed status. Additions and reverses will happen to the one record according to the activity processing.

There are constant hard code suspense statuses that cannot be configured. However, additional statuses can be created and stored in the AsCodes table via AsCodeSuspenseStatus. Additional statuses may be needed if the GenerateSuspense rule is used to configure the creation of suspense records for transaction processing.

Hard Coded Suspense Statuses

01: NotApplicable

02: Pending

12: Deleted

15: Open

16: Closed

Additional Suspense Status Examples

03: Processed
04: Reversed

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Suspense > Suspense Database Tables

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Suspense Database Tables

There are several database tables that contain information related to suspense accounts and suspense records. All suspense account information is saved in the AsChartOfAccount tables. All suspense record information is saved to AsAccountingDetail and the four AsSuspense tables. An explanation of the database tables is provided below.

Suspense Accounts

Suspense accounts are created through Chart Of Accounts in the Rules Palette. The AsChartOfAccountsEntry table holds information about the account.

AsChartofAccountsEntry
CHARTOFACCOUNTSENTRYGUID
CHARTOFACCOUNTSENTITYGUID
DEBITCREDITCODE
ENTRYDESCRIPTION
ACCOUNTINGTYPECODE
ACCOUNTINGAMOUNTFIELD
GAINLOSSFLAG
FLIPONNEGATIVEFLAG
EFFECTIVEFROMDATE
EFFECTIVETODATE
FORWARDACCOUNTINGSTATUSCODE
REVERSALACCOUNTINGSTATUSCODE
DOREVERSALACCOUNTINGFLAG
ORIGINALDISBURSEMENTSTATUSCODE
FUNDTYPECODE
ACCOUNTNUMBERFORMAT

AsChartofAccountsEntry Database Table

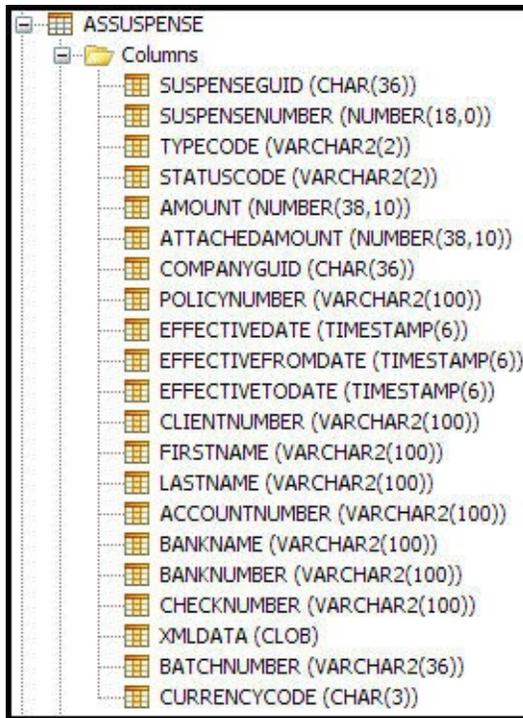
- **AsChartofAccountsEntry:** The Debit/Credit entry for an account that must be tied to transaction or suspense.
- **ChartOfAccountsEntryGUID:** Unique identifier for this table.
- **ChartOfAccountsEntityGUID:** GUID to the AsChartofAccountsEntity table.
- **EffectiveFromDate:** Effective Date the entry will be effective from.
- **EffectiveToDate:** Effective Date the entry will be effective to.
- **DebitCreditCode:** Indicates 01 for debit or 02 for a credit.
- **EntryDescription:** Description of the debit or credit.
- **AccountingTypeCode:** Stores the code for the way the accounting amount will be obtained. 01 for Total of Funds, 02 for ByFund, 03 for MathVariable, 04 for suspense, 05 for disbursement.
- **AccountingAmountField:** Name of the MathVariable to use as the amount if AccountingTypeCode is set as a type MathVariable. For disbursements, this is automatically set to **DisbursementAmount**.
- **GainLossFlag:** Indicates if the debit/credit amount will be calculated based on gain/loss values.
- **FlipOnNegativeFlag:** Flips credits to debits and vice-versa when an amount is negative.

Suspense Records

Suspense records can be created from the Suspense Screen in OIPA or they can be generated by the GenerateSuspense business rule when it is attached to an activity that is processed in OIPA.

Suspense Record Created from Suspense Screen

When a new suspense record is created in OIPA through the Suspense Screen, a record is written to AsSuspense.



The image shows a screenshot of a database tool displaying the structure of the AsSuspense table. The table is named 'ASSUSPENSE' and contains the following columns:

Column Name	Data Type
SUSPENSEGUID	CHAR(36)
SUSPENSENUMBER	NUMBER(18,0)
TYPECODE	VARCHAR2(2)
STATUSCODE	VARCHAR2(2)
AMOUNT	NUMBER(38,10)
ATTACHEDAMOUNT	NUMBER(38,10)
COMPANYGUID	CHAR(36)
POLICYNUMBER	VARCHAR2(100)
EFFECTIVEDATE	TIMESTAMP(6)
EFFECTIVEFROMDATE	TIMESTAMP(6)
EFFECTIVETODATE	TIMESTAMP(6)
CLIENTNUMBER	VARCHAR2(100)
FIRSTNAME	VARCHAR2(100)
LASTNAME	VARCHAR2(100)
ACCOUNTNUMBER	VARCHAR2(100)
BANKNAME	VARCHAR2(100)
BANKNUMBER	VARCHAR2(100)
CHECKNUMBER	VARCHAR2(100)
XMLDATA	CLOB
BATCHNUMBER	VARCHAR2(36)
CURRENCYCODE	CHAR(3)

AsSuspense Table

If dynamic fields are configured in the Suspense Screen then records are written to the AsSuspenseField table when a suspense record is updated/created.

Column Name	Data Type
SUSPENSEGUID	CHAR(36)
FIELDNAME	VARCHAR2(100)
FIELDTYPECODE	CHAR(2)
DATEVALUE	TIMESTAMP(6)
TEXTVALUE	VARCHAR2(255)
INTVALUE	NUMBER(10,0)
FLOATVALUE	NUMBER(38,10)
CURRENCYCODE	CHAR(3)
OPTIONTEXTFLAG	NUMBER(10,0)
OPTIONTEXT	VARCHAR2(1020)

AsSuspenseField Table

If dynamic fields are configured in the Suspense Screen and multiple values are present then records are written to the AsSuspenseMultiValueField table when a suspense record is updated/created. The FIELDINDEX Column identifies the multiple values.

Column Name	Data Type
SUSPENSEGUID	CHAR(36)
FIELDNAME	VARCHAR2(100)
FIELDINDEX	NUMBER(10,0)
FIELDTYPECODE	CHAR(2)
DATEVALUE	TIMESTAMP(6)
TEXTVALUE	VARCHAR2(255)
INTVALUE	NUMBER(10,0)
FLOATVALUE	NUMBER(38,10)
OPTIONTEXTFLAG	NUMBER(10,0)
OPTIONTEXT	VARCHAR2(510)
CURRENCYCODE	CHAR(6)

AsSuspenseMultiValueField Table

Transaction that Generate Suspense Records

Any time accounting is performed on a transaction that is configured with the suspense element, the information is saved to the AsAccountingDetail table. A suspense account must exist. The account is identified by the CHARTOFACCOUNTENTRYGUID.

ASACCOUNTINGDETAIL	
Columns	
ACCOUNTINGDETAILGUID	(CHAR(36))
ACTIVITYGUID	(CHAR(36))
SUSPENSEGUID	(CHAR(36))
FUNDGUID	(CHAR(36))
CHARTOFACCOUNTSENTRYGUID	(CHAR(36))
ACCOUNTNUMBER	(VARCHAR2(100))
DEBITCREDITCODE	(VARCHAR2(2))
AMOUNT	(NUMBER(38,10))
STATUSCODE	(VARCHAR2(2))
GAINLOSSFLAG	(VARCHAR2(2))
DOREVERSALACCOUNTINGFLAG	(VARCHAR2(2))
REVERSALACCOUNTINGSTATUSCODE	(VARCHAR2(3))
ENTRYDESCRIPTION	(VARCHAR2(510))
ENTRYDATE	(TIMESTAMP(6))
ACTIVEDATE	(TIMESTAMP(6))
ENTRYGMT	(TIMESTAMP(6))
CURRENCYCODE	(CHAR(3))

AsAccountingDetail Table

You are here: Configuration > Withholding > Withholding Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Withholding Overview

Withholdings are specific amounts or percentages that can be held from disbursements that are subject to state and federal taxes. Withholding only applies to plans with taxable events, such as a Variable Annuity.

Withholding can be defined at four levels: plan level, policy level, activity level and client level. The WithholdingScreen business rules defines the fields that are used to capture withholding information. All levels of withholding reference this rule to determine the fields displayed to the OIPA user when withholding information is captured. The withholding information entered into those fields is saved in two [withholding database tables](#). The AsWithholding table holds the withholding record GUID and the level of withholding. The AsWithholdingFields table holds all the actual withholding values.

Plan Level Withholding

[Plan level withholding](#) defaults are set when the plan data is configured in the Rules Palette. The Withholding tab of the Plan Data editor shows the fields configured in the WithholdingScreen business rule. Values entered into those fields are saved in the AsWithholdingField table. Plan level withholding is applied to policies and activities with the withholding configuration present in the transaction XML `<withholding>Yes</withholding>`.

Policy Level Withholding

[Policy level withholding](#) values are captured in OIPA. The WithholdingScreen business rule defines the fields that will appear to the user in OIPA. A Withholding link will appear in the Left Navigation Menu, if the [Policy screen](#) was configured to include the Withholding button and if proper security privileges belong to the user.

When an OIPA user selects the Withholding link at the policy level, the fields displayed are the fields configured in the WithholdingScreen business rule. The values entered are saved in the AsWithholdingField table, and the policy level designation is saved in AsWithholding. Policy level withholding takes precedence over plan level withholding and will apply to all activities with the withholding configuration present in the transaction XML `<Withholding>Yes</Withholding>..`

Policy level withholding can be locked down to prevent a user from entering withholding values when the policy is in a specified status. The optional `<DisableWithholdingFields>` element can be used in the WithholdingScreen configuration to accomplish this lock down of Policy level withholding.

Activity Level Withholding

Withholding can be defined at the activity level in configuration. This configuration makes a Withholding window available on the activity. An activity's withholding details can be set to process according to policy level withholding or activity level withholding. If activity level withholding is supported, then a Withholding link will appear in the Add Activity window when the activity is added.

Client Level Withholding

Client level withholding values are captured in OIPA. The WithholdingScreen business rule defines the fields that will appear to the user in OIPA. A Withholding link will appear in the Left Navigation Menu if the proper security privileges belong to the user.

Client level withholding is applicable when money is being disbursed at a client level via a client level activity. If client withholding has been established and activity withholding has been configured, the Withholding Type combo field on the activity withholding window would have client and activity values available. In this scenario, when activity is selected, the fields hold the client withholding values but are enabled and available for update.

CopyToWithholdingFields Business Rule

The [CopyToWithholdingFields attached business rule](#) can be configured and attached to a transaction. Then when the transaction is processed as an activity in OIPA, data can be passed from the activity to fields in AsWithholdingField.

The <Test> element within <Tests> can be used to set conditions for the update of the Withholding fields of a policy or client. When multiple test conditions are configured to determine if a pending activity or activities Withholding fields should be updated, all conditions must be met in order for the update to be made.

CopyToPendingActivityFields Business Rule

The [CopyToPendingActivityFields business rule can](#) be used to update withholding field values in policy activities that have not yet become active on the system and have not processed the transaction math, specifically, those activities in a Pending status.

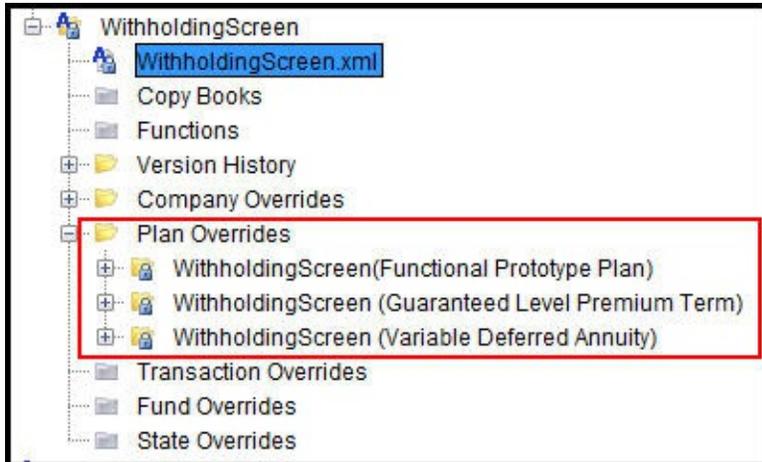
Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Withholding > Plan Level Withholding

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Plan Level Withholding

Plan level withholding is defined via a two step process in the Rules Palette. First, a plan level override of the WithholdingScreen business rule is created. The fields that will capture plan level withholding information are configured in this rule override.



Plan Overrides of the Withholding Screen Business Rule

The second part of the process requires the user to enter withholding field values via the [Plan Data file](#) in the Rules Palette. The Withholding tab displays the fields that were configured in the plan level override of the WithholdingScreen business rule. Plan level withholding is established by setting these field values. Withholding information is saved to the AsWithholding and AsWithholdingField tables.

View Withholding in OIPA

Any user with security privilege can access plan level withholding information by selecting **Plan Withholding** from the **Plan Main** menu.

To grant a user security access to view Plan Withholding in OIPA, open **Admin Explorer | Security | Application Security | Security Groups** and double-click the security group associated with the user. Then open **Company Security | Company Pages** and check out the PlanWithholding page. Check the box for Plan Withholding then check in the file.

▲ PLAN WITHHOLDING SEARCH CRITERIA	
Company Name:	Primary Company 1 ▼
Plan:	Guaranteed Level Premium Term ▼
▲ PLAN WITHHOLDING VALUES	
Federal Percent: 0%	Federal Amount: 0.00 USD
State Percent: 0%	State Amount: 0.00 USD
Opt-Out:	State Form Returned:

Plan Withholding Data in OIPA

You are here: Configuration > Withholding > Policy Level Withholding

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Policy Level Withholding

The [PolicyScreen](#) business rule is configured with a Withholding button, which allows the Withholding Left Navigation link to be available to OIPA users with proper security access. The WithholdingScreen business rule is configured with the fields that are presented to the OIPA user for entering Withholding values when the Withholding link is clicked from the Left Navigation menu on a Policy screen.

WithholdingScreen Business Rule

The WithholdingScreen business rule has a dual rule context: PrimaryComapny and Plan. These contexts define the available overrides allowed for the rule. Rule overrides allow for variation in configuration of the same rule based context. Refer to the [CopyBook Override Context](#) section for a description of available contexts.

Withholding Values

WithholdingField values are stored in the AsWithholdingFields table. Refer to the [Withholding Database section](#) for additional information.

Lock Withholding Fields

Withholding fields at the policy level can be locked by policy status. The optional **<DisableWithholdingFields>** element can be used in the WithholdingScreen configuration to disable Policy level withholding. Multiple **<DisablePolicyStatus>** elements can be used to define more than one status where the withholding fields are unavailable for update.

Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Screen Rules | Withholding Screen**.

The screenshot displays the Oracle Policy Administration interface. The top navigation bar includes 'ORACLE' and 'Policy Administration'. Below this, there are tabs for 'POLICY', 'CLIENT', 'DISBURSEMENT', 'SUSPENSE', and 'TABLES'. The main content area shows details for 'POLICY: VDA31011900'. On the left, a sidebar lists navigation options: POLICY, ACTIVITIES, ROLES, SEGMENTS, VALUES, ALLOCATIONS, and WITHHOLDING (highlighted). The main area contains the following information:

- Policy Name: [Blank]
- Policy Number: VDA31011900
- Policy Status: Active
- Plan Date: 03/01/2002
- Company: Primary Company 1
- Plan: Variable Deferred Annuity

The 'WITHHOLDING DETAIL' section includes the following fields:

- Federal Percent: 0%
- State Percent: 0%
- Opt-Out: No Yes
- Federal Amount: 0.00 USD
- State Amount: 0.00 USD
- State Form Returned: No Yes

A 'Save' button is located at the bottom right of the form.

Withholding link on a Variable Deferred Annuity Policy in OIPA

You are here: Configuration > Withholding > Activity level Withholding

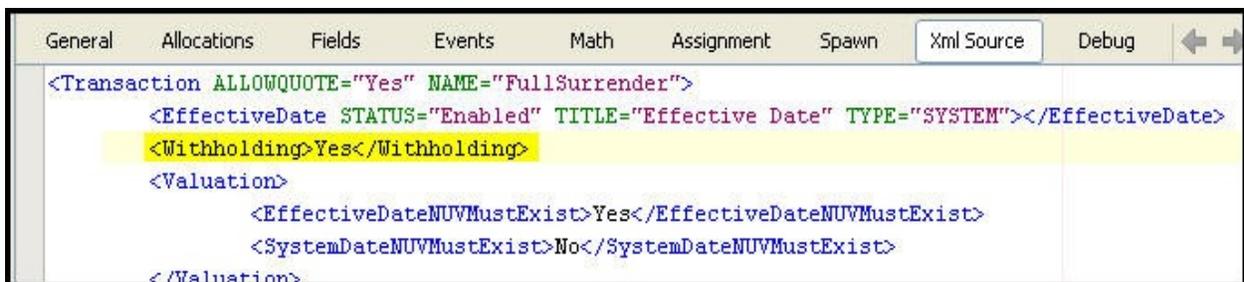
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Activity Level Withholding

Withholding can be specified at the activity level when configuration exists within the transaction. A <Withholding> element with a YES value the transaction XML determines whether activity level withholding applies, making a Withholding window available on the activity.

MathVariables are configured in the Math section of the transaction to define the Federal and State withholding. These can be configured using visual editing tools. The values can be passed to a disbursement transaction. Also the disbursement element has a sub-element called <WithholdingFields> that defines the federal or state withholding amounts and writes to AsDisbursementRecord.

Use the Main Explorer tab to open a **Company | Product | Plan | Transaction** folder. The Main Explorer is the only tab that allows a user to create new transactions or update existing transactions. Select the transaction that requires activity withholding and double-click the XML file. Then click the XML Source tab to determine if the Withholding element is present in the configuration. The tags can contain either **Yes** or **No**.



```
General Allocations Fields Events Math Assignment Spawn Xml Source Debug
<Transaction ALLOWQUOTE="Yes" NAME="FullSurrender">
  <EffectiveDate STATUS="Enabled" TITLE="Effective Date" TYPE="SYSTEM"></EffectiveDate>
  <Withholding>Yes</Withholding>
  <Valuation>
    <EffectiveDateNUVMustExist>Yes</EffectiveDateNUVMustExist>
    <SystemDateNUVMustExist>No</SystemDateNUVMustExist>
  </Valuation>
```

Withholding Element in Transaction

WithholdingScreen Business Rule

If Policy level withholding has been established and transaction withholding configuration is present, the Withholding Type combo field on the activity's Withholding window will present values of Policy and Activity. The Policy view presents disabled withholding fields with the policy level data populated while the Activity selection presents these same fields populated with the policy level data but are enabled and available for update.

If Policy level withholding established has not been established and Plan level withholding exists, the Withholding Type combo field on the activity's Withholding window will present values of Plan and Activity. The Plan selection will present the withholding fields as view only populated with any plan level data. The Activity selection will present these same fields with any Plan level data populated however the fields are enabled and available for update.

If neither Plan level nor Policy level withholding data exists, Activity is the only Withholding Type combo field value available. The Activity view presents enabled withholding fields with Null values.

Withholding Values

WithholdingField values are stored in the AsWithholdingFields table. Refer to the [Withholding Database section](#) for additional information.

View Withholding in OIPA

When an activity is configured to support activity level withholding, the Add Activity window will contain a Withholding link.

FullSurrender (01/29/2031) - AVA31ALIC010812

Details | **Withholding** ← **Click Withholding link**

WITHHOLDING

Withholding Type: **Activity** (Specify activity level withholding)

Federal Percent: 6% Federal Amount: 20.00 USD

State Percent: 7% State Amount: 30.00 USD

Opt-Out: No Yes State Form Returned: No Yes

Ok Cancel

Withholding Details in Activity Details Window

When a processed activity is configured to support activity level withholding, the withholding information can be reviewed when the Activity Detail icon is clicked on the Activity screen.

ACTIVITY

Display Shadows Reversals Future

From Date: To Date: 01/29/2031

Refresh

Activity	Activity Date	Status
FullSurrender	01/29/2031	Pending
AnniversaryProcessing	01/30/2006	Pending
EndOfQuarterProcess	04/29/2005	Pending

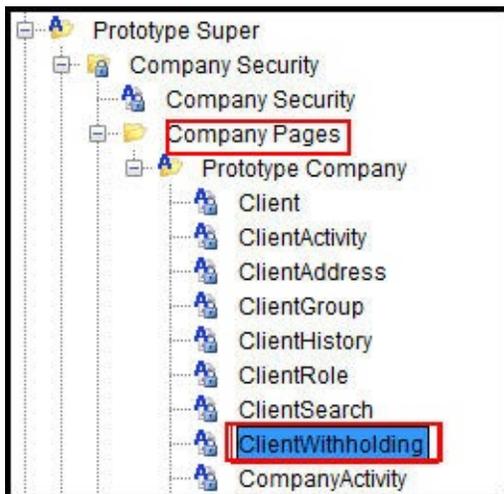
Activity Detail Icon on Activity Screen in OIPA

You are here: Configuration > Withholding > Client Level Withholding

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Client Level Withholding

In order for client level withholding information to be available to a user, the desired withholding fields must first be configured in the WithholdingScreen business rule. Security must also be applied to the Company Security page named [ClientWithholding](#). When security is added, the Withholding link will be available in the Left Navigation menu of the Client screen.



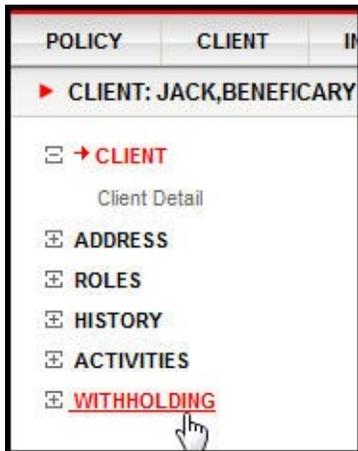
ClientWithholding in Company Pages

WithholdingScreen Business Rule

The WithholdingScreen business rule has a dual rule context: PrimaryComapny and Plan. These contexts define the available overrides allowed for the rule. Rule overrides allow for variation in configuration of the same rule based context. Refer to the [CopyBook Override Context](#) section for a description of available contexts.

WithholdingValues

WithholdingField values are stored in the AsWithholdingFields table. Refer to the [Withholding Database section](#) for additional information.



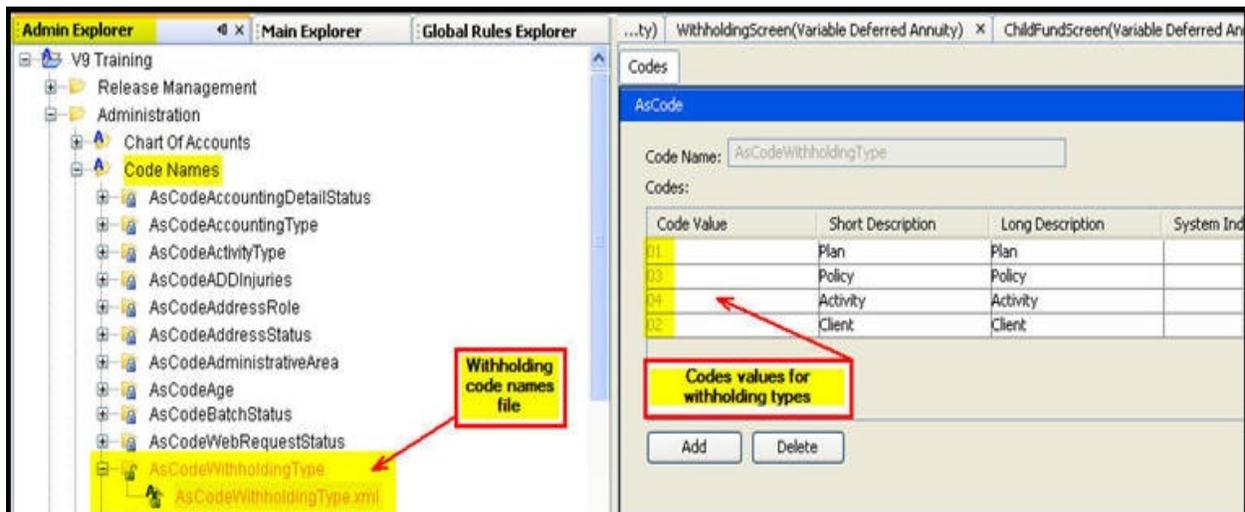
Client Level Withholding in OIPA

You are here: Configuration > Withholding > Withholding Codes and Database Structure

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Withholding Codes and Database Structure

Withholding codes can be added and modified through the Rules Palette Admin Explorer tab. Open the Rules Palette Admin Explorer tab and double-click on the Code Names folder. Scroll down to the AsCodeWithholdingTypes folder. Open the folder and check-out the XML file. Add and modify code values from this file.



Withholding Code Names File in Admin Explorer Tab

Database Structure

The following is an explanation of the tables that make up the withholding database structure.

AsWithholding Table

WithholdingGUID: Unique GUID that identifies a withholding record that was written.

TypeCode: Code for the level and type of withholding record. Codes from AsCodeWithholdingType.

RelatedGUID: This GUID links to either AsPlan, AsClient, AsPolicy or AsActivity. Relevant type codes are Plan, Client, Policy and Activity corresponding to the value of the related GUID .

AsWithholdingField Table

WithholdingGUID: The GUID from AsWithholding is used to associate dynamic field values with the correct Withholding record.

FieldName: Name of the dynamic field.

FieldTypeCode: Type code of the dynamic field from AsCodeField.

DateValue: If field is a date value it will be stored here, if not null will be stored.

TextValue: If field is a text value it will be stored here, if not null will be stored.

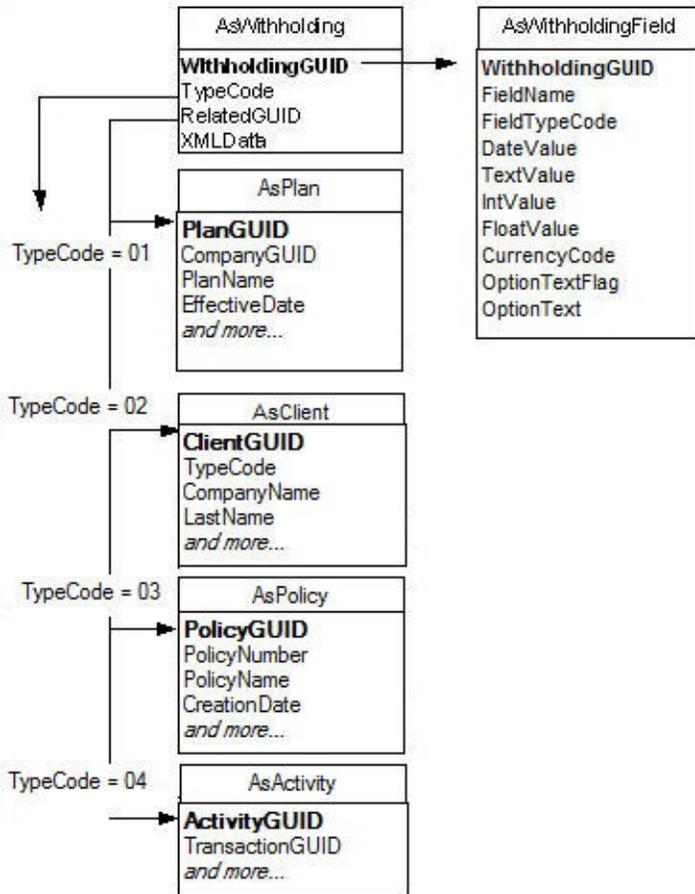
IntValue: If field is an integer value it will be stored here, if not null will be stored.

FloatValue: If field is a float value it will be stored here, if not null will be stored.

CurrencyCode: The currency code to assign money.

OptionTextFlag:

OptionText:



Withholding Database Tables

You are here: Configuration > Allocations > Allocation Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Allocation Overview

After funds are created, the next step is to structure how funds are selected in OIPA. The term **allocation structure** is used to describe the form, fields and tables that are used for fund selection in OIPA. Both the allocation structures and the logic that defines what the user is displayed for selection must be configured. When users select allocations, they map the movement of money in and out of funds for a policy.

OIPA functionality allows allocations to be configured that display either specific funds, fund classes or fund models.

Allocation Levels in OIPA

Allocations can be defined at the plan, policy and [transaction level](#). An explanation of each type is given below.

- **Plan-level allocations:** allocations that are specific to a plan, not a client or policy. For example, during the Freelook period all deposits are invested in Fund A until the end of the Freelook period is reached. After the Freelook period the fund allocations specified by the client will take place. Configure the [AllocationScreen business rule](#) and the [Plan Data node](#) associated with the plan in the Rules Palette.
- **Policy-level allocation:** these are typically selections for future allocations or systematic withdrawal. Configure the [PolicyAllocationScreen](#) business rule for allocations at this level.
- **Transaction level-allocations:** after the policy goes into effect, allocation links can be disabled and allocations are added via an activity. The [TransactionAllocationScreen](#) business rule should be used for allocations at this level.

Allocation Configuration Methods: [Default](#) or [Model Supported](#)

Allocations structures can list specific funds for users to select or they can use allocation models so users have to select from a predetermined diversifying strategy of funds. There are two different methods to configure allocations depending whether you are going to upgrade from V7 or V8 and your allocations only select funds; or if you configure allocations that use allocation fund classes or models. You must distinguish in the New Plan setup (or Plan Maintenance after the plan has initially been setup) which method you will use.

When setting up a plan in the New Plan Wizard, there are two options for plan allocations: Default or Model Supported. Default is for configuring allocations that will just use funds for selections. The Default selection is intended for clients that want to upgrade from V7 or V8 to V9 because all the same rules and syntax are used and this is backwards compatible. For clients new to V9 or previous V7 or V8 clients that use fund classes or models, selecting Model Supported is the optimal choice, as new rules were introduced that separate allocation structure by level, which reduces the confusion of having all allocations configured in one rule.

Configuring Allocations Rules

There are various rules that can be used to configure allocations. The following list of rules shows the pre-built functionality that can be used. The rules are divided according to what can be used for the Default method, the Model Supported method and also rules that can be used in both methods.

Rules and Elements For Configuring the Default Method

- **AllocationScreen:** This business rule is used to configure the funds allocation structure for a product at the plan, policy and segment level. This rule should be used when configuring allocations that use the Default (aka upgrade method) and not the Model Supported method (aka new V9 clients).
- **<Allocation> element:** This element in a transaction rule allows the configuration of fund allocation structures in an activity. These elements can only be used when Default Model method is selected for a Plan.
- **FundAllocation:** This business rule allows the display of allocations for an activity on the Activity Details screen.

Rules and Elements For Configuring the Model Supported Method

- **[PolicyAllocationScreen](#):** This business rule is used to configure policy-level allocation structures that will use allocation models. This rule can only be used when the Model Supported method is selected for a plan.
- **PlanAllocationScreen:** This business rule is used to configure plan-level allocation structures that will use allocation models. This rule can only be used when the Model Supported method is selected for a plan.
- **[TransactionAllocationScreen](#):** This attached business rule is used to configure activity-level allocations. This rule can only be used when the Model Supported method is selected for a plan.

Rules and Elements For Configuring both Default and Model Supported Methods

- **FundListForAllocation:** This attached business rule controls the fund drop-down box on the Activity screen.
- **WriteDefaultAllocations:** This attached business rule determines whether a transaction uses the default plan or policy level allocations.
- **ReassignAllocations:** Use this attached business rule to configure all allocations to move to an assigned fund. For instance, with new policies all allocations might be moved into a low risk account/fund until a waiting period has expired ensuring money.

Database Tables for Allocations

Refer to the [Allocation Database Table](#) section.

Code Names Needed For Allocations

The following code names and values must be created to support Allocations.

- AsCodeAllocationMethod: the method in which the allocation will be made, such as units, amounts or percent.
- AsCodeAllocationType: this is used when configuring the Allocation screen. This code identifies the type of allocation being configured.
- AsCodePlanAllocationMethod: this is identified in the Plan setup and it tells the system whether to use funds for allocations or allocation models.

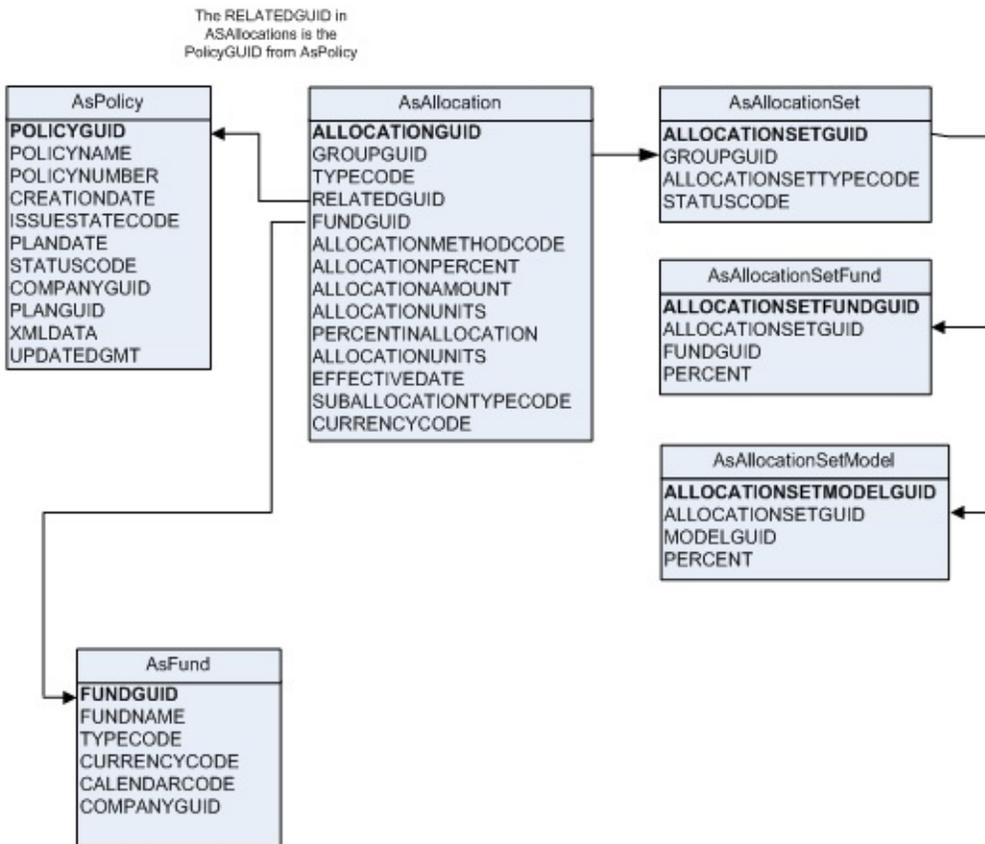
You are here: Configuration > Allocations > Allocation Database Tables

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Allocation Database Tables

When a user assigns allocations in OIPA, the AsAllocations table stores the fund allocation data attached to a policy, a plan or an activity.

Configuration can be tested by creating a policy and attaching allocations. Verify the correct records were written to the database. Allocations are linked to a policy via the AsAllocations table, using the RELATEDGUID and the POLICYGUID from the AsPolicy table. The GROUPGUID is used to combine allocations that were made at the same time, where each allocation has its own ALLOCATIONGUID.



Allocation Database Tables

You are here: Configuration > Allocations > Configure Allocations > Configure Allocations Using Models

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Configure Allocations Using Models

Allocation Models are made up of allocation structures that allow users to select funds according to a pre-determined strategy. Models are templates of funds or other portfolios that are grouped together with a certain investment goal.

An explanation of the types of models supported in OIPA is provided in the [Allocation Model](#) section of this help. An [XML example](#) of the OIPA PolicyAllocationScreen business rule configuration is also provided.

A model in OIPA is a template of selected fund allocations that can be actual funds or the ability to select funds that are in predetermined asset classes. To configure allocation models in OIPA, a plan must support the use of models. If asset class models are used, then funds must also be associated with [fund asset classes](#). After these two preliminary steps are complete, the allocation model must be configured. Then it can be associated to an allocation structure. Multiple models can be created for a plan, as they can be used in different scenarios, such as the creation of an aggressive or conservative model. The final step is to associate the models with the plans that will use them.

When using fund asset classes and/or models, all previous versions of OIPA must follow these steps. Upgrades from V7 and V8 will have to go through the process of setting up funds, fund asset classes and models as listed below. If upgrading to V9 and the products only used straight funds, then refer to the section called [Configure Allocations Using Default Method](#) .

High Level Steps to Configure Allocation Models

1. In the [New Plan Wizard](#) or [Plan Maintenance](#) screen, select **Model Supported** for the plan allocation method.
2. Set up the funds that will be used in the allocation.
3. Create a new allocation model.
4. Configure the allocation model.
5. Associate the allocation model with a plan.
6. Add the allocation model in the appropriate business rule (PolicyAllocationScreen or TransactionAllocationScreen) according to where the allocation model will be used in OIPA.
7. Test OIPA to ensure all the configuration works.

Steps to Configure Allocation Models

1. [Create a new plan](#) using the Plan Wizard and select **Model Supported** as the plan allocation method. If a plan already exists, then right-click on the plan in the Main Explorer and select **Edit Plan**. Select Model Supported for the plan allocation method.
2. Set up the funds that will be used in the allocation. The model will have predetermined funds already selected for the policy owner unless an owner is able to select the funds according to predetermined fund classes. In either case, proper set up is important.
 - Fund Allocation Model: if setting up an allocation model that uses determined funds, refer to the [Create Funds](#) section for further information.
 - Fund Asset Class Allocation Model: if setting up allocations that use fund asset classes, refer to the [Fund Asset Classes](#) section for information on how to create associations between funds and asset classes.
3. Create a new Allocation Model.
 - Navigate to the **Admin Explorer** and open **Administration |**

Allocation Models. The Allocation Models node is pre-populated with all existing Primary Companies.

- Right-click on the Allocation Models node and select **Add New Allocation Model**.
- In the wizard, enter the following information:
 - **Company:** Select the company that will use this model.
 - **Model Name:** Enter an identifiable name for the model.
 - **Model Type:** Select **Non-Asset Class Model** if using specific funds for the allocation model or **Asset Class Model** if using Fund Asset Classes. Selecting Asset Class Model allows a user to select specific funds in each asset class to be used to set up models for the allocations. These type values come from AsCodeModelType and are system codes.
 - **Model Category:** Select the category. Model category values come from AsCodeModelCategory of AsCode table.
 - **Model Description:** Enter a description. This is optional and can include additional information about the model.
 - **Effective Date:** The default is the present day's date but a different date can be used.
 - **Expiration Date:** This is blank and may be left blank if there is no expiration date for the model. A future date can be entered.
 - **Fund Percent Lock:** If this checkbox is checked, then the total percentage of all funds must equal 100% and the Model cannot be edited in OIPA. If this checkbox is unchecked, then total percentage of all funds must be equal to, or less than 100%, and the Model may be edited in OIPA. Percent Lock information is stored in database AsModelDefinition table's PercentLock column.
 - Select **Finish**. The new model sub-node will be added to the Primary Company node. The newly added model node will also have a contextual menu sub-node with the same name as the newly-created model. The newly-created

contextual sub-node's look and feel will emulate that of a newly created business rule (or transaction) before it is checked-in: it will be blue, and its content will be editable in the Model Grid pane on the right.

- Check-In and Check-Out the model that was created in the Admin Explorer window to commit it to the database.

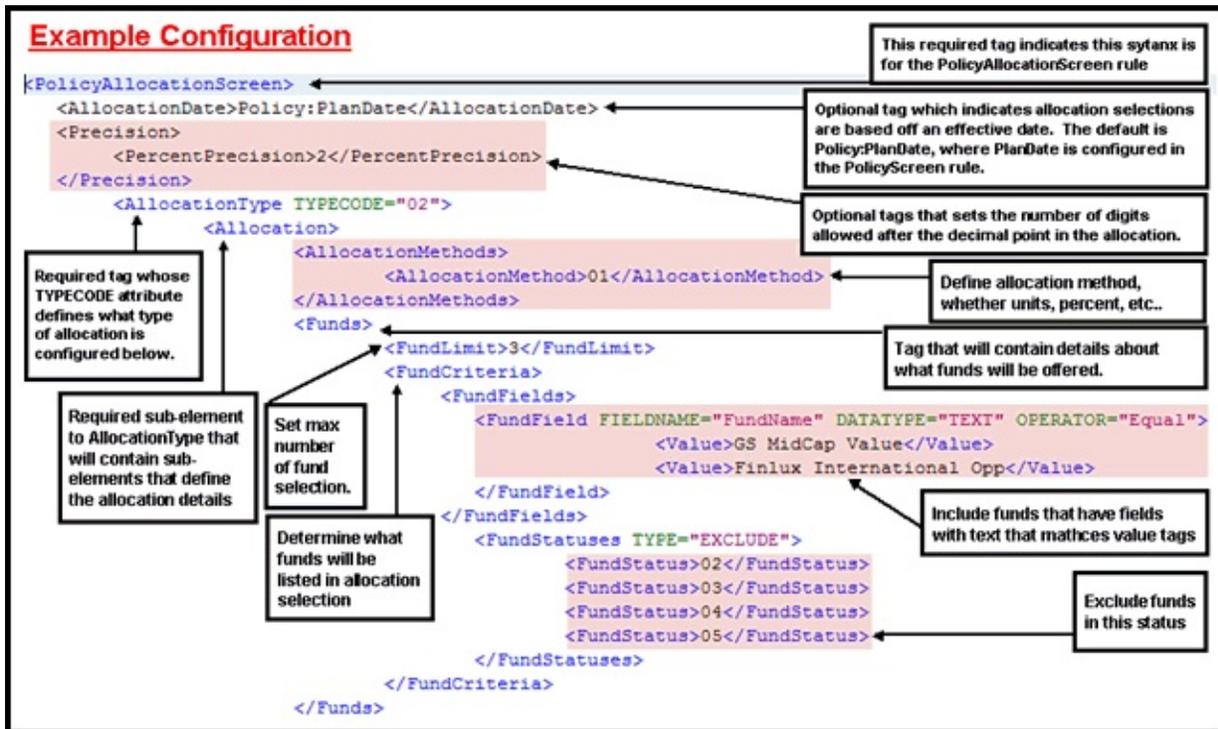
4. Configure the Allocation Model Details.

- Add a Variant under Model Variants if necessary. A variant is a version of the base model. Each model may have any number of versions. Model variants (versions) can be added, deleted and expired based on Effective / Expiration dates. Only one Model Variant can be active at a time. When a Model is first created, there will always be at least one Variant record in this section. The record will contain Variant Effective Date field (with date from the “New Allocation Model” wizard) and Variant Expiration Date.
- Select the variant the model will use. The dates will be highlighted blue for the variant that is selected.
- Create the model.
 - If using the Asset Class Method:
 1. Select the **Add Asset Class** button for each asset class in the model.
 - a. Select the Asset Class name. The Fund field will remain disabled until an Asset Class is selected.
 - b. Enter a Percent for each Asset Class. The total percentage of all Asset Classes must add to 100%.
 - c. Select **Add Fund** for each fund that will be associated with the Asset Class. The Asset Class field filters this field so that only funds associated with that Asset Class will be available.
 - d. For each record, select the Asset Class and fund and then enter a percent. All percentages must equal the percent selected for the Asset Classes.
 2. Select the **Add Fund** button to identify specific funds that will be available.

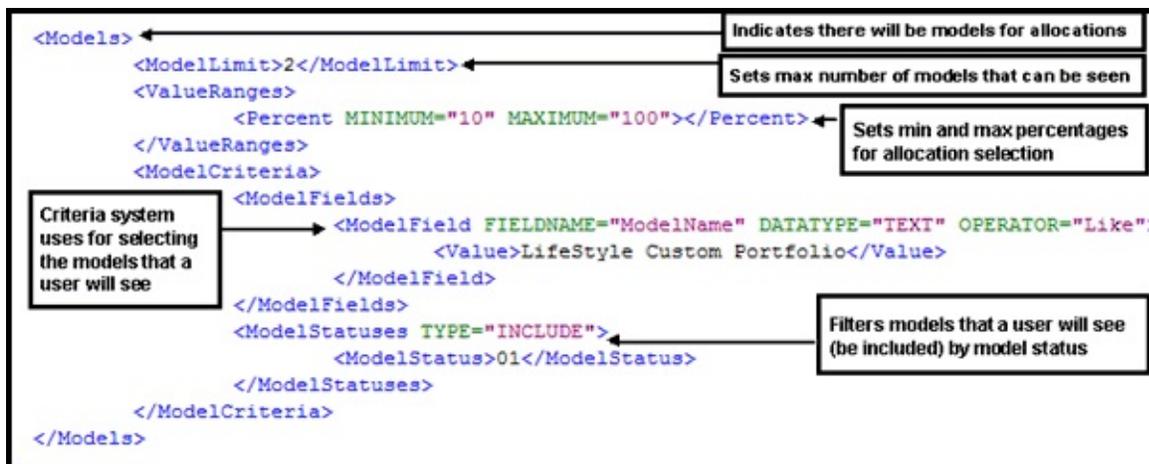
- a. Select the Asset Class.
 - b. Select the Fund Name.
 - c. Enter a Percent.
 3. Check in the file when complete.
 - If using a **Non Asset Class Model**:
 1. Select the **Add Fund** button for each fund in the model.
 2. Select each fund from the drop-down box that should be included in the model.
 3. Enter the fund percent for each fund. Make sure the aggregate is 100%.
 4. Check in the file when complete.
5. Associate the allocation model with a plan.
 - Navigate to the **Admin Explorer** and open **Administration | Allocation Models | Subsidiary Companies | Company | Plan**.
 - Right-click on the plan node and select **Add Plan Model Link**.
 - The Plan/Model Link Wizard opens. Enter the following information:
 - **Allocation Model**: Select the model to link to the plan.
 - **Effective Date**: Enter a date and select any variants. Make sure the dates coincide.
 - **Expiration Date**: This is optional.
 - **Status**: Select the status of the model.
 - Select **Finish**.
 - Check in the new Plan Link from the Admin Explorer window.
 - Add another plan model status if necessary from this pane.
6. Add the Allocation Model in the appropriate business rule (PolicyAllocationScreen or TransactionAllocationScreen) according to where the model will be used in OIPA. These business rules store

the allocation instructions for the varying allocations that may be presented at the different levels in OIPA.

7. Check allocations in OIPA to ensure they have been configured properly. Navigate to the appropriate screen depending on the configuration:
 - **Policy Allocations:** Locate or create a policy, then use the Left Navigation **Allocations** link.
 - **Transaction Allocations:** Locate the activity on the Activity screen that is associated with the transaction that was configured.
 - **Plan Allocations:** Please see the [AllocationScreen business rule](#) and the [Plan Data](#) page.



Basic Allocation Configuration and Fund Selection



Configuration Showing Allocations in Models

You are here: Configuration > Allocations > Configure Allocations > Allocation Models

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Allocation Models

Allocation models can be created to present possible asset allocation strategies for clients. Diversifying investments by asset categories can offset the fluctuations of the performance of the total portfolio.

Besides offering models, OIPA supports the use of funds through allocation configuration. Both options can be configured for use in OIPA at the same time. The presence of the configuration for both funds and models will indicate to OIPA that both should be offered.

Configuring Models

Refer to the section [Configure Allocations Using Models](#) for step by step instructions on configuring models. An [XML example](#) is also provided to demonstrate the elements needed to configure models.

Types of Allocation Models

OIPA supports two types of allocation models: [Asset Class](#) and [Non-Asset Class](#). Both of these models can be set up to be edited or can be locked down.

- **Non-Editable Non Asset Class Models** are locked down models that do not have asset class information available. The characteristics of this model are as follows:
 - funds are explicitly defined.
 - individual allocation percentages for each fund is fixed
 - asset class is not available.
- **Non-Editable Asset Class Models** are the most common asset allocation model used. These models are locked down models that will not be editable at any hierarchical level. Insurance companies or their advisors pre-determine the asset classes and their allocation percentages as well as the funds within each asset class and the corresponding percentages based on target risk profile and fund performances. The characteristics of this model are as follows:
 - Funds are explicitly defined for each asset class
 - individual allocation percentages of each fund are fixed
 - asset class percentages are fixed.
- **Editable Non-Asset Class Models** are flexible models with a pre-selected array of funds. During allocation, the user selects the funds within the model and the amount allocated to each fund. The characteristics of this model is as follows:
 - funds in the model are explicitly defined
 - individual allocation percentages of each fund can be changed.
 - allocation percentage can be set-up at a default value.
- **Editable Asset Class Models** have editable allocation of funds within asset class. The characteristics of this model are as follows:
 - total funds in the model are explicitly defined for each asset class
 - asset class allocation percent is fixed
 - within each asset class, individual allocation percentages of funds are

editable

- the allocation percentage could be set to a default value.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Allocations > Configure Allocations > Allocation Model XML Example

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Allocation Model XML Example

An explanation of the XML configuration needed to offer models in allocations is shown below.

- `<Models>` Begins the offering and filter of models in the allocation structure.
- `<ModelLimit>[Integer]</ModelLimit>` Limits the number of models that a user will be able to select.
- `<ValueRanges><Percent MINIMUM="[Constant]" MAXIMUM="[Constant]" />` Use this to set Min/Max limits on how much a user can allocate.
- `</ValueRanges>` Closes the ValueRanges section.
- `<ModelCriteria>` This starts a filtering configuration for models to limit or define what model a user sees.
- `<ModelFields>` In this section, configure only fixed model fields as criteria for the models an OIPA user will have for allocation selection.
- `<ModelField FieldName="[Model Name]" DATATYPE="[TEXT(DEFAULT)|..]" OPERATOR="[Equal|NotEqual|Like]">` Name the field from AsModelDefinition and use an operator to filter the funds the user will see.
- `<Value>[Literal or Policy:PolicyField]</Value>` List the value of the model field listed above that will be used for fund selection.
- `<Value>[Literal or Policy:PolicyField]</Value>` This demonstrates repeatability of this element.
- `</ModelField>` This closes the Model Field section.
- `<ModelField FieldName="[Model Category]" DATATYPE="[TEXT|INTEGER|..]" OPERATOR="[Equal|NotEqual|Like]">` This demonstrates repeatability of this element.
- `<Value>[Literal or Policy:PolicyField]</Value>` This demonstrates repeatability of this element.
- `</ModelField>`

- `</ModelFields>`
- `<ModelStatuses TYPE="[INCLUDE|EXCLUDE]">` The attribute allows inclusion or exclusion of the following models according to their status.
- `<ModelStatus>`[Model status as in AsCodeModelStatus]
`</ModelStatus>` The status code listed on the Plan Model Link Status pane for the allocation, but the code value is derived from AsCodeModelStatus.
- `<ModelStatus>`[Model status as in AsCodeModelStatus]
`</ModelStatus>`
- `</ModelStatuses>`
- `</ModelCriteria>`
- `</Models>`

You are here: Configuration > Allocations > Configure Allocations > Allocations in Transactions

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Allocations in Transactions

Configuring transactions that use allocations requires a unique syntax. . The allocation structure is still created, but there are additional tags for moving money in funds to other funds and setting default allocations. Also, transaction rules have an **Allocations** pane where the configuration is performed.

Steps to Configure Allocations in Transaction Rules

1. Check **Fund Allocation**, as this generates the <FundAllocation> tag, which is required when configuring allocations in a transaction rule. Money cannot move in or out of a policy without this tag, unless the ReassignAllocations business rule is used.
 - Click **Fund Allocation** to bring up the screen used to configure the funds that are available in the allocation structure. The display and format of allocation amounts and percents is determined here.
 - If upgrading from V8 or V7, then models or classes should be configured now. If creating a new product with V9, then refer to [Configuring Allocations that Use Allocation Models](#).
 - Use the XML Configuration Guide for a complete definition of what can be configured here. It is in the Help menu of the Rules Palette. You may also request through My Oracle Support the Fund and Allocation documentation for V8, as this configuration is primarily for upgrades.
2. Click **Allocation From** to configure this section if money will be moved out of a fund for transfers or rebalancing. This selection will inherit attribute values from the Fund Allocation if nothing is specified in the configuration.
 - Configure an assignment that is either TYPE="Transfer" or TYPE="REBALANCE". Please see the [Assignment](#) section.
 - ShowValuation restricts the system to only display funds in the policy that have value.
 - If the Fund Allocation has similar attributes configured, it will take precedence over what is configured for Allocation From.
3. Click **Allocation To** to configure this section if money will be moved in or out of a fund. This section identifies the funds that are available for allocations that move money into funds.
 - If this tag isn't configured then the attributes that were defined in the <FundAllocation> apply to the Allocation To selection in OIPA.
 - The FundAllocation selection initially sets up fund allocations that are moving money into funds. The Allocation selection has additional attributes that can be used to further define what funds are listed in the Allocation To selection area in OIPA. Since the same attributes could

exist in both places, only configure them in one tag, not both. The selections for FundAllocation will always take precedence.

- This tag is sometimes called the AllocationTo tag because it moves money into funds. There is no <AllocationTo> tag in this configuration, although there is an <AllocationTo> tag in the AllocationScreen business rule.
- The <AllocationFrom> tag controls the fund options for the Allocation From selection area.

4. Click **DefaultAllocation** to configure this section. Allocation selections will default from the allocation type that is already set up in the system.
 - FundAllocation must be selected to setup default allocations.
 - The Restriction selection controls the user's ability to modify the default allocations displayed in the Allocation section of the Activity Detail screen. If the value is Locked, then the allocations cannot be modified. If the value is Yes, then the allocations can be modified.
 - The TYPE selection can be used to define the default allocations that will be displayed as system types in OIPA.
 - The CODE selection can be used to define the default allocations that will be displayed via a TypeCode from AsCodeAllocationType.
5. Check in the transaction to save the allocation information.

You are here: Configuration > Allocations > Configure Allocations > Policy Allocation Screen

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Policy Allocation Screen_Rule

The following is a syntactical description of the allocation configuration for the PolicyAllocationScreen rule. It is important to understand the main elements of this rule. They are the same elements that are used in the TransactionAllocationScreen rule.

- `<PolicyAllocationScreen>` This is the main element.
- `<AllocationDate>[Policy:PlanDate]</AllocationDate>` This indicates that funds and models available for allocation should be based on the plan effective date specified in this element. Default for Allocation Date is `AsPolicy:PlanDate`.
- `<Precision>`: Precision at this level will define precision for all allocation types if not defined at the individual allocation type level.
- `<PercentPrecision>`: `[Constant]</PercentPrecision>`
- `</Precision>`
- `<AllocationType TYPECODE="[AllocationTypeCode]">`: AllocationType uses the type code to specify the allocation that the following configuration will apply to. Multiple allocations can exist at the policy level, but each is configured separately. The type codes are found in `AsCodeAllocationType`.
- `<Allocation SWITCH="To/From" ALLOWMIXEDMETHODS="Yes/No">`: The Allocation element is required and begins the actual configuration of the allocation, but the SWITCH attribute is optional and indicates that the allocation is a transfer. The ALLOWMIXEDMETHODS attribute is also optional, and defines whether mixed allocation methods are allowed.
- `<AllocationMethods>`: A container element for the individual `<AllocationMethod>` elements.
- `<AllocationMethod>[AsCodeAllocationMethod code value]</AllocationMethod>`: Specifies the allocation method(s) being used.
- `</AllocationMethods>`
- `<Funds>`: This starts the fund section, which is used by OIPA to determine

the funds available to a user.

- `<FundLimit>[Integer]</FundLimit>`: This limits the amount of funds a user can select in OIPA.
- `<ValueRanges>`
- `<Percent MINIMUM="[Constant]" MAXIMUM="[Constant]"/>`: Use this to set Min/Max limits on how much a user can allocate.
- `</ValueRanges>`
- `<FundCriteria>`: This starts a filtering configuration for funds, to limit or define what funds a user can see.
- `<Query>`: In this section, query can be initiated to filter initial list of funds that is existing in a policy
- `<FundFields>`: In this section, configure actual fixed or dynamic fund fields as criteria for the funds an OIPA user will have for allocation selection.
- `<FundField fieldName="[Fund Field Name]" DATATYPE="[TEXT (DEFAULT)|INTEGER..]" OPERATOR="[Equal|NotEqual|Like]">`: Name a fund field from AsFund or AsFundField and use an operator to filter the funds the user will be able to see.
- `<Value>[Literal or Policy:PolicyField]</Value> . . .`
`</FundField>`: This demonstrates repeatability of this element.
- `<FundField fieldName="[Fund Field Name]" DATATYPE="[TEXT|INTEGER..]" OPERATOR="[Equal|NotEqual|Like]">`: This demonstrates repeatability of this element.
- `<Value>[Literal or Policy:PolicyField]</Value>`: This demonstrates repeatability of this element.
- `</FundField>`
- `</FundFields>`
- `<FundStatuses TYPE="[INCLUDE|EXCLUDE]">`: Include or exclude funds by status.
- `<FundStatus>[Status as in AsCodeFundStatus]</FundStatus>`: For a fund that is associated to a plan, the status is set on the Plan Fund Status page in the Rules Palette and saved to AsPlanFundStatus. The code is derived from AsCodeFundStatus.

- `<FundStatus>`[Status as in AsCodeFundStatus]`</FundStatus>` :
This demonstrates repeatability of this element.
- `</FundStatuses>`
- `</FundCriteria>`
- `</Funds>`

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Allocations > Configure Allocations > Transaction Allocation Screen

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transaction Allocation Screen_Rule

The Transaction Allocation Screen populates the default policy level allocation details to the "Allocation" tab in activity process. For Funds allocation, the user has the option to add or modify the funds within the allocation sections and the fund list can be filtered by adding a query element. Support is provided to filter funds using and/or logic in business rules that compliments the syntax.

The following is a syntactical description of the allocation configuration for the TransactionAllocationScreen rule. Only the varying elements from the [PolicyAllocationScreen](#) rule are described below.

```
<TransactionAllocationScreen>
```

```
    <AllocationDate>[Policy:PlanDate]</Al  
    <Precision>
```

```
        <PercentPrecision/>
```

```
    </Precision>
```

```
    <Allocation SWITCH="FROM">
```

```
    <FundFields>
```

```
    <And>
```

```
    <FundField FIELDNAME="FundName" DATATYPE="TEXT"  
    OPERATOR="Equal">
```

```
    <Value>High Income</Value>
```

```
    <Value>Balanced</Value>
```

```
    </FundField>
```

```
<FundField FIELDNAME="FundCategory" DATATYPE="TEXT"
OPERATOR="Equal">
<Value>01</Value>
<Value>02</Value>
</FundField>
</And>
<Or>
<FundField FIELDNAME="FundMODEL" DATATYPE="TEXT"
OPERATOR="Equal">
<Value>Model01</Value>
<Value>Model02</Value>
</FundField>
<FundField FIELDNAME="FundValue" DATATYPE="TEXT"
OPERATOR="NotEqual">
<Value>100</Value>
<Value>200</Value>
</FundField>
</Or>
</FundFields>
<Events>
<Event TYPE="ONLOAD">
<ActionSet ID="OnLoadActionSet"></ActionSet>
</Event>
</Events>
<Actions>
<ActionSet ID="OnLoadActionSet">
<Condition IF="Activity:AutoTransfer='YES'">
<Action ACTIONTYPE="HIDE" SECTIONNAME="AllocationFrom">
</Action>
<Else>
<Action ACTIONTYPE="SHOW" SECTIONNAME="AllocationFrom">
</Action>
</Else>
</Condition>
</ActionSet>
</Actions>
```

<AllocationMethods>

```
        <AllocationMethod>[AsCodeAl
</AllocationMethods>

<FilterOnValuation VALUATIONLEVE
<DefaultAllocation LEVEL="[Plan|
</DefaultAllocation> Default All
<Funds>

    <FundLimit>[Integer]</FundL
    <ValueRanges>
        <Percent MINIMUM="[Con
    </ValueRanges>
    <FundCriteria>
        <FundFields>
            <FundField FieldN
                <Value>[Literal o
        </FundField>
```

```
        </FundFields>
        <FundStatuses TYPE="[INCLUD
            <FundStatus>[Model sta
        </FundStatuses>
        </FundCriteria>
    </Funds>
</Allocation>
<ScreenMath>
    <Math ID="ScreenMathName">
        ...
    </Math>
</ScreenMath>
<Events>
    <Event TYPE="ONLOAD|ONCHANGE|ONS
        ...
    </Event>
</Events>
<Actions>
    <ActionSet ID="ActionSetName">
        ...
    </ActionSet>
</Actions>
```

</TransactionAllocationScreen>

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Allocations > Configure Allocations > Allocation Transfers

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Allocation Transfers

The <Allocation SWITCH="FROM"> and <Allocation SWITCH="TO"> elements are needed in configuration to support allocations that transfer money FROM funds TO funds. The rest of the fund and model configuration is the same as that described in the [Allocation Model XML Example](#).

```
<AllocationType TYPECODE="09">
  <Allocation SWITCH="From">
    <AllocationMethods>
      <AllocationMethod>01</AllocationMethod>
    </AllocationMethods>
    <Funds>
      <FundLimit>3</FundLimit>
      <FundCriteria>
        <FundFields>
          <FundField FIELDNAME="FundName" DATATYPE:
            <Value>GS MidCap Value</Value>
            <Value>Finlux International Opp</Va.
          </FundField>
        </FundFields>
        <FundStatuses TYPE="EXCLUDE">
          <FundStatus>02</FundStatus>
          <FundStatus>03</FundStatus>
          <FundStatus>04</FundStatus>
          <FundStatus>05</FundStatus>
        </FundStatuses>
      </FundCriteria>
    </Funds>
  </Allocation>
  <Allocation SWITCH="To">
    <AllocationMethods>
      <AllocationMethod>01</AllocationMethod>
    </AllocationMethods>
    <Funds>
      <FundLimit>3</FundLimit>
      <FundCriteria>
        <FundFields>
```

```

        <FundField FIELDNAME="FundName" DATATYPE:
            <Value>GS MidCap Value</Value>
            <Value>Finlux International Opp</Va.
        </FundField>
    </FundFields>
        <FundStatuses TYPE="EXCLUDE">
            <FundStatus>02</FundStatus>
            <FundStatus>03</FundStatus>
            <FundStatus>04</FundStatus>
            <FundStatus>05</FundStatus>
        </FundStatuses>
    </FundCriteria>
</Funds>
<Models>
    <ModelLimit>2</ModelLimit>
    <ValueRanges>
        <Percent MINIMUM="10" MAXIMUM="100"/>
    </ValueRanges>
    <ModelCriteria>
        <ModelFields>
            <ModelField FIELDNAME="ModelName" D,
                <Value>LifeStyle Conservative I
            </ModelField>
        </ModelFields>
        <ModelStatuses TYPE="INCLUDE">
            <ModelStatus>01</ModelStatus>
            <ModelStatus>09</ModelStatus>
            <ModelStatus>08</ModelStatus>
        </ModelStatuses>
    </ModelCriteria>
</Models>
</Allocation>
</AllocationType>

```

You are here: Configuration > Allocations > Allocation Screens in OIPA

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

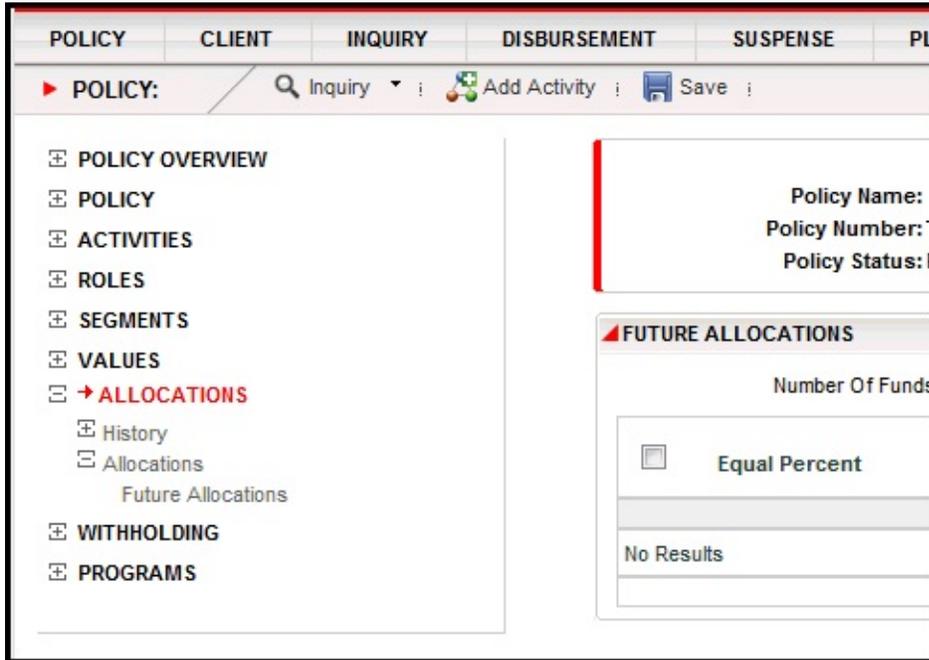
Allocation Screens in OIPA

There are several screens in OIPA that display the allocation configuration performed in the Rules Palette. It is a good idea to check the configuration in OIPA to make sure it is really configured as needed.

Below are some examples of the various Allocation screens.

Policy Level Allocations

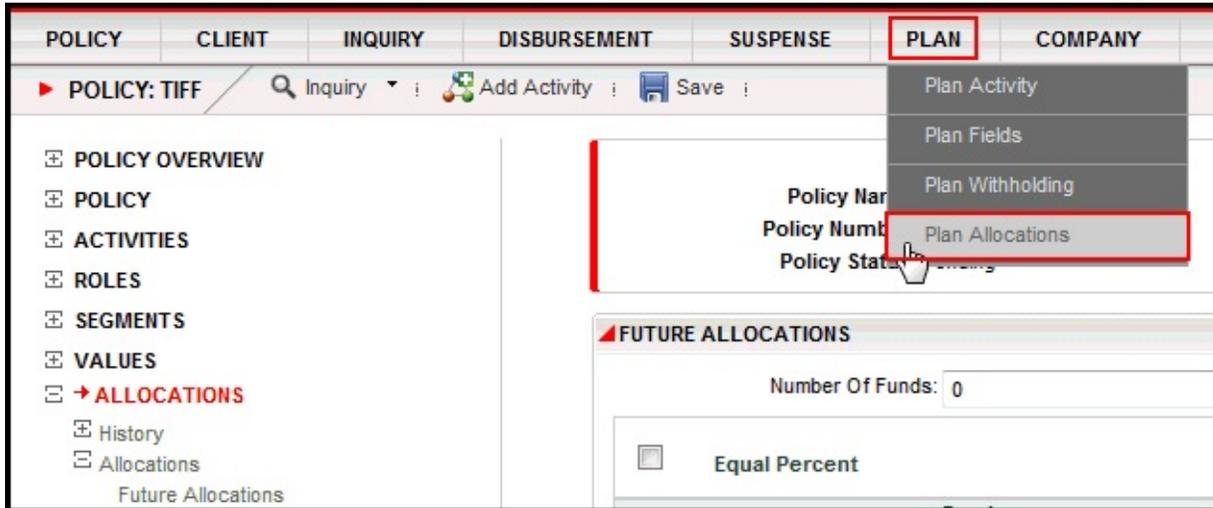
Notice on the Left Navigation menu of the Policy screen that there is an **ALLOCATIONS** selection. This is where the policy's fund allocations are selected.



Policy Screen Allocation Link

Plan Level Allocations

From the Main menu, the **Plan** option provides a link to the plan level allocations.



Plan Menu on Main Menu in OIPA

Activity (Transaction) Level Allocations

Notice in this example, an activity is going to run against a policy. The activity has fund allocations options.

The screenshot displays the Oracle Insurance Policy Administration interface. The main header shows 'ORACLE Oracle Insurance Policy Administration'. Below this, there are navigation tabs: POLICY, CLIENT, INQUIRY, DISBURSEMENT, SUSPENSE, PLAN, and TABLES. The current view is for 'POLICY: VDA31024243'. A left-hand navigation menu includes: POLICY, ACTIVITIES, ROLES, SEGMENTS, VALUES, ALLOCATIONS, and WITHHOLDING. The main content area shows 'POLICY' details: Policy Name, Policy Number: VDA31024243, Policy Status: Active, Plan Date: 1/29/2031, Company Name: Primary Company 1, and Plan Name: Variable Deferred Annuity. A modal window titled 'FUNDTRANSFER - VDA31024243' is open, showing 'Details' and 'Allocations' tabs. The 'ALLOCATIONS' section contains a table with the following data:

Fund Name	Cash Value
Equity Income (Band160)	7,070.00 USD
Balanced (Band125)	7,070.00 USD

Below the allocations table, there are sections for 'ACTIVITY' with 'FROM FUNDS' and 'TO FUNDS' options. Each section includes a 'Number Of Funds' dropdown (set to 0) and an 'Allocation Method' dropdown (set to Amount). The 'FROM FUNDS' section shows a table with columns 'Fund' and 'Amount', and a 'Total: 0.00 USD' at the bottom. The 'TO FUNDS' section also shows a table with columns 'Fund' and 'Amount', and a 'Total: 0.00 USD' at the bottom.

Activity Allocations in OIPA

You are here: Configuration > Valuation > Valuation Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Valuation Overview

Valuation is the calculation of a policy's worth. When OIPA performs valuation, it calculates the policy's current fund and deposit values. Policies that accumulate cash value, such as variable annuity or unit linked, are the only types of policies that are subject to valuation calculations. Term policies do not accumulate cash value, so there is no value to calculate.

The calculation of a policy's value is determined by the type of funds in the policy. A policy owner invests money in a policy by making deposits into specific fund(s). The value of a policy depends on the amount in the funds, the type of funds the policy owner has elected to purchase and the funds' performance in the marketplace. The policy owner may move money out of funds or transfer between funds, which will also affect the policy's value.

The movement of money in and out of a policy requires complex calculations and logic. The OIPA system offers an agile environment that can support the various configuration scenarios needed to manage a wide array of valuation requirements and fund types.

There are two different methods of configuring valuation: traditional and [Point-in-Time](#). When **traditional valuation** is used, all policy valuation calculations are made using all data from the inception of the fund. When Point-in-Time valuation is used, policy valuation calculations are made using values from the last processed valuation, rather than from inception. Since traditional valuation must recalculate all valuation data each time valuation is processed, it is much more resource intensive than Point-in-Time valuation.

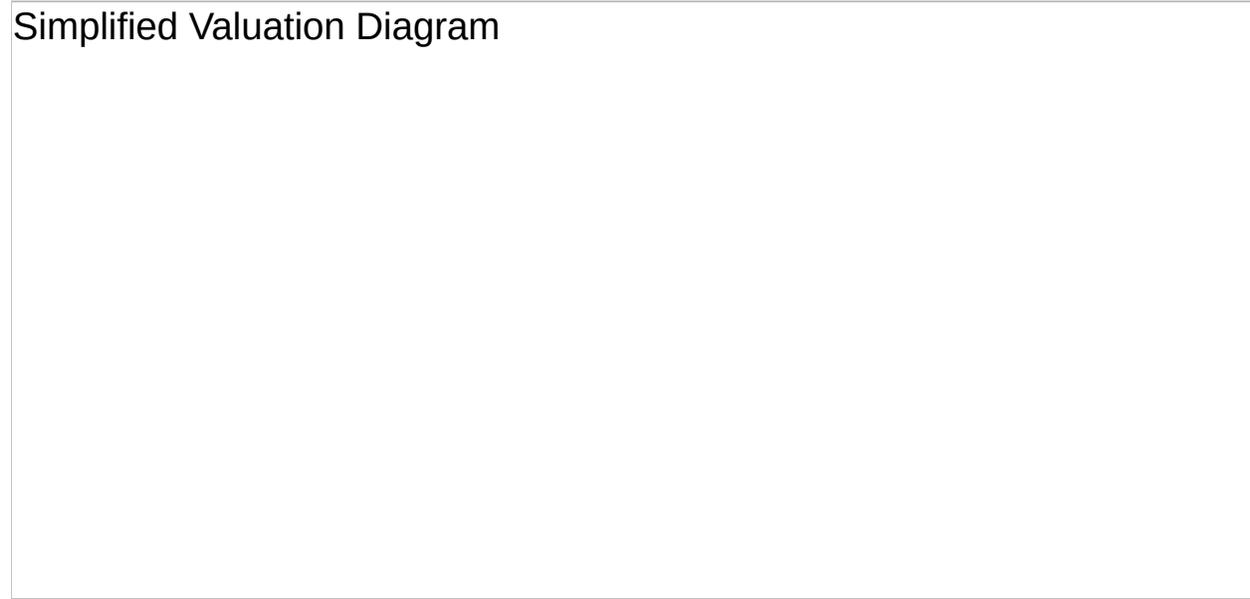


A plan that uses traditional valuation with fixed or variable funds can be converted to use Point-in-Time valuation. The only exception involves unit linked funds. They are not supported at this time.



The interest rates [retrieved](#) during fixed fund valuation must be maintained and accessible from AsRate and AsRateGroup tables. These cannot be delegated to external resources or customized implementation.

Simplified Valuation Diagram



Simplified Valuation Diagram

Valuation and Unit Linked Funds

Unit linked funds require a special type of valuation.

When configuring valuation the following areas should be considered:

- Funds
- Allocations
- Transactions
- Assignments
- Any additional rules to enhance processing

You are here: Configuration > Valuation > OIPA Valuation Mechanisms

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Understanding OIPA Valuation Mechanisms

Before configuring a plan that will value policies it is important to understand how OIPA processes values. The OIPA code will perform hard coded calculations depending on the type of funds being used for valuation. OIPA will also perform certain predefined steps when valuation is run. This is known as running the **valuation engine**.

A transaction may be configured to execute or not execute the valuation engine. Upon execution of the valuation engine, the OIPA system gathers the fund information needed to perform the valuation steps. The information gathered is dependent on the use of Point-In-Time valuation or Traditional valuation.

- Point-In-Time valuation: uses the fund and/or deposit information and values as of the last time valuation was executed by an activity on the policy. When an activity executes Point-in-Time valuation for a fixed fund, the rounded cash value is stored in the CashValue column on AsFundValue and AsDepositValue. Additionally, the unrounded cash value is stored in the RawCashValue columns on these tables as well. When valuation is subsequently executed for the fixed funds, the RawCashValue data is used as the beginning value for the fund.
- Traditional valuation: uses all of the policy's deposits and removals since inception of the policy. Traditional valuation gathers this information by calculating the policy's fund values as of each deposit and removal date until it reaches the last. The data accumulated to this point is kept in OIPA system memory and is known as the **Valuation Structure**.

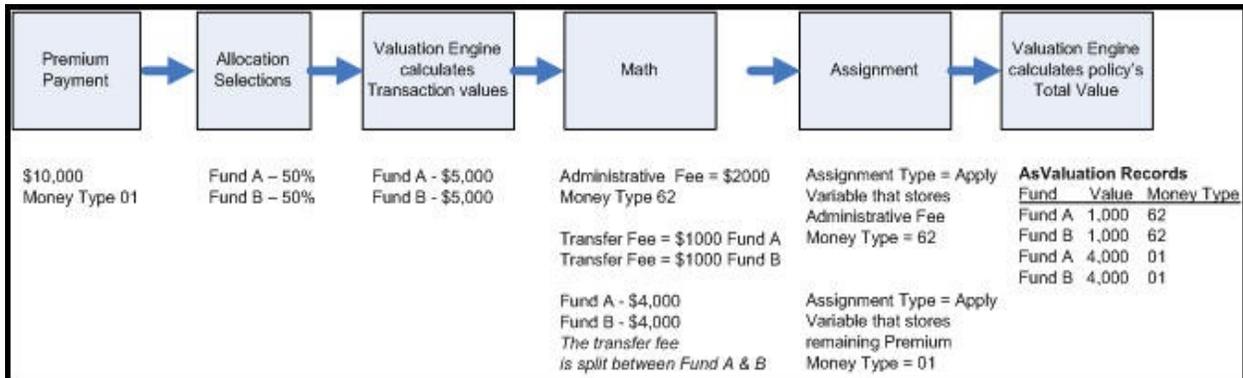
With both methods now at a similar point, fund and deposit values are calculated to the current valuation date. Results of the calculations are stored in the [Valuation Structure](#). Data in the Valuation Structure are available to the Math section. Each fund type follows a different calculation path in the OIPA system to achieve the policy's/fund's/deposit's values.



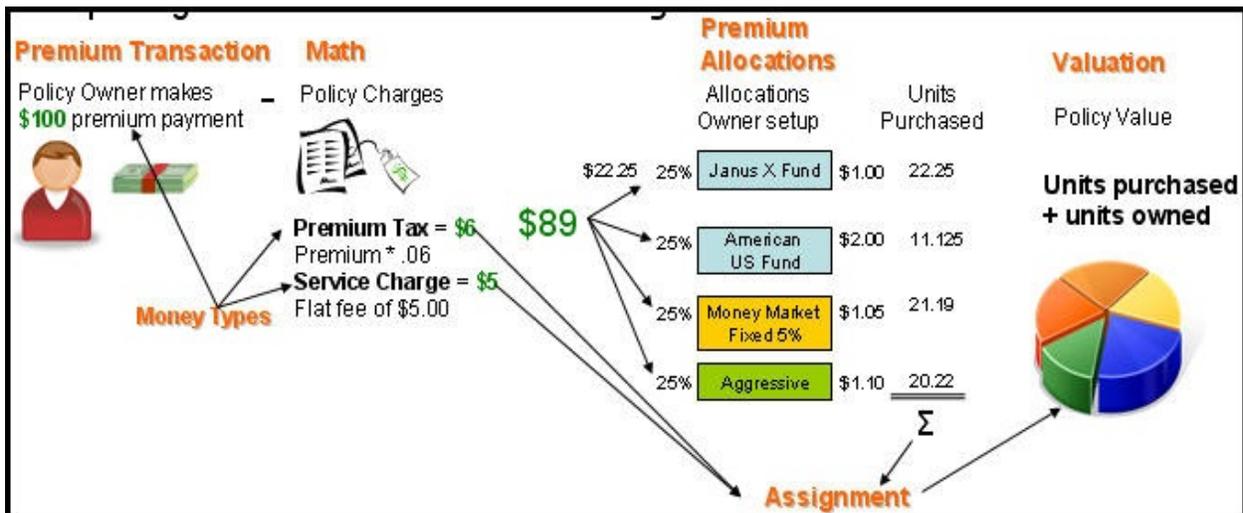
Refer to the **XML Configuration Guide** topic in this help system for information on using Valuation prefixes to access values within the valuation structure. Select **Common Elements | Available Prefixes and Fields for Configuration**.

Configuring Valuation

Valuation can be configured and run at the transaction level. The output can be viewed on the OIPA Activity Results screen. Valuation can also be configured at the policy level. The output can be viewed on the OIPA Values screen.



High Level Valuation Steps



High Level Work Flow Diagram

Valuation Structure

In traditional valuation, when the system calls the valuation engine, the engine creates a **valuation structure**. The valuation structure includes the policy, fund, deposit and removal information since the inception of the policy. The Valuation Structure is then stored in XML when the activity is completed. [View sample valuation structure](#).

Active Valuation

Initially all valuation records are in a pending state but as more values are known the status moves to an active state. This update is system driven and not configurable, much like normal disbursement and activity status transitions.

As the assignment is written, redemption records receive negative amounts and units while subscriptions are positive.

For unit linked funds, an attached activity could be in NUV Pending status and its valuation records would need to be excluded from the valuation process. This means that a unit linked valuation record is only active if:

- It has an effective date less than or equal to the valuation date input.
- The current system date falls between its active dates.
- When all the Redemption amounts and units for an activity are known and written, the Subscription amounts can also be derived and written at the same time so all Valuation records can also move to an active state. Otherwise the status is unchanged. The activity will remain in an NUV Pending state until the Redemption units are known.

When all three of the above criteria are met, the record is written to **AsActiveValuationActivity** and will be included when valuing the policy.

Valuation XML

The Valuation XML is made up of specified values that can be used in a [transaction's math section](#). Most of the valuation structure cannot be changed and is hard coded. The valuation structure and its content cannot be changed during the execution of the activity that created it. There is one business rule called [PolicyValues](#), which allows custom configuration of values and is added to the valuation structure. The structure with its associated values are located in the XMLData column of the AsValuationXML table.

Use these values in a transaction's math by prefixing a MathVariable with **Valuation:** and the node path separated by colons. This is discussed in more detail in the [Configuring Math section](#).

With Point-In-Time valuation, a valuation structure is also created. This structure consists of the last known policy/fund/deposit values. Its values are available in a transaction's math section using the same syntax as above. The structure is not saved in the AsValuationXML table.

[Example of XML Valuation Structure taken from AsValuationXML](#)

You are here: Configuration > Valuation > OIPA Transaction Valuation from the End Users Perspective

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

OIPA Transaction Valuation from the End User's Perspective

When valuation is configured at the transaction level using the <Valuation> element, valuation values can be viewed from the Activity Results screen in OIPA. The user may drill down to very granular information.

If valuation is configured to calculate multiple times and for different dates within a single activity then a math valuation object will remain in memory until it is replaced by another math valuation object or when the activity's math section has completed execution. Only one math valuation object will exist in memory at the same time as activity valuation. Refer to the [MathStatement](#) section for additional information on configuring this type of valuation.

It is important to understand that valuation details are not available for viewing until the fund's unit values are added to the system data. Prior activities where unit values are available will have the valuation details appropriate for the activity. Policies entirely invested in fixed funds may always execute regardless of unit values.



If valuation XML is generated, the valuation tab will display data. If the XML is not generated, no data will display in the Valuation tab. Whether or not XML is generated is determined in the PAS.properties file. Refer to the OIPA System Properties document in the [9.6.0.0 Documentation Library on OTN](#) for additional information on this property.

For a non-processed activity, an Activity Detail window displays when the icon to the left of the activity name is clicked. The window opens and allows users to add/modify the entry information for the activity.

For a processed activity, an Activity Results window displays when that icon to the left of the activity name is clicked. The Detail tab is the default

tab that will display. Click the Valuation tab to view valuation results for the activity.

ACTIVITY

Display Shadows Reversals Futures Documents Financial

From Date: To Date: Filters:

Auto-Process

1 2 Maximum Results 10

	Activity	Activity Date	Status	Detail	Action
	SystematicWithdrawalStop	01/01/2004	Pending		
	AnniversaryProcessing	01/03/2001	Pending		
	SystematicWithdrawal	07/15/2000	Pending		
	EndOfQuarterProcess	07/03/2000	Pending		
	EndOfMonthProcess	05/03/2000	Pending		
	Disbursement	04/15/2000	Pending		
	SystematicWithdrawal		Active		
	EndOfQuarterProcess		Active		
	EndOfMonthProcess		Active		
	EndOfMonthProcess	03/03/2000	Active		

Activity Details Display Valuation Information

ActivityDetail Icon on the Activity Screen in OIPA

Detail |
 Entry Fields |
 Allocations |
 Math |
 Spawn |
 Accounting |
 Valuation |
 Disbursement |
 Suspense

VALUATION DETAILS

Effective Date: 01/29/2031 Principal: 57,512.50 USD
 Active Date: 01/29/2031 Gain: 27,819.53 USD
 Cash Value: 85,332.03 USD Free Amount: 0.00 USD
 Policy Values:

DEPOSIT DETAILS

Date	Type	Cash Value	Surrender	Principal	Gain
------	------	------------	-----------	-----------	------

FUND VALUATION

	Name	Cash Value	Units	Unit Value	Principal	Gain
	GMB	355,567.27 USD	0.0000	0.00 USD	100,000.00 USD	255,567.27 USD
	Money Market (Band125)	0.00 USD	-0.0004	11.97 USD	2,933.27 USD	(2,933.27) USD
	Balanced (Band125)	85,332.03 USD	6,034.7969	14.14 USD	54,579.23 USD	30,752.80 USD

	Date	Type	Transaction	Cash Value	Units	Principal	Gain
	02/18/2005	Premium	AutoTransfer	85,332.03 USD	6,034.7969	54,579.23 USD	30,752.80 USD

	Date	Type	Transaction	Amount	Units	Interest Amount
	04/29/2005	GMB Fee	EndOfQuarterProcess	(412.50) USD	-42.9308	0.00 USD
	07/29/2005	GMB Fee	EndOfQuarterProcess	(412.50) USD	-42.9811	0.00 USD
	10/31/2005	GMB Fee	EndOfQuarterProcess	(412.50) USD	-42.6636	0.00 USD
	01/30/2006	GMB Fee	EndOfQuarterProcess	(412.50) USD	-41.2500	0.00 USD

Valuation Tab in ActivityDetails Window in OIPA

EndOfQuarterProcess (04/03/2000) - AVA31ALIC010978

Detail | Entry Fields | Allocations | Math | Spawn | Accounting | Valuation | Disbursement | Suspense

VALUATION DETAILS

Effective Date: 04/03/2000
 Active Date: 01/29/2031
 Cash Value: 157,330.14 USD
 Policy Values:

Principal: 148,075.49 USD
 Gain/Loss: 9,254.66 USD
 Free Amount: 0.00 USD

SEED DEPOSIT DETAILS

Date	Type	Cash Value	Surrender	Principal	Gain/Loss
------	------	------------	-----------	-----------	-----------

FUND VALUATION

Name	Cash Value	Units	Unit Value	Principal	Gain/Loss		
High Income (Band125)	40,517.29 USD	2,160.8277	18.75 USD	37,018.87 USD	3,498.42 USD		
	Date	Type	Transaction	Cash Value	Units	Principal	Gain/Loss
	02/02/2000	Premium	AutoTransfer	40,517.29 USD	2,160.8277	37,018.87 USD	3,498.42 USD
Balanced (Band125)	37,382.55 USD	4,914.8775	7.61 USD	37,018.87 USD	363.68 USD		
Mid Cap (Band125)	39,936.89 USD	3,852.2036	10.37 USD	37,018.87 USD	2,918.02 USD		
GMDB Rollup	151,810.48 USD	0.0000	0.00 USD	150,000.00 USD	1,810.48 USD		
Equity Income (Band125)	39,493.41 USD	2,976.5619	13.27 USD	37,018.87 USD	2,474.54 USD		

Valuation Details in ActivityDetails Window in OIPA

You are here: Configuration > Valuation > Valuation Database Diagrams

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

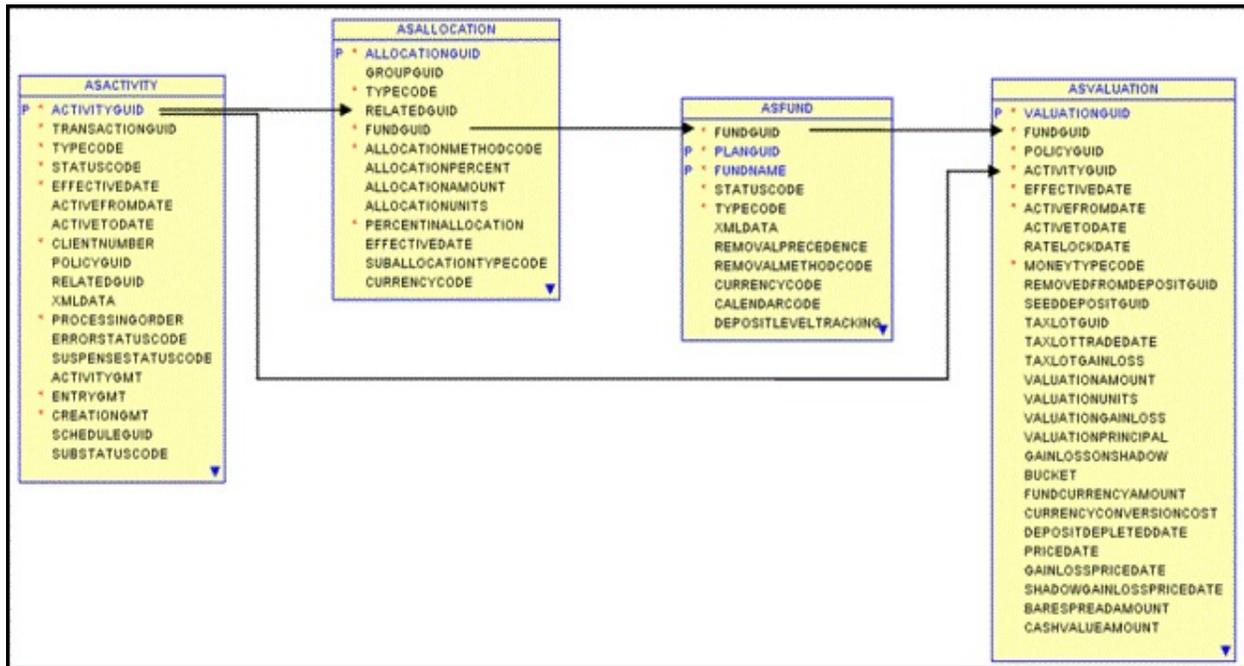
Valuation Database Diagrams

Before configuring valuation, review the valuation database diagrams. They may be used to test the valuation configuration being developed.

- [High Level Valuation](#)
- [Valuation](#)
- [Funds](#)
- [Allocations](#)

High Level Valuation

At a very high level, there is a relationship between activity, allocation, fund and the valuation record.

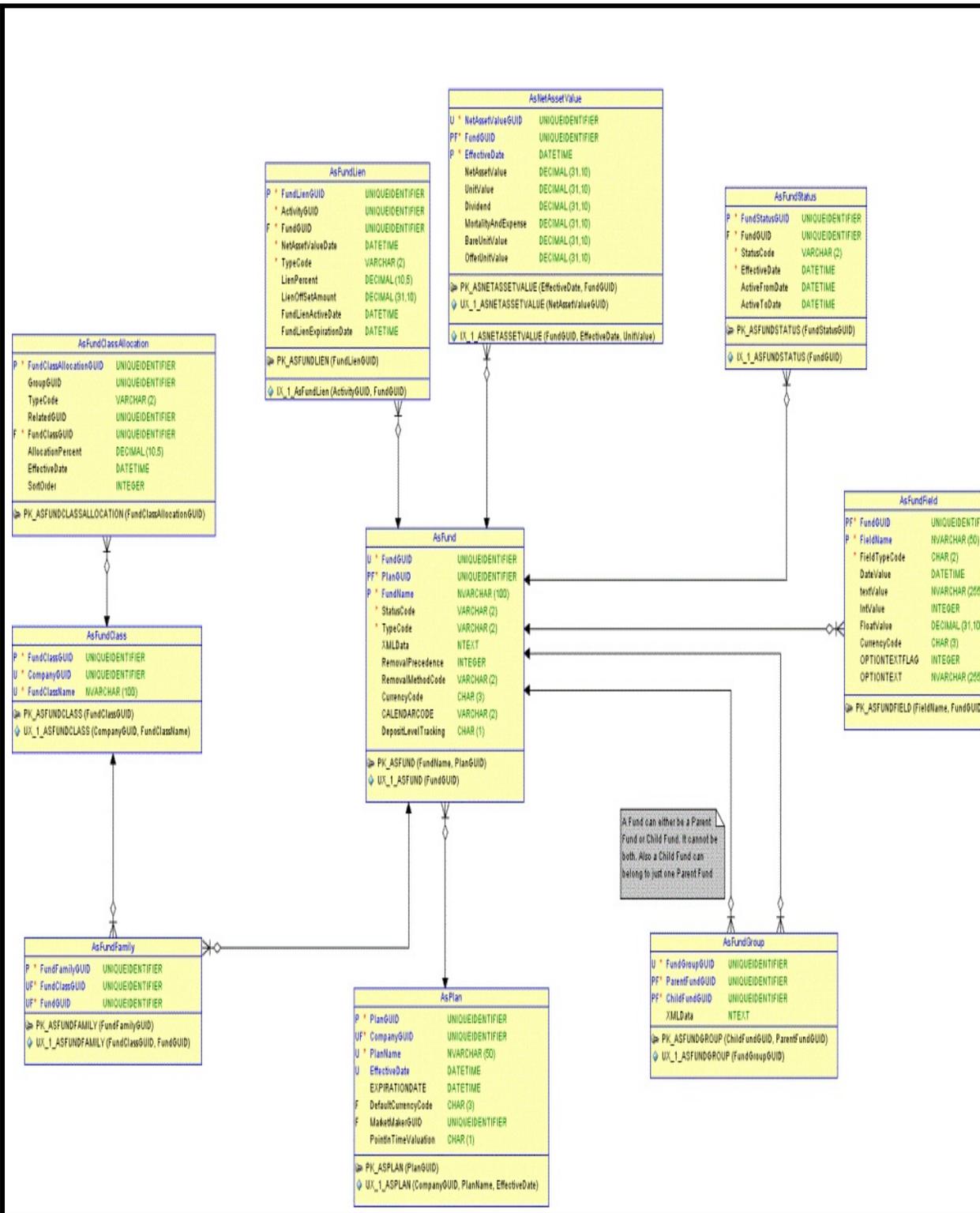


Tables Showing Relationship Between Activity, Allocation, Fund and Valuation Records

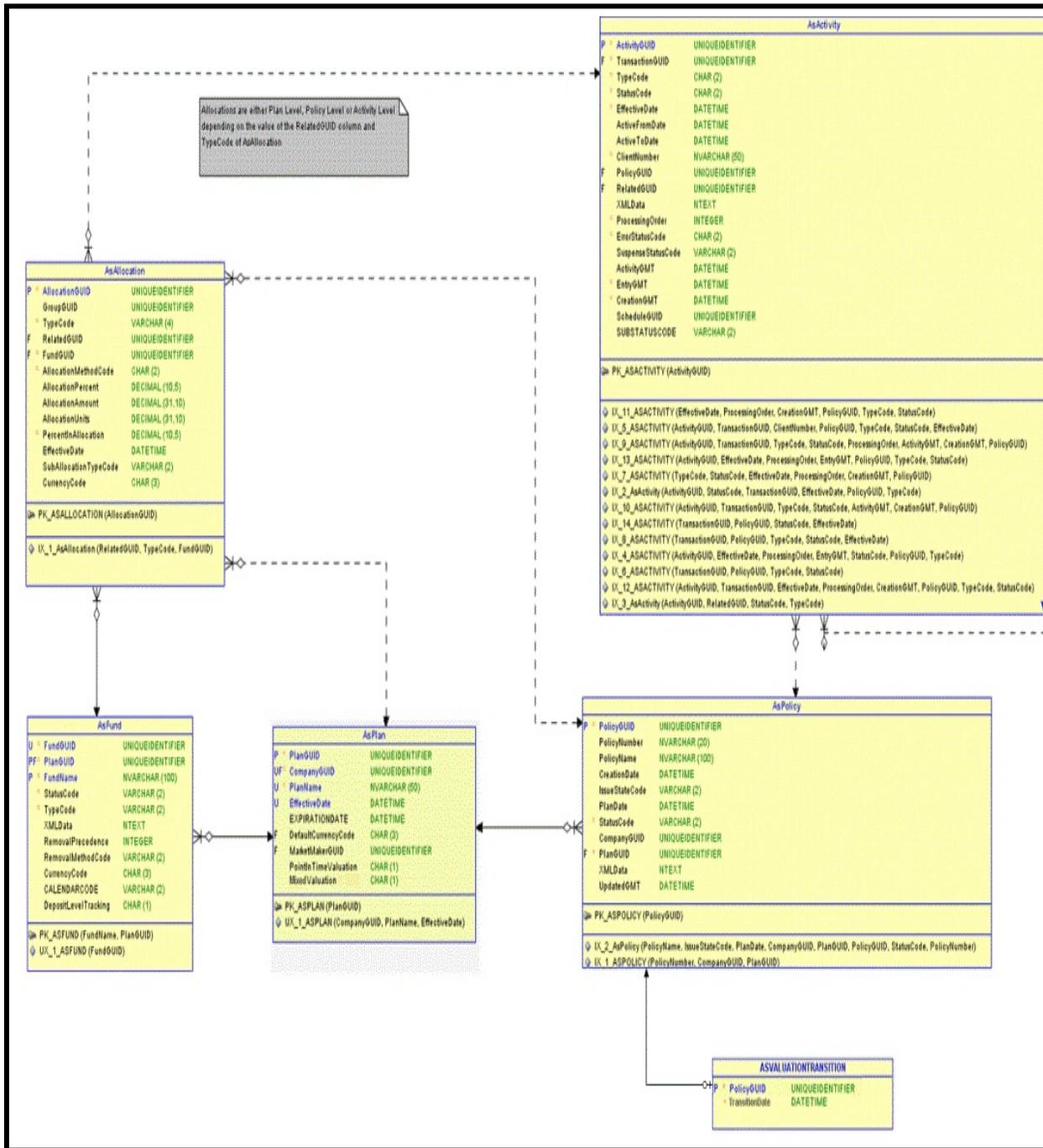
Valuation Tables



Fund Tables



Allocation Tables



You are here: Configuration > Valuation > Valuation Database Definitions

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Valuation Database Definitions

The following database definitions describe the tables used in valuation and the associated columns.

Select a link to view a specific database table and the associated definitions.

- [AsValuation](#)
- [AsFund](#)
- [AsNetAssetValue](#)
- [AsAllocation](#)

PIT Database Tables

- [AsPlan](#)
- [AsValuationTransition](#)

AsValuation Data Definitions

The AsValuation table contains the basic data set for each deal at the fund level.

Name	Datatype	Null Option	Description
ValuationGUID	uniqueidentifier	NOT NULL	Primary Key. A separate ValuationGUID is written for each fund affected by the transaction.
FundGUID	uniqueidentifier	NOT NULL	Link to AsFund
PolicyGUID	uniqueidentifier	NOT NULL	Link To AsPolicy
ActivityGUID	uniqueidentifier	NOT NULL	Link to AsActivity
EffectiveDate	datetime	NOT NULL	The activity as of date.
ActiveFromDate	datetime	NOT NULL	The system date.
ActiveToDate	datetime	NULL	The system date when the associated activity was reversed/undone.
RateLockDate	datetime	NULL	Date used for rate lookup.
MoneyTypeCode	varchar(2)	NOT NULL	Indicates the type of money used and can be found in AsCodeMoneyTypeCode.
RemovedFromDepositGUID	uniqueidentifier	NULL	DepositGUID where money is being removed.
SeedDepositGUID	uniqueidentifier	NULL	Original DepositGUID where money is deposited.
TaxlotGUID	uniqueidentifier	NULL	Foreign key into AsTaxLot.
TaxlotTradeDate	datetime	NULL	
TaxlotGainLoss	money	NULL	
ValuationAmount	money	NULL	A monetary amount of purchase or removal.
ValuationUnits	decimal(18,10)	NULL	Number of units purchased or removed.
ValuationGainLoss	money	NULL	Gain or loss due to backdated activities (activities effective on a date different than the system date).
ValuationPrincipal	money	NULL	Principal balance for simple interest calculations.
GainLossOnShadow	money	NULL	Gain/loss due to reversal/undo of activities that are effective on dates different than the system date.
Bucket			Used for Equity Index Fund.
FundCurrenyAmount			The valuation amount converted to a fund's currency.
CurrencyConversionCost			Cost of converting the plan's currency to the fund's currency. Usually seen on a premium.
DepositDepletedDate			Date the deposit's value was completely removed.
PriceDate			This is used for Unit Link Funds. This is the guarantee date which can be different from the effective date.
GainLossPriceDate			This is used for Unit Link Funds. The date that purchases and removals are actually bought and sold. This will determine the gain/loss for the company.
ShadowGainLossPriceDate			This is used for Unit Link Funds. The date that purchases and removals are actually bought and sold when the activity is reversed/undone.
BareSpreadAmount			Difference in unit value between bear price and bid or offer price. Deposits use offer price. Withdrawals use bear price.
CashValueAmount			

AsFund Data Definitions

Name	Datatype	Null Option	Description
FundGUID	uniqueidentifier	NOT NULL	Primary Key
PlanGUID	uniqueidentifier	NOT NULL	Link to AsPlan
FundName	uniqueidentifier	NOT NULL	Fund's name.
StatusCode	uniqueidentifier	NOT NULL	Indicates the status of the fund
TypeCode	datetime	NOT NULL	From AsCode.CodeValue where CodeName = AsCodeFundType
XMLData	datetime		
RemovalPrecedence	datetime	NULL	
RemovalMethodCode	datetime	NULL	
CurrencyCode	varchar(2)	NOT NULL	The ISO 4217 three letter currency code that is used for the fund.
CalendarCode	uniqueidentifier	NOT NULL	The fund's working calendar which associates it to a market.
DepositLevelTracking	singlecharacter	NULL	

AsNetAssetValue Data Definitions

Name	Datatype	Null Option	Description
NetAssetValueGUID	uniqueidentifier	NOT NULL	Primary Key
FundGUID	uniqueidentifier	NOT NULL	Link to AsFund.
EffectiveDate	Date	NOT NULL	Date of the unit deals.
NetAssetValue	Decimal(19,4)	NULL	Monetary value of the asset value.
UnitValue	Number(18,10)	NULL	Monetary value of a unit of the asset.
Dividend	Decimal(18,10)	NULL	Dividend Amount
MortalityAndExpense	Decimal(18,10)	NULL	The M&E for the asset on the effective date.
BareUnitValue	Number(18,10)	NULL	System does not set to bid if no bare or the same. Up to user to supply value.
OfferUnitValue	Number(18,10)	NULL	Up to user to supply value. If there is no Offer price, it should be set by the user to the Unit Value.

AsAllocations Data Definitions

Name	Datatype	Null Option	Description
AllocationGUID	CHARACTER(36)	NOT NULL	Primary key
GroupGUID	CHARACTER(36)	NULL	Identifies the group to which this allocation belongs
TypeCode	VARCHAR(2)	NOT NULL	01: Plan Default 02: Future Allocations 03: Activity Allocation 05:Segment Default 51: Original Benefit 61: Directed Deductions 99: Deleted Allocation (used by Reversal) 100: Conversion Premium 101: Conversion Premium Tax 102: Conversion Interest 103: Conversion Interest Expense ‐ Gain 104: Conversion Interest Expense ‐ Loss 105: Conversion Optional Death Benefit Fee 106: Conversion Withdrawl 107: Conversion Redemption Fee 108: Conversion Annual Contract Fee 109: Conversion Split Deposit Removal 110: Conversion Split Deposit Apply
RelatedGUID	CHARACTER(36)	NULL	Either PolicyGUID, ActivityGUID or PlanGUID
FundGUID	CHARACTER(36)	NOT NULL	Link to AsFund
AllocationMethodCode	VARCHAR(2)	NULL	From AsCode.CodeValue where CodeName ='AsCodeAllocation 01:Percent 02:Amount 03:Units 04:Preferred Pro Rata
AllocationPercent	DECIMAL(18,10)	NULL	Percent entered
AllocationAmount	DECIMAL(38,10)	NULL	Amount entered
AllocationUnits	DECIMAL(18,10)	NULL	Units entered
PercentInAllocation	DECIMAL(18,10)	NOT NULL	Prorata calculated percentage
EffectiveDate	TIMESTAMP	NULL	As of date

AsPlan Data Definitions

Name	Datatype	Null Option	Description
PlanGUID	uniqueidentifier	NOT NULL	Primary Key
CompanyGUID	uniqueidentifier	NOT NULL	Link to AsCompany.
PlanName	VARCHAR2	NOT NULL	Name of the plan.
EffectiveDate	Date	NULL	Effective date of the plan inception.
ExpirationDate	Date	NULL	Expiration date of the plan.
DefaultCurrencyCode	CHAR(3)	NULL	The default currency for any currency transactions on policies in this plan.
MarketMakerGUID	CHAR(36)	NULL	Link to AsMarketMaker.
PointInTimeValuation	CHAR(1)	NULL	Point-in-Time valuation indicator. T: Transition from Traditional to Point-in-Time valuation Y: Point-in-Time N NULL: Traditional valuation
MixedValuation	CHAR(1)	NULL	Mixed valuation indicator. Y: Valuation records written as determined by AsValuationTransition:TransitionDate. N NULL: Always write Point-in-Time valuation records.

AsValuationTransition Data Definitions

Name	Datatype	Null Option	Description
PolicyGUID	uniqueidentifier	NOT NULL	Primary Key
TransitionDate	Date	NOT NULL	Transition date for transition from Traditional to Point-in-Time valuation.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Valuation > AsValuation XML Example

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

AsValuation XML Example

```
<Valuation>
  <EffectiveDate>01/04/2011</EffectiveDate>
  <ActiveDate>01/29/2031</ActiveDate>
  <Policy>
    <PolicyGUID>FEEE6129-C0EE-4F57-9FBC-F69F428033E4</PolicyGUID>
    <EffectiveDate>01/04/2011</EffectiveDate>
    <ActiveDate>01/29/2031</ActiveDate>
    <CashValue>1026850.47</CashValue>
    <SurrenderValue>0.0</SurrenderValue>
    <Principal>836215.34</Principal>
    <Gain>190635.13</Gain>
    <NetGain>190635.13</NetGain>
    <FreeAmount>0.0</FreeAmount>
    <FreeAmountXML/>
    <MVAAmount>0.0</MVAAmount>
    <RedemptionAmount>0.0</RedemptionAmount>
    <GuaranteedAmount>0.0</GuaranteedAmount>
    <PolicyValuesXML>
      <PolicyValues>
        <ContractValue>1026850.47</ContractValue>
        <PolicyCashValue>1026850.47</PolicyCashValue>
        <SegmentGUID>75C71375-EDBE-4373-9202-B02E-000000000000</SegmentGUID>
        <Zero>0</Zero>
        <Checked>CHECKED</Checked>
        <Yes>Yes</Yes>
        <EffectiveDate>01/04/2011</EffectiveDate>
        <Blank/>
        <No>No</No>
      </PolicyValues>
    </PolicyValuesXML>
    <FundList>
      <Fund>
        <FundGUID>19FD991F-E071-4E66-B02E-DAD0991E-000000000000</FundGUID>
        <FundName>GMIB</FundName>
        <TypeCode>05</TypeCode>
        <UnitValue>0.0</UnitValue>
        <Units>0.0</Units>
        <CashValue>1539964.76</CashValue>
      </Fund>
    </FundList>
  </Policy>
</Valuation>
```

```
<SurrenderValue>0.0</SurrenderValue>
<Principal>1000000.0</Principal>
<Gain>539964.76</Gain>
<NetGain>539964.76</NetGain>
<FreeAmount>0.0</FreeAmount>
<MVAAmount>0.0</MVAAmount>
<RedemptionAmount>0.0</RedemptionAmount>
<ShortTermRedemptionUnits>0.0</ShortTerm
<GuaranteedAmount>0.0</GuaranteedAmount>
<DepositList>
  <Deposit>
    <ValuationGUID>39E64C08-9605-4:
    <EffectiveDate>03/01/2002</Effi
    <RateLockDate/>
    <SeedDepositGUID>E5155974-BEF9
    <MoneyTypeCode>04</MoneyTypeCo
    <InitialUnits>0.0</InitialUnit:
    <InitialCashValue>1000000.0</I
    <Units>0.0</Units>
    <CashValue>1539964.7629756918<
    <SurrenderRate>0.0</SurrenderR
    <SurrenderValue>0.0</Surrender'
    <Principal>1000000.0</Principa.
    <Gain>539964.7629756918</Gain>
    <NetGain>539964.7629756918</Ne
    <InterestBonus>0.0</InterestBo
    <TransactionName>InitialPremiu
    <MVAAmount>0.0</MVAAmount>
    <MVAFormulaXML/>
    <RedemptionAmount>0.0</Redempt.
    <ShortTermRedemptionUnits>0.0<
    <GuaranteedAmount>0.0</Guaranti
    <ActivityGUID>E5155974-BEF9-49:
    <ActivityGMT>06/08/2009</Activ.
    <RemovalList>
      <Removal>
        <ValuationGUID/>
        <EffectiveDate>03/01.
        <MoneyTypeCode>06</M
        <Units>0.0</Units>
        <Amount>0.0</Amount>
        <Principal>0.0</Prin
```

```
<UnitValue>0.0</UnitValue>
<InterestRate>0.0</InterestRate>
<GuaranteedInterestRate>0.0</GuaranteedInterestRate>
<CurrentInterestRate>0.0</CurrentInterestRate>
<CashValue>0.0</CashValue>
<Days>0</Days>
<InterestAmount>0.0</InterestAmount>
<TransactionName/>
<GuaranteedAmount>0.0</GuaranteedAmount>
<ActivityGUID/>
<ActivityGMT/>
<AdditionalAdditiveInterest/>
</Removal>
<Removal>
  <ValuationGUID/>
  <EffectiveDate>01/04/2008</EffectiveDate>
  <MoneyTypeCode>06</MoneyTypeCode>
  <Units>0.0</Units>
  <Amount>0.0</Amount>
  <Principal>0.0</Principal>
  <UnitValue>0.0</UnitValue>
  <InterestRate>0.05</InterestRate>
  <GuaranteedInterestRate>0.05</GuaranteedInterestRate>
  <CurrentInterestRate>0.05</CurrentInterestRate>
  <CashValue>0.0</CashValue>
  <Days>3230</Days>
  <InterestAmount>5399.00</InterestAmount>
  <TransactionName/>
  <GuaranteedAmount>0.0</GuaranteedAmount>
  <ActivityGUID/>
  <ActivityGMT/>
  <AdditionalAdditiveInterest/>
</Removal>
</RemovalList>
</Deposit>
</DepositList>
</Fund>
<Fund>
  <FundGUID>6B4937BB-E9CB-4672-8F80-5EB8B30A0000</FundGUID>
  <FundName>Balanced (Band125)</FundName>
  <TypeCode>02</TypeCode>
  <UnitValue>10.0</UnitValue>
```

```
<Units>102685.047</Units>
<CashValue>1026850.47</CashValue>
<SurrenderValue>0.0</SurrenderValue>
<Principal>836215.34</Principal>
<Gain>190635.13</Gain>
<NetGain>190635.13</NetGain>
<FreeAmount>0.0</FreeAmount>
<MVAAmount>0.0</MVAAmount>
<RedemptionAmount>0.0</RedemptionAmount>
<ShortTermRedemptionUnits>0.0</ShortTerm
<GuaranteedAmount>0.0</GuaranteedAmount>
<DepositList>
  <Deposit>
    <ValuationGUID>610ED818-7C06-41
    <EffectiveDate>12/04/2010</Effi
    <RateLockDate/>
    <SeedDepositGUID>E5155974-BEF9
    <MoneyTypeCode>01</MoneyTypeCo
    <InitialUnits>102685.047</Init:
    <InitialCashValue>1026850.47</:
    <Units>102685.047</Units>
    <CashValue>1026850.47</CashVal
    <SurrenderRate>0.0</SurrenderR
    <SurrenderValue>0.0</Surrender
    <Principal>836215.34</Principa
    <Gain>190635.13</Gain>
    <NetGain>190635.13</NetGain>
    <InterestBonus>0.0</InterestBo
    <TransactionName>RebalanceTran:
    <MVAAmount>0.0</MVAAmount>
    <MVAFormulaXML/>
    <RedemptionAmount>0.0</Redempt:
    <ShortTermRedemptionUnits>0.0<
    <GuaranteedAmount>0.0</Guaranti
    <ActivityGUID>DA2B8667-A5F4-44:
    <ActivityGMT>06/08/2009</Activ:
    <RemovalList/>
  </Deposit>
</DepositList>
</Fund>
</FundList>
</Policy>
```

</Valuation>

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Valuation > Valuation Configuration > High Level Steps to Configure for Valuation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

High Level Steps to Configure for Valuation

Several portions of configuration must be completed before the valuation functionality will become enabled. Keep the following information in mind when [configuring valuation](#).

Plans

Make sure the type of valuation is selected at the plan level. To check this on an existing plan, right-click on the plan name and select **Edit Plan**. This opens the [Plan Maintenance window](#), where the type of valuation can be selected.

 If the plan previously used traditional valuation and now needs to use Point in Time valuation, the valuation type can be converted.

Funds

Initially, funds must be configured for each plan that will use valuation. After funds are configured, additional business rules can be configured as desired to suit specific business needs.

Business Rules

The following list contains commonly used valuation-related business rules:

- InterestRateCalculation business rule
- Rates, as required for use in the InterestRateCalculation business rule
- Valuation Calculation, such as FreeAmountFormula and/or MVA
- For variable funds, GainLossCalculations business rule
- PolicyValues business rule
- ValuesScreen and/or InquiryScreen business rule

There are other rules that may be configured depending on the type of funds you are using. Refer to the [Funds](#) section for additional information about fund types.

Transactions

In order for Valuation to be executed, transactions requiring Valuation must have the <Valuation> section properly configured. In many cases where Allocations and Assignments are also used, the <Allocations> and <Assignment> sections must also be configured.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Valuation > Valuation Configuration > Configure the Plan's Funds

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Configure the Plan's Funds

Funds must be set-up prior to configuring valuation calculations specific to the company's business requirements. See the [Funds](#) section for details on entering funds into OIPA. There are different sets of steps according to what type of funds you are configuring.

Steps According to Funds

- [Fixed Funds](#)
- [Variable Funds](#)

Fixed Funds

If using **fixed funds**, complete the following steps.

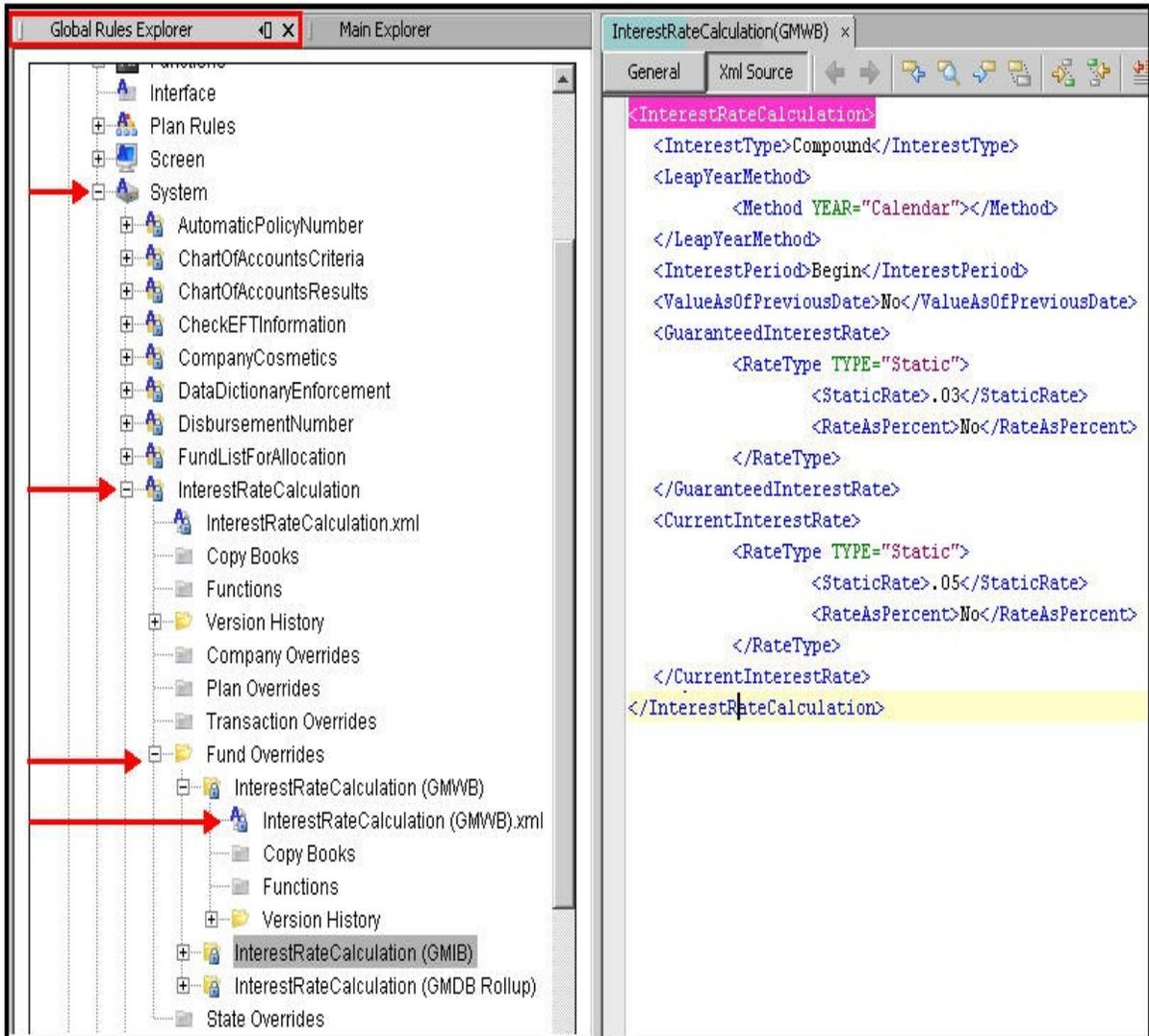
1. Configure the interest rate via the InterestRateCalculation rule.
2. Make sure the interest rate is added to the system if it will be looked-up by the InterestRateCalculation rule.

Interest Rate Logic

The **InterestRateCalculation** rule supports interest calculation. It supports guaranteed interest calculations, bonus interest calculations, current interest calculations, compound interest, simple interest, leap year methods, numerator calculation methods and previous day valuation.

Steps to Configure Interest Rate Calculation

1. Navigate to the **Global Rules Explorer**.
2. Expand the **Business Rule** folder.
3. Expand the **System** folder.
4. Right-click on **InterestRateCalculation**.
5. Select **New InterestRateCalculation Override**. The Override Wizard opens.
6. Select the **Next** button for step one.
7. Select the company, plan and fund where this override should be used.
8. Select **Finish**.
9. Configure your interest logic in the XML Source Pane of the override of the **InterestRateCalculation** that was created. View the **V9 XML Configuration** topic in this help system for additional information. View **Business Rules | System Rules | InterestRateCalculation**.



InterestRateCalculation Rule in Global Explorer

Interest Rate to be used with the Fund

In addition to configuring the logic for the interest rate calculation you must also ensure there is an interest rate for the fund to use. In the InterestRateCalculation rule you can enter it OR you can retrieve it from AsRate via the AsRateGroup name. If you decide to retrieve the rate from the rate tables, please see the [Rates](#) section for details on uploading rates via RateGroups in the Admin Explorer.

Variable Funds

If using variable funds, complete the following steps.

1. Verify that the AsNetAssetValue table has unit values for funds to use.
2. Configure the PrecisionValues rule.

Step 1: Verify the AsNetAssetValue Table

A very important part of this is the price of a fund unit at a specific point in time. The AsNetAssetValue table stores the unit values for the various funds. If you configured funds whose value varies due to market conditions, then those values must be stored in the AsNetAsset table according to dates.

To update the database manually use an INSERT statement.

Example of a Manual SQL Statement

```
INSERT INTO AsNetAssetValue
(NetAssetValueGUID,FundGUID,EffectiveDate,NetAssetValue,Un
VALUES (newid(),'054BD142-1AD5-7074-5E67-
68EA40C405FD', '7/23/2030',11,11,0,0)
```

Example of a Manual Oracle Statement

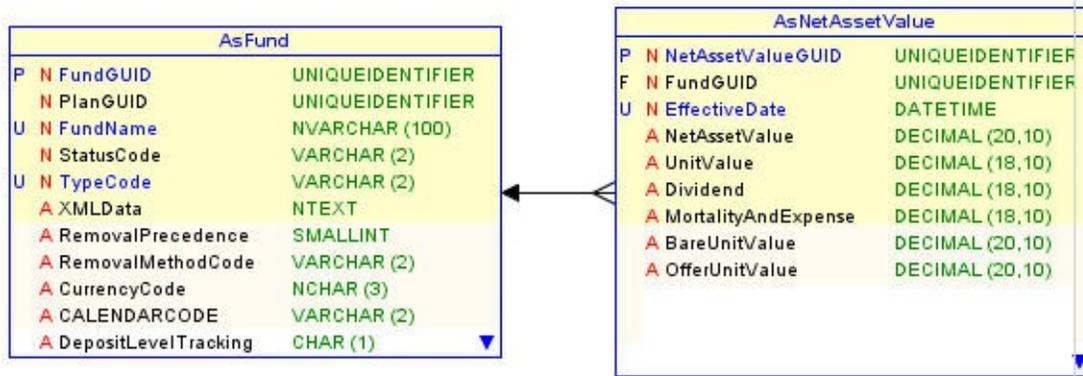
```
INSERT INTO AsNetAssetValue (NetAssetValueGUID,
FundGUID, EffectiveDate, NetAssetValue, UnitValue, Dividend,
MortalityandExpense,
BAREUNITVALUE,OFFERUNITVALUE)VALUES ('7D1D5390-
F351-11E0-B1C5-A3C64724019B', 'A5027FFD-2925-4156-
A62B-46A075C7C379', TO_TIMESTAMP('29-JAN-31','DD-MON-
RR'), 8, 8, 0, 0, 0, 0)
```

Example of a Manual DB2 Statement

```
INSERT INTO AsNetAssetValue (NerAssetValueGUID,
FundGUID, EffectiveDate, UnitValue, BareUnitValue,
OfferUnitValue) Values ('03E375CD-EA75-49EA-8A08-
0997D1C9C215', '00A5133A-DB51-4B96-8E25-699500A1EB3B',
TO_TIMESTAMP('29-JAN-3100:00:00.123000','DD-MON-
RRHH24:MI:SS.FF'), 1, 1, 1)
```

- The Files administration functionality allows you to select, transform and upload incoming data and insert it directly into the OIPA database. Please see the [Files](#) section for additional information on how to configure.

You can review the [Funds](#) data definitions to see how the Net Unit Values are associated.



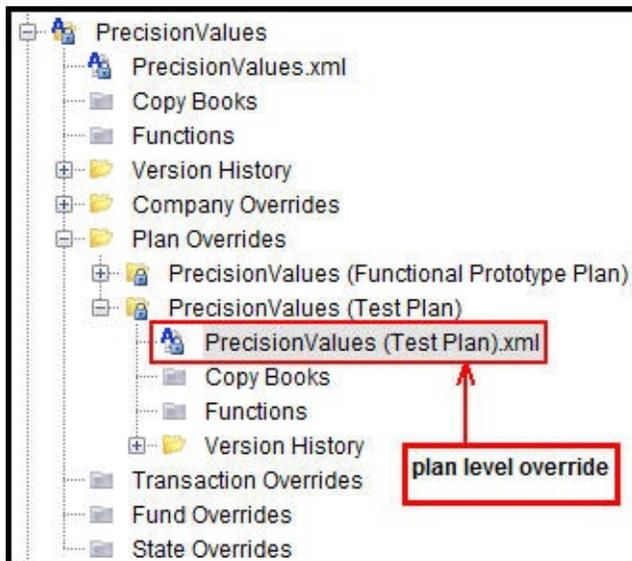
Step 2: Configure the PrecisionValues Rule

The [PrecisionValues rule](#) provides rounding precision convention (i.e. number of digits allowed after the decimal point) for the display of Unit Values and for the calculation and display of Number of Units.

To use the same unit precision across multiple plans, the PrecisionValue business rule should be configured at a global level.

Steps to Configure Precision Values Rule Override

1. Navigate to the **Global Rules Explorer**.
2. Expand the **Business Rule** folder.
3. Expand the **System** folder.
4. Right-click **PrecisionValues**.
5. Select **New PrecisionValues override**. The Override Wizard will open.
6. Select **Next** for step one.
7. Select the level you want the unit precision to apply to.
8. Select **Finish**.
9. Configure the unit precision value in the XML Source Pane of the override of the PrecisionValues rule that was created. View the **V9 XML Configuration** topic in this help system for additional information. View **Business Rules | System Rules | PrecisionValues**.



PrecisionValue Rule Overridden at PlanLevel in Global Explorer

You are here: Configuration > Valuation > Valuation Business Rules > Configure the ValuationDetails Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

ValuationDetails Rule

This global rule indicates at what level rounding will take place during valuation and assignment processing. When the level is Deposit, each deposit will be rounded to the currency's specific rounding rules. When the level is Fund, each fund will be rounded to the currency's specific rounding rules. The deposits within the fund are not rounded and are carried to their maximum decimal places.

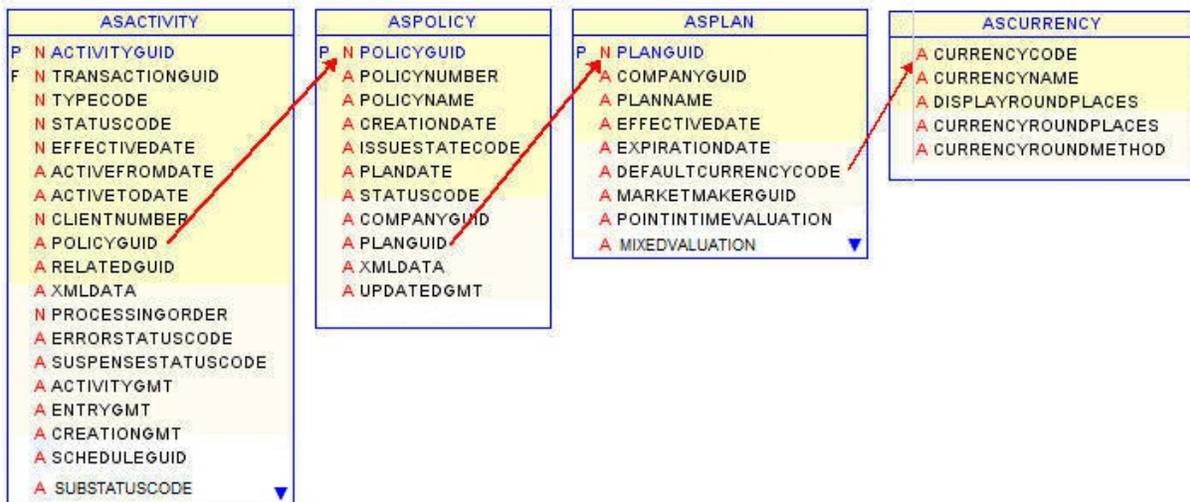
Steps to Configure the Valuation Details Rule

1. Navigate the Global Rules Explorer.
2. Expand the **Business Rules | System** folder.
3. Check-out the ValuationDetails.xml file.
4. Configure the rule as shown in the XML Syntax example below.
5. Check-in the rule to save your changes to the database.

XML Syntax Example for ValuationDetails rule

```
<ValuationDetails>  
  <Rounding LEVEL="Deposit"></Rounding>  
</ValuationDetails>
```

The following image shows how currency code affects rounding.



Currency Code and Rounding

You are here: Configuration > Valuation > Valuation Business Rules > Configure the PolicyValues Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

PolicyValues Rule

This rule is configured at the plan level. It is used by the ValueScreen rule transactions and InquiryScreen rules that perform valuation. This business rule is the ONLY configurable portion of the valuation structure. The rest of the valuation structure is hard coded by the system. This rule specifies the math variables that supplement OIPA's pre-defined valuation structure. It allows for the calculation of product specific policy, fund and deposit values as needed. Review the [OIPA Valuation Mechanism](#) for a description of the valuation structure.

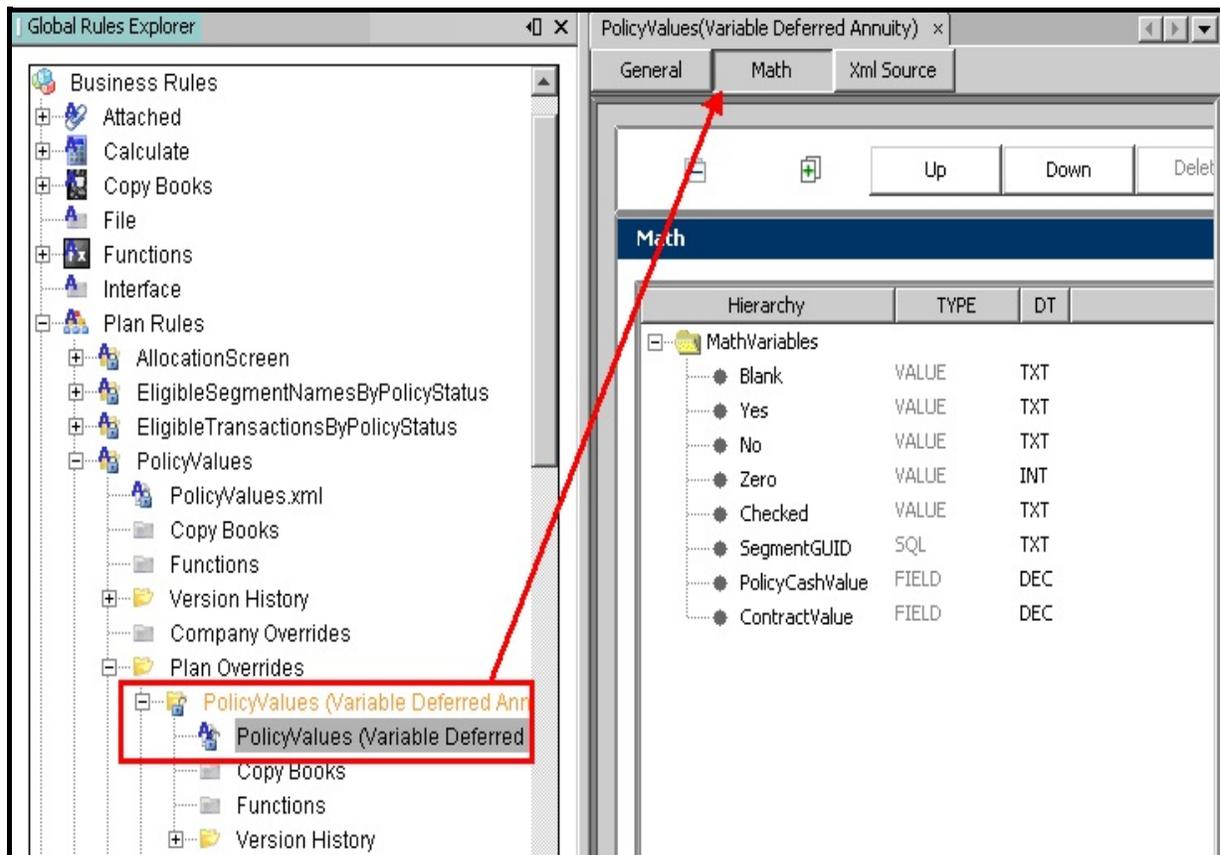
When this rule is used in conjunction with the ValueScreen rule, it becomes the data source for the display of the OIPA Value Screen. The Value screen rule is configured to control the display format of data pulled from the PolicyValues rule. Please refer to the [Value screen](#) section to learn how to configure the Value screen.

When this rule is used in conjunction with a transaction, the transaction's XML must contain the <Valuation> element. This element has a POLICYVALUES attribute that controls execution of the PolicyValues rule. By default, the PolicyValues rule will execute with valuation. However, this attribute may turn that mechanism off. The POLICYVALUES attribute controls the value processing as follows:

- When the attribute is set to Yes or is not provided, policy values will be executed.
- When the attribute is set to No, calculation of policy values will be turned off and the PolicyValue information in the Valuation XML ([traditional valuation](#)) will be empty.

Steps to Configure the PolicyValues Business Rule

1. Navigate to the Plan folder via the Main Explorer.
2. Expand the **Plan Rules | PolicyValues** folder.
3. Check-out the PolicyValues.xml file.
4. Configure the business rule.
5. Check-in the rule to save the changes to the database.



PolicyValues Rule in Global Rules Explorer

Each math variable from the PolicyValues business rule will be listed in the ValuationXML. The valuation structure has a special section that begins and ends with the <PolicyValuesXML> element. An example of the policy values XML is illustrated below.

```
<PolicyValuesXML>
  <PolicyValues>
    <Valuation_Policy_Principal>60000.0</Valuation_Policy_Principal>
    <Plan_MaximumPremiumOver70>500000.0</Plan_MaximumPremiumOver70>
    <Policy_CompanyGUID>18B611A8-4429-4C67-94E6-3F4A882C9A8D</Policy_CompanyGUID>
    <Valuation_Deposit_E63CA0CA-EA28-1375-4D30-A580031F0DE8_Principal>60000.0</Valuation_Deposit_E6
    <Valuation_Fund_D98E91F8-C1EC-4A65-1111-8CE2F378EC98_FreeAmount>0.0</Valuation_Fund_D98E91F
    <Plan_ContractFeeAmount>30.0</Plan_ContractFeeAmount>
    <ASUTL_WARNINGLIST></ASUTL_WARNINGLIST>
    <Policy_IssueStateCode>15</Policy_IssueStateCode>
    <Valuation_Fund_D98E91F8-C1EC-4A65-1111-8CE2F378EC98_ShortTermRedemptionUnits>0.0</Valuation_
    <Policy_SweepDate>2030-05-18 00:00:00.0</Policy_SweepDate>
    <Plan_MinimumSWPAmount>100.0</Plan_MinimumSWPAmount>
    <Policy_PolicyGUID>86325C01-0B3F-57B8-F5DD-B42BD4C6ADCF</Policy_PolicyGUID>
    <Policy_PlanDate>2031-05-07 00:00:00.0</Policy_PlanDate>
    <No>No</No>
  </PolicyValues>
</PolicyValuesXML>
```

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Valuation > Valuation Business Rules > GainLossCalculation Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

GainLossCalculation Rule

This rule is usually configured at the plan level or the transaction level. Gain/loss is the gain or loss incurred by a company when a financial activity buys and/or sells variable fund units. This rule determines whether or not the system should calculate Gain/Loss for back-dated activities. For example, if an activity was processed on a date that is a couple of days after the effective date of the activity, then the difference in the unit price would be calculated as a gain or loss to the company.

There are two columns in the AsValuation table that are used to store Gain/Loss information.

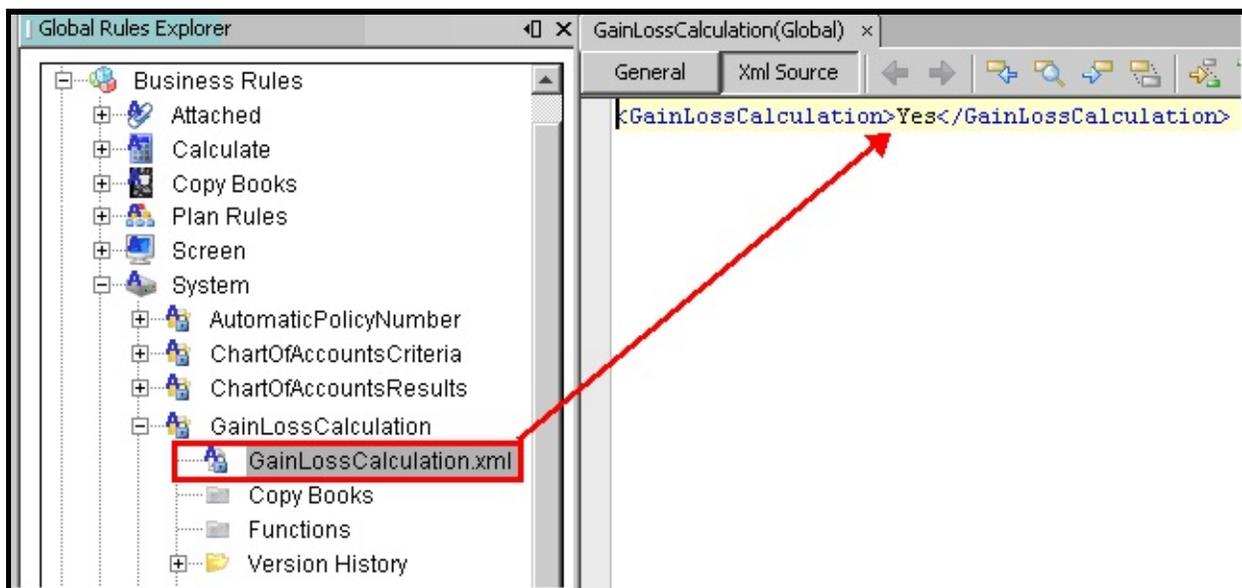
- ValuationGainLoss: ValuationGainLoss is populated during forward activity processing.
- GainLossOnShadow: GainLossOnShadow is populated when a financial activity is reversed.

XML Schema

```
<GainLossCalculations> [Yes|No]</GainLossCalculations>
```

Steps to Configure the GainLossCalculation Business Rule

1. Navigate to the Global Rules Explorer.
2. Expand the **Business Rules | System | GainLossCalculation** folders.
3. Check-out the GainLossCalculation.xml file or create a [plan or transaction level override of the rule](#).
4. Configure the business rule.
5. Check-in the rule to save the changes to the database.



GainLossCalculation Rule in Global Rules Explorer

You are here: Configuration > Valuation > Valuation Configuration > Configure the Transaction that Runs Valuation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Configure the Transaction that Runs Valuation

Certain transactions require that a policy is valued because money is either moving in, transferring between funds or money is being withdrawn. Transactions such as Withdraw, FullSurrender or Additional Payment would affect the policy's value hence valuation would need to be configured.

The first step is to [create a new transaction](#) in the appropriate plan. Once the transaction is created, [security must be added](#) and the transaction must be assigned a status in the [EligibleTransactionsByPolicyStatus rule](#), which will allow OIPA users to access the transaction.

When the new transaction is first checked-in, a translation must be assigned to the transaction name. This is a required step before the transaction can be accessed in OIPA.

After the initial set-up steps are complete, the transaction sections can be configured.

The four major parts to consider when configuring a transaction that will run valuation are:

1. [Allocation](#)
2. [Valuation](#)
3. [Assignment](#)
4. [Math](#)

You are here: Configuration > Valuation > Valuation Configuration > Transaction Allocation Configuration

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transaction Allocation Configuration

Allocation configuration allows the end user to indicate the desired money movements for the transaction. Using the <Allocation> elements and attributes will create the allocation structure displayed on the transaction. The logic configured for the allocation structure defines the funds included, excluded, the method in which allocations are displayed and other display guidelines.

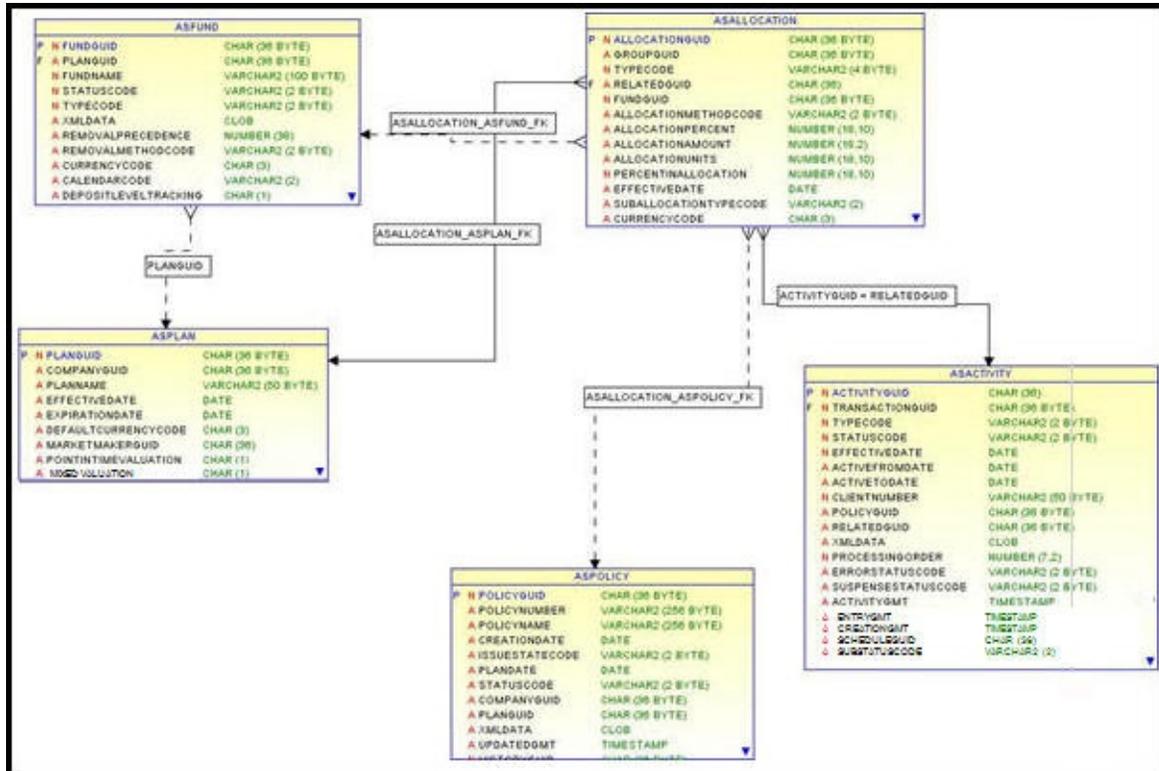
Funds should already be set-up in the system prior to configuring allocations. Refer to the section on [Adding Funds](#) for additional information. Refer to the [Allocations pane](#) section to review the steps for configuring transaction level allocations. Additional allocation information is also available in [Allocations and Assignments](#) in the Fund section.

Example of <Allocation> configuration from the Variable Deferred Annuity product template

```
<Allocation FUNDLIMIT="18" EXCLUDETYPE="05,06,07"  
ALLOWMIXEDMETHODS="No"/>
```

Database Tables

Allocation records will be created and stored in the AsAllocation table.



AsAllocation and Related Tables

Activity Detail Screen in OIPA

The images below provide an example of the Allocations Detail screen for an activity in the Variable Deferred Annuity product template.

ADDITIONALPAYMENT () - VDA31011738

Details | Allocations

ALLOCATIONS

ACTIVITY

Number Of Funds: 2 Allocation Method: Percent

Equal Percent

Fund	Percent
Balanced	50%
Equity Income	50%
Total: 100%	

Ok Cancel

ADDITIONALPAYMENT (1/29/2031) - VDA31011738

Detail | Entry Fields | Allocations | Math | Spawn | Accounting | Valuation | Disbursement

Suspense :

FUND ALLOCATION

Fund	Percent	Amount	Units	Unit Value
Mid Cap	33.34%	0.00 USD	0.0000000000	13.1700000000 USD
Equity Income	33.33%	0.00 USD	0.0000000000	13.1300000000 USD
Balanced	33.33%	0.00 USD	0.0000000000	14.1400000000 USD

FUND ACTIVITY

Fund Allocation	Type	Amount	Units	G/L	Shadow G/L
Mid Cap (Band125)	Premium	8,335.00 USD	632.8777524677	0.00 USD	
Equity Income (Band125)	Premium	8,332.50 USD	634.6153846154	0.00 USD	
Balanced (Band125)	Premium	8,332.50 USD	589.2857142857	0.00 USD	

You are here: Configuration > Valuation > Valuation Configuration > Transaction Valuation Configuration

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transaction Valuation Configuration

Valuation configuration tells the system to run the valuation engine. Running the valuation engine means the system will create a valuation structure in memory that pulls all the policy records from the AsValuation table and performs core and configured calculations. The resulting values can be accessed by the transaction's <Math> section. <MathVariables> then can be used in <Assignment>.

Valuation can be calculated multiple times for different dates within a single activity. This is supported by allowing transaction math to execute valuation and pass the results of that valuation to math variables. Refer to the [MathStatement](#) section and the [ValuationValue](#) section for configuration information. This method can be used in addition to running traditional activity valuation using the activity effective date.

The activity's valuation, if calculated using a <Valuation> element, will exist in memory and its values will appear on the Valuation tab of the Activity Results screen. Accessing the activity's valuation can be accomplished by using the 'Valuation:' syntax. The activity's effective date will always be the activity's valuation date.

Assignment configuration uses the Allocation or ReassignAllocations information and writes records to AsValuation for each deposit by fund and Money type.

Example of <Valuation> configuration from the Variable Deferred Annuity product template

```
<Valuation>  
<EffectiveDateNUVMustExist>Yes</EffectiveDateNUVMustEx.  
<SystemDateNUVMustExist>No</SystemDateNUVMustExist>  
</Valuation>
```

In the Rules Palette, the Valuation Engine is configured in the General pane.

General	Allocations	Address	Fields	Events
<input checked="" type="checkbox"/> Transaction				
<input checked="" type="checkbox"/> Effective Date				
<input type="checkbox"/> Values Block				
<input type="checkbox"/> Withholding				
<input type="checkbox"/> Valuation				
<input checked="" type="checkbox"/> Valuation				
<input type="checkbox"/> Effective Date NUV Must Exist				
<input type="checkbox"/> System Date NUV Must Exist				
<input type="checkbox"/> Suspense				
<input type="checkbox"/> MultiSuspense				
<input type="checkbox"/> Allow Comments				
<input type="checkbox"/> Transitions				

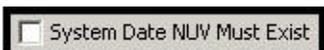
Valuation Check Box and Field Descriptions



- The Valuation checkbox should be checked if the transaction will need valuation to run.
- This **MUST** be checked if Assignments are configured.
- This **MUST** be checked if you want to pull values from the Valuation XML.



If this box is checked, the Unit Values (NUVs) as of the activity's Valuation Date must be present in order for the activity to process at all. If there are no Unit Values, the system displays an error message stating **NUV's Missing (Effective Date)**.



If the box is checked, Unit Values as of the System Date must be present in order for the activity to process at all. If no Unit Values are present, the system displays an error message stating **NUV's Missing (System Date)**.

If the box is unchecked for a variable product and NUV's do not exist for the System Date, but Unit Values are present for the Valuation Date, the activity will process in its entirety, except for the calculation of any Gain/Loss. The Activity List screen will display a lightning bolt icon next to the activity with a status of Gain/Loss Pending until Unit Values are added for the System Date. After Unit Values have been entered for the System Date and either the lightning bolt is clicked again or Cycle is run, the Activity Status will change to Active and Gain/Loss will be calculated.



For Fixed products, this should never be checked.

You are here: Configuration > Valuation > Valuation Configuration > Transaction Assignment Configuration

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transaction Assignment Configuration

The Assignment element is used to configure the application of money into or out of a policy and consequently has an impact on the policy's value. Assignment processing controls the creation of AsValuation records, which record deposits into and out of funds listed in the Allocation records. Assignment processing is triggered by the presence of the Assignment configuration within a transaction's configuration. Each sub-element of Assignment must have a money type associated with it, since money can be applied to a policy from different sources and must be tracked for auditing and tax purposes.

The application of money-in needs to come from a positive money value from a field or MathVariable. The application of money-out should be from a negative money value from a field or MathVariable.

The Rules Palette offers a variety of ways to assign money-in and money-out. Depending on the assignment chosen the system will require specific input and process money in or out in a pre-defined manner.

For a complete list of assignment types, see the XML Configuration Guide in the **Help** menu at the top of the screen. Assignment information can be found under **Transactions | Transaction Elements | Assignment Elements**.

Example of <Valuation> configuration from the Variable Deferred Annuity product template

```
<Assignment TYPE="Apply">  
<MoneyType NAME="GrossAmountMV">01</MoneyType>  
</Assignment>
```

Note: For additional information on how to configure assignments in the Rules Palette, see the [Assignment](#) section.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Valuation > Valuation Configuration > Transaction Math Configuration

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Transaction Math Configuration

In the Math section of the transaction, values in the valuation structure may be used. This is accomplished by using the Field variable type. Valuation is used as a prefix. Each node in the valuation structure is a continuation of the prefix, separated by a colon, until the desired value is reached. The name of the math variable will then store the desired value.

Steps to Configure Math in a Transaction

1. In the Rules Palette, configure a Field math variable and use Valuation for the field prefix. This tells the system to retrieve a value from the XMLData structure in the AsValuationXML.
2. In the Search Palette window, select the **Field** math variable and drag it onto the Math section under Hierarchy.
3. Enter the name for the math variable. Make sure the new math variable is selected.
4. Select the Data Type.
5. Enter **Valuation:** in the Prefixes field and all node names separated by a colon (:).
6. Enter the actual name of the field whose value is desired.

FullSurrender(Variable Deferred Annuity) x

General Allocations Fields Events Math Assignment Spawn Xml Source

Math

Hierarchy	TYPE	DT	Attributes	Text	DT
PayoutMethodMV	FIELD	TXT	Activity:PayoutMethod		Txt
IssueDateMV	FIELD	DAT	Activity:IssueDate		DT
PolicyCashValueMV	FIELD	DEC	Valuation:Policy:CashValue		DT
PreTEFRAAmount	VALUE	DEC	0.0		DT
PostTEFRAAmount	VALUE	DEC	0.0		DT
CostBasis	VALUE	DEC	0.0		DT

Name: PolicyCashValueMV

Data Type: DECIMAL

Variable Type: FIELD

Log: Yes No

Round: Yes No Decimal places:

Default:

Prefixes: Valuation:

Value: Policy:CashValue

Math Pane of a Transaction

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Valuation > Valuation Business Rules > InterestRateCalculation Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

InterestRateCalculation Rule

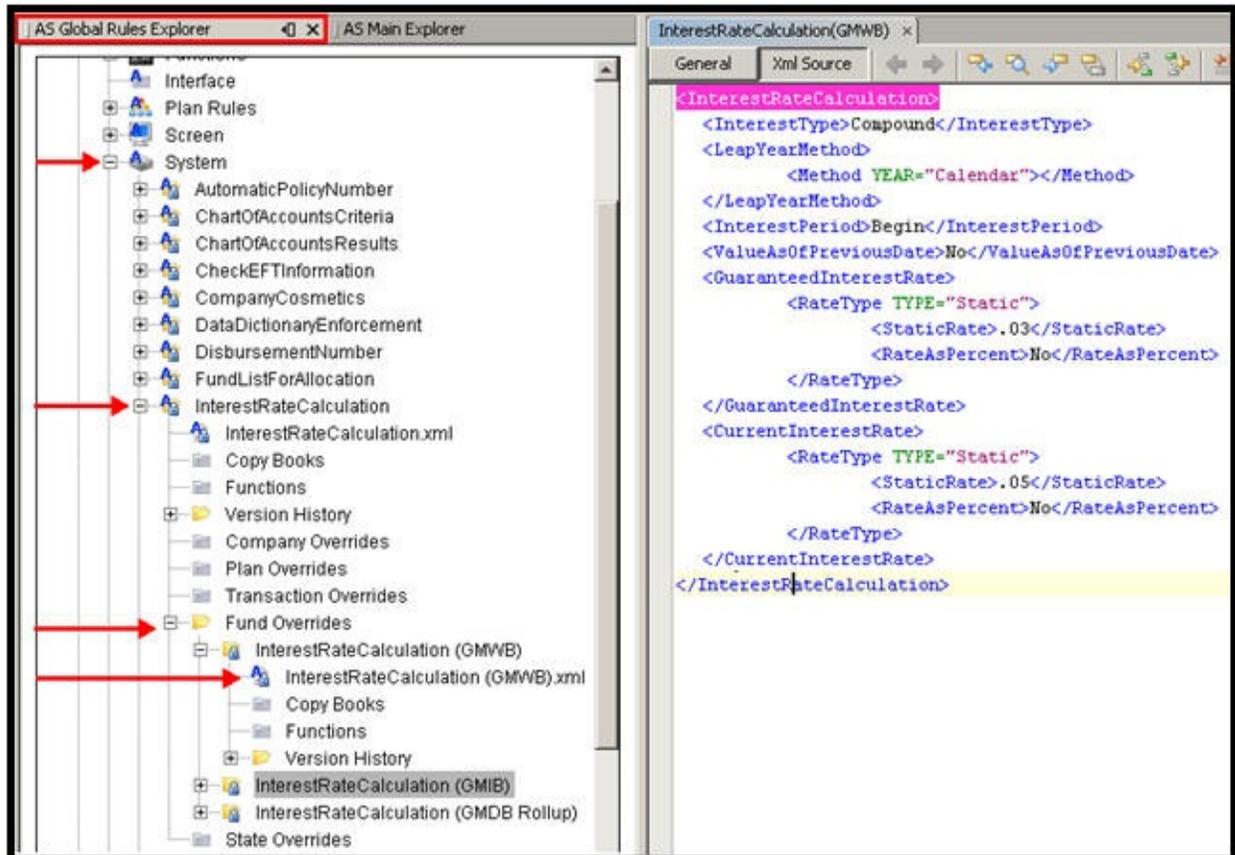
The InterestRateCalculation rule supports fixed interest calculation. It supports guaranteed and current interest calculations as well as any number of additive interest rates such as bonus interest rates. The interest calculation may be compounded or simple and there are methods for different interpretations of the year (360, 365, 365 plus leap year determination).

In its simplest implementation, the InterestRateCalculation rule can be used to set static fixed rates for a fixed fund. More complex configurations, such as retrieving interest rates from a rate table can also be configured. Please refer to the **XML Configuration Guide** under **Business Rules | System Rules | InterestRateCalculation** . A [prototype sample](#) is also included in this help.

Steps to Configure the InterestRateCalculation Business Rule

1. Navigate to the Global Rules Explorer.
2. Expand the Business Rule folder.
3. Expand the System folder.
4. Right-click on InterestRateCalculation.
5. Select **New InterestRateCalculation override**. The Override Wizard opens.
6. In step 1, select **Next**.
7. In step 2, select the company, plan and fund where this override will be used.
8. Select **Finish**.

Configure the interest calculation logic in the Xml Source Pane of the override of the InterestRateCalculation that was created. Refer to InterestRateCalculation in the V9 XML Configuration Guide under **Tools | Help** for a complete explanation of XML elements and attributes.



InterestRateCalculation rule in Global Rules Explorer

Example of InterestRateCalculation business rule for a fixed fund:

```
<InterestRateCalculation>
  <InterestType>Compound</InterestType>
  <LeapYearMethod>
    <Method YEAR="Calendar"></Method>
  </LeapYearMethod>
  <InterestPeriod>Begin</InterestPeriod>
  <ValueAsOfPreviousDate>No</ValueAsOfPreviousDate>
  <GuaranteedInterestRate>
    <RateType TYPE="Static">
      <StaticRate>.03</StaticRate>
      <RateAsPercent>No</RateAsPercent>
    </RateType>
  </GuaranteedInterestRate>
  <CurrentInterestRate>
    <RateType TYPE="Static">
      <StaticRate>.05</StaticRate>
      <RateAsPercent>No</RateAsPercent>
    </RateType>
  </CurrentInterestRate>
</InterestRateCalculation>
```

You are here: Configuration > Valuation > Valuation Business Rules > FundListForAllocation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

FundListForAllocation Rule

The FundListForAllocation business rule allows for the specification of funds that will be available to a specific plan for allocation.

FundListForAllocation should be overridden at the plan level. In its most basic implementation, FundListForAllocation can be configured to make all of a plan's funds available.

XML Example

```
<FundListForAllocation>  
  <FundHierarchy>Parent</FundHierarchy>  
</FundListForAllocation>
```

Refer to the **XML Configuration Guide** under **Business Rules | Attached Rules | FundListForAllocation** for more information.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Configuration > Valuation > Valuation Business Rules > UnitValuePrecision Rule

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

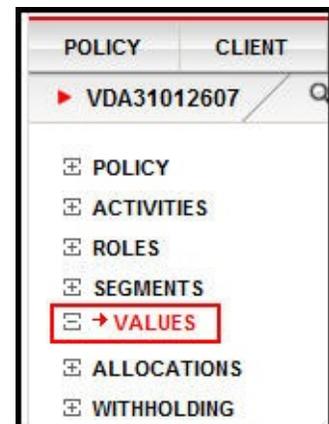
PrecisionValues Rule

The PrecisionValues business rule provides rounding precision convention (i.e. number of digits allowed after the decimal point) for the display of Unit Values and for the calculation and display of Number of Units.

To use the same unit precision across multiple plans, configure the PrecisionValues business rule at the global rule.

This rule controls the value displays in several locations in OIPA:

- **OIPA Policy screen Values link:** This link is in the left navigation menu of the Policy screen. It opens to reveal the Fund Details on the Value screen.



Fund values are displayed in Units and in Unit Values.

Values Link in Policy Left Navigation Menu

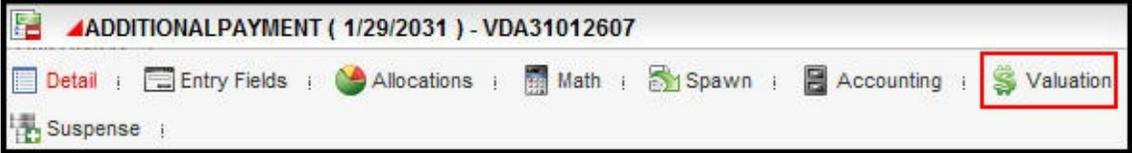
- **OIPA Activity Result Window Allocation link:** The Activity Results window may be accessed from the Activity Screen (via the Activity Detail Icon), or from the Activity Detail window (via the Quote button, when configured).



Allocation Link in Activity Result Window

- **OIPA Activity Result Window Valuation link:** The Activity Results

window may be accessed from the Activity screen (via the Activity Detail Icon), or from the Activity Detail window (via the Quote button, when configured). The Valuation link at the top of the window opens the fund valuation information in Units and Unit Values.



Valuation Link in Activity Result Window

XML Sample of Precision Values Rule

```
<PrecisionValues>  
  <PrecisionValue TYPE="UnitValue">  
    <Precision TYPE="Display" METHOD="Truncate">2</Precision>  
  </PrecisionValue>  
  <PrecisionValue TYPE="Units">  
    <Precision TYPE="Calculation">6</Precision>  
    <Precision TYPE="Display">4</Precision>  
  </PrecisionValue>  
</PrecisionValues>
```

PrecisionValues XML for Displaying Units and UnitValues in the Rules Palette

Policy Name: UL-ck-2010	Plan Date: 1/29/2010
Policy Number: UL31024776	Company Name: International Child Company
Policy Status: Active	Plan Name: Unit Linked Template

▲ VALUATION DATE

Valuation Date: Valuation Date: 

▲ POLICY VALUES

Issue Date:

▲ FUND DETAILS

Fund Name	Cash Value	Units	Unit Value
Fund A	242,751.61 THB	3,527.3416	1.90

Precision Demonstrated in OIPA in Values Link on Policy Screen

Steps to Configure PrecisionValues Business Rule Override

1. Navigate to the Global Rules Explorer.
2. Expand the Business Rule folder.
3. Expand the System folder.
4. Right-click on PrecisionValues.
5. Select **New PrecisionValues override**. The Override Wizard opens.
6. In step 1, select **Next**.
7. In step 2, select the level of unit precision to apply.
8. Select **Finish**.
9. Click the XML Source pane and configure the rule. This rule must be configured manually in the XML Source pane.



Refer to PrecisionValues in the **XML Configuration Guide** for a complete explanation of XML elements and attributes.

You are here: Configuration > Valuation > Scheduled Valuation > Scheduled Valuation Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Scheduled Valuation Overview

Scheduled valuation calculates the value of a group of policies at a specific time and stores that value for subsequent use. The time interval for running the scheduled valuation and the frequency at which policies valued can be selected by the client. Typically, these intervals are quarterly, semi-annually or annually.

Determine Timing for Scheduled Valuation

Scheduled/daily valuation is invoked by a rule called ScheduledValuation, which is attached to Plan-Financial type transactions. Within this rule is a Query tag that contains a SQL query that will determine what policies in the current plan should have valuations performed. The PolicyValues tag will specify whether or not the PolicyValues rule should be processed during the valuations. By default, PolicyValues will be processed according to the plan's setup. This tag provides a way to override the plan's current setting for whether or not PolicyValues should run. The WriteDeposits tag will contain a math variable that resolves to either Yes or No. By default the value will be Yes. This tag controls whether or not the scheduled valuation code should write deposit records to the AsScheduledValuationDeposit table. The math variable will be resolved against the transaction and the policy values rule. The Deletes tag will continue to support recovery processing in the event valuation must be re-processed.

The transaction type for Scheduled valuation is Plan-Financial.

[View scheduled valuation prototype.](#)

You are here: Configuration > Valuation > Scheduled Computation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Scheduled Computation

Scheduled Computation is run at the plan level and provides the ability to calculate values on policies or segments that do not have a variable component. For policies with a variable component, use [scheduled valuation](#) to calculate values.

A plan level transaction is typically used to execute scheduled computation. The ScheduledValuation rule is attached to the transaction. A specific selection of policies or segments can be identified through the use of a SQL query in the rule. Make sure that the planGUID is hard coded in the query. Once the policies or segments have been selected, a computation rule is called to perform various calculations such as cash value, surrender charge, and anything else required using standard math syntax. The output values that are calculated are stored in the database in order to be accessed for reporting or to be picked up by downstream systems.

[View Scheduled Computation prototype example...](#)

Configure Scheduled Computation

The ScheduledValuation business rule is used to perform either scheduled valuation or scheduled computation. The rule configuration drives the type of action that is performed. This rule is attached to a plan level transaction. An explanation of the configuration needed to perform scheduled computation is provided below.

High Level Steps to Configure

1. A plan level transaction must be created to initiate the process.
2. A ScheduledValuation rule must be configured to perform computation. This is an attached business rule that will be added to the plan level transaction.
3. A Computation rule must be configured to perform the calculations. This rule is identified in the <Computation> element inside the ScheduledValuation rule.
4. Use the System Properties to define the number of requests that can be simultaneously processed by the system. The scheduledValuation.batchSize and scheduledValuation.period properties control this. An explanation of all OIPA System Properties can be found on the **Oracle Technology Network (OTN)** in the documentation library that supports this release.

Plan Level Transaction

A prototype transaction was created to demonstrate the configuration needed for scheduled computation. Navigate to **Global Explorer | Transactions | ScheduledComputation Policy OR ScheduledComputationSegment**. The following configuration requirements must exist in the transaction.

- Asynchronous configuration must be present in the transaction.
`<Asynchronous CHECKINTERVAL="30" TIMEOUT="1000" CONTINU`
- A math section must define the math variables that are used in the Mapping section of the Computation rule.

ScheduledValuation Rule

A prototype rule was created to demonstrate the configuration of this rule. Navigate to **Global Explorer | Business Rules | Attached Rules | ScheduledValuation | Transaction Overrides | Scheduled Valuation (ScheduledComputationPolicy-Functional Prototype Plan)**.

- The <Query> element is used to add a SQL query. It identifies the segments or policies that should be included in the scheduled computation. It will return the PolicyGUID(s) or SegmentGUID(s) that the calculations are run against.



A best practice would be to create a validation count of the number of policies or segments the rule is expected to process within the activity math that could be used to compare to the actual number of requests created.

- The <Computation> element must be included in this rule. This tells the rule to perform computation instead of valuation and identifies if it is policy or segment level. The RULENAME attribute within this element identifies the Computation rule that will be called to perform calculations.

Computation Rule

- This rule is a plan activity context. It can exist at global, activity or plan level. It is a system rule and can be found in the System folder in the Global Explorer.
- This rule contains an Input section that defines a MathVariable section containing standard math variable syntax and operations. The ActivityFunction is not supported at this time.
 - The context defined in the ScheduledValuation rule's <Query> element will carry over to this rule.
 - Policies can execute loops of their segment records and segments will have access to their related policy records.

- In lieu of segment loops, the <Query> element can be used to specify the SegmentGUIDs that the Computation rule is executed against.
- Copybooks and Functions with contextual overrides are supported within the Math syntax.
- This rule contains an Output section that defines a Mappings section containing named Computation fields and their data types and maps each to a MathVariable from the Input section.
- The Mapped fields are created on the AsScheduledComputationField table and associated to the record on AsScheduledComputation.

Database Tables

Three tables were created to support scheduled computation:

1. AsScheduledComputation: holds records created from the ScheduledValuation rule when Policy or Segment is the value of the <Computation> element and a computation rule output relates to a PolicyGUID or SegmentGUID.
2. AsScheduledComputationField: holds output from Computation rule.
3. AsComputationRequest: holds a record for each time a Plan level activity executes with an attached ScheduledValuation rule that is configured with a valid Computation element and computation rule.

You are here: Configuration > Valuation > Point in Time Valuation > Point In Time Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Point-In-Time Overview

Point-in-Time valuation allows valuation calculations to be made using values from the last valuation date, rather than from inception. When Point-in-Time valuation is not enabled, Traditional valuation is used. In this method policy valuation calculations are made using all data from the inception of the policy. Traditional valuation is much more resource intensive than Point-in-Time, since each valuation activity must be recalculated every time valuation is executed.



When using the Debugger, the deposit effective date is the date when the depositing activity applied money to the fixed fund.

Converting from Traditional to Point-in-Time Valuation

Plans that are configured to use Traditional valuation may be converted to use Point-in-Time valuation, provided that the fund types are supported. At present, Point-in-Time valuation supports all fund types except Unit Linked Variable funds.

A special utility is needed to perform the conversion. To request this utility, go to My Oracle Support at <https://support.oracle.com> and file a service request. A support specialist will facilitate the process.



Prior to configuring Point-in-Time valuation, ensure that basic fund valuation is already fully working in the environment. Funds, allocations and assignments must already be set up and configured before attempting to enable Point-in-Time valuation.

Once a plan is configured to use Point-in-Time valuation, it cannot be transitioned back to Traditional valuation.

Point-in-Time Valuation Configuration

A [plan is configured to support Point-in-Time valuation](#) when the following two steps are performed.

1. Valuation must be defined at the plan level. This is done when a new plan is created using the [New Plan editor](#). Existing plans can be updated using the [Plan Maintenance editor](#). The Point-in-Time valuation field on both editors must be set to Yes.
2. Two business rules must be configured.
 - [PointinTimeValuation](#): This is a system rule that indicates how often valuation records are written to the database and displays a beginning valuation area on the Activity Results screen.
 - [WriteValuationElements](#): This business rule limits the amount of valuation information that is written. Existing valuation fields can be written to the system and are divided into sections: Policy, Fund, Deposit, and PolicyValues. Inside of these sections the available elements are from a set list. The only exception is the PolicyValues section, which can be configured to access any variables from the PolicyValues business rule. CopyBooks may be used in this rule. The rule can be overridden by plan.

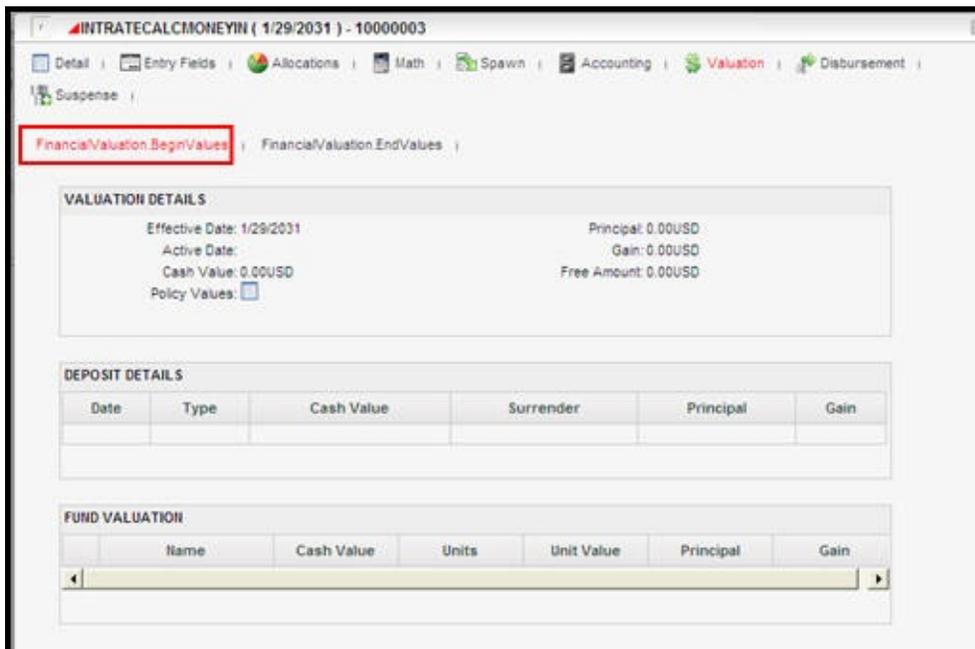
You are here: Configuration > Valuation > Point in Time Valuation > Activity Detail Screen

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Activity Detail Screen

When Point-in-Time valuation is enabled, the Activity Detail screen includes two additional views in the Valuation pane: BeginValues and EndValues.

BeginValues displays the valuation data prior to valuation assignment processing (values available when the activity executed).



Activity Detail Screen Valuation Tab with Begin Values Highlighted

The EndValues displays the resulting valuation data after assignment has been executed.

You are here: New Business Underwriting > New Business Underwriting Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

New Business Underwriting Overview

The Rules Palette contains a specific set of components that configure OIPA's New Business Underwriting (NBU) functionality, which is used to process applications for policies. This page contains a high-level overview of these components, along with links to more detailed information.

Enabling NBU Functionality

NBU functionality is only available for configuration if the NBU system is enabled for the environment. Enabling the NBU system requires administrative access to OIPA's Web Application Utility, as well as the inclusion of several special properties in the PAS.properties configuration file. Its availability is therefore at the discretion of the build manager.

For instructions on enabling NBU in the Web Application Utility, see the [Build Manager Environment Creation](#) page in this help system. For a more comprehensive explanation of environment creation, see the Rules Palette Set Up Instructions and the Rules Palette Environment Creation Instructions located in the OIPA Documentation Library on the Oracle Technology Network. For instructions on adding the necessary NBU properties to PAS.properties, see the System Properties document located in the OIPA Documentation Library on the Oracle Technology Network.

NBU Process Flow

Although not all stages of the NBU process require configuration, the general order by which records in the NBU system will be accessed and processed requires the following components used in the following order:

[Case](#) > [Application](#) > [Requirements](#) > [Impairments](#)

NBU Components

This section contains a list of the components used to configure NBU functionality.

Requirement-Related Components

A significant component of the NBU process revolves around the use of requirements, which are used to define conditions under which an application for a policy will be accepted.

Requirement Rule Context

If NBU is enabled, certain attached rules are able to be attached to requirements. A list of these rules is in the Attached Rules (For Requirements) section below.

Requirement Business Rules

In addition to the rules able to be attached to requirements, NBU requirements can be configured with a number of other business rules in order to invoke certain behavior. See the lists of rules below for names of the various types of business rules able to be used in conjunction with NBU requirements.

For more information on NBU requirements, see the [Requirements in New Business Underwriting](#) page. For more information on creating and configuring requirements in general, see the **Requirements** folder in this help system (**Admin Explorer | Requirements**). For a detailed explanation of requirement configuration syntax, see the **Requirements** folder in the XML Configuration Guide.

Business Rules

The following business rules are used to configure the display and behavior of the various stages of the NBU process. See the [Overview of Screen Rules](#) page for more information on screen rules and the

[Overview of Attached Rules](#) page for more information on attached rules. For a detailed explanation of the configuration syntax for the rules below, see the following folders in the XML Configuration Guide:

- Plan Rules: **Configuration Overview | Business Rules | Plan Rules**
- Screen Rules: **Configuration Overview | Business Rules | Screen Rules**
- Attached Rules: **Configuration Overview | Business Rules | Attached Rules**

Plan Rules

- [EligibleRequirementsByPolicyStatus](#)

Screen Rules

- CaseScreen
- CaseSearchScreen
- PolicyRequirementScreen
- RequirementResultSearchScreen

Attached Rules (For Transactions)

- AddRequirements
- AddImpairments
- ProcessRequirements

Attached Rules (For Requirements)

- AddRequirements
- StatusChange
- MatchRequirementResult
- CopyToRequirementFields
- CopyToPolicyFields
- CopytoClientFields
- SpawnActivities
- ProcessActivities

Override Levels

If NBU is enabled, new business rule overrides will need to have a

"System" override level specified. The options for this override are "NBU" and "OIPA."

For more information on creating business rule overrides, see the [Create a New Rule or Rule Override](#) page in this help system.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: New Business Underwriting > Case Screens

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Case Screens

The Case screens in OIPA are used to process Case records as part of the New Business Underwriting (NBU) process. The Case Search screen is used to search for Case records, while the Case screen is used to create and edit Case records. These screens are configured by the CaseSearchScreen and CaseScreen business rules, respectively.

Navigate in the **Global Rules Explorer to Business Rules > Screen > CaseSearchScreen** and **Global Rules Explorer to Business Rules > Screen > CaseScreen** to view these business rules in the Rules Palette.

Case Search Screen

The Case Search screen uses an assortment of fixed fields, including Case Name, Case Number, Last Name, First Name and Tax ID to aid in identifying Case records, and can have additional search fields configured, as well.

The Case Search screen has two sections:

- The Case Search Criteria section displays the screen's fixed fields.
- The Case Search Results section displays any Case records returned from a search.

Case Screen

The Case screen uses a number of fixed fields, including Case Name, Case Number, Case Status, Date Created and Date Last Updated, all of which correspond to columns within the AsCase database table.

Additionally, dynamic fields can be configured to display other information on the screen. A case's status is represented by a two digit role code, and these role codes are defined in the AsCodeCaseStatus code name.

Masking and field-level security is supported on the Case screen.

The Case screen has three sections:

- The Case General Info section displays the screen's fixed fields, such as case name and number.
- The Policy/Application Table section displays the table defined in the <Table> tags within the <Policies> tags of the CaseScreen rule.
- The Case Detail section displays the dynamic, configurable fields defined in the CaseScreen rule's configuration. These fields' values are stored in the AsCaseField database table.

You are here: New Business Underwriting > Creating the Application Screen

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Creating the Application Screen

OIPA's Application screen holds all of the information contained within a given case's application. It is configured using a system code-level override of the PolicyScreen business rule. Overriding PolicyScreen at the system code level will populate a record in the SystemCode column of the AsBusinessRules database table. This record is then used by the system to determine which PolicyScreen rule to use as the Application screen in OIPA.

See the steps below for the process needed to create an Application screen for a policy.

Overriding PolicyScreen at the System Level

1. Navigate to the Global Rules Explorer.
2. Select **Business Rules | Plan Rules**.
3. Right-click on the **PolicyScreen** node and select **New PolicyScreen Override**.
4. The New PolicyScreen Override wizard will open. Click **Next**.
5. Select a company from the Company drop-down.
6. Select **NBU** from the System drop-down
7. Make any other necessary selections to further define the override's context.
8. Click **Finish**.

You are here: New Business Underwriting > Requirements in New Business Underwriting

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Requirements in New Business Underwriting

The New Business Underwriting process may require several different types of requirements to be fulfilled before an applicant is able to open a policy with the company. These requirements could verify medical or financial factors surrounding the client, as well as the client's age, the product line of the policy for which they are applying, or the policy's face amount, among other possible requirements. In other scenarios, requirements may be dependent on the fulfillment of a predecessor requirement, or can be triggered by an amendment or Home Office Endorsement in the event that information on an application changes.

Because of the significant role that requirements play in the New Business Underwriting process, a number of business rules and other configuration tools exist in the Rules Palette to tailor the processing of requirements to specific business needs. This page outlines the configuration available for New Business Underwriting requirements.

See the Requirements folder in this help system for more information on creating and configuring requirements in general. Navigate to **Admin Explorer | Requirements**.

Configuration Foundation

Requirement configuration consists of three major building blocks: Requirement Detail configuration, Requirement Result configuration and Requirement Definition configuration. In OIPA, these building blocks control the Requirement user interface, the Requirement Result screen and the processing of the requirement, respectively. In the Rules Palette, each of these areas of configuration has its own configuration tab in the Requirement Editor.

The configuration for each of the building blocks is stored in the AsRequirementDefinition database table. A record must exist in this table in order for a requirement to exist, although requirements can have NULL values in the XMLData, XMLResult and/or XMLDefinition columns. Requirements may also be configured partially or entirely through CopyBooks.

Requirement Detail

Requirement Detail configuration controls OIPA's requirement user interface, including fields, events, actions and ScreenMath. This component is configured in the XML Definition tab of the requirement editor, and the configuration is stored in the XMLData column of AsRequirementDefinition.

For detailed information on configuring Requirement Details, navigate to **Requirements | Requirement Detail** in the XML Configuration Guide.

Requirement Result

Requirement result configuration controls the information that displays on the Requirement Result screen. Requirement results can be formatted in three ways:

- **Fields**—Individual fields associated with an individual requirement result. These fields may be obtained from AsRequirementResult and

AsRequirementResultField.

- **Text**—A complete text document, which is stored in the ResultText column of AsRequirementResult.
- **Table**—A table that contains data obtained from AsRequirementResultTypeField,

Requirement results may arrive electronically or in paper form, but regardless of the medium, a result must have a record in AsRequirementResult in order to be visible in the system. Depending on the format of a result (fields, text or table), additional records may be created in AsRequirementResultField, AsRequirementResultType and/or AsRequirementResultTypeField. Records do not need to be present in these additional tables for a result to be visible in OIPA.

If a result arrives electronically, its record will be inserted into AsRequirementResult via AsFile.

If the RequirementGUID of the ordering requirement is known when a result record is created, then the record is inserted into the AsMatchedRequirementResult database table. If the RequirementGUID is not known, then the result record is inserted into the AsUnmatchedRequirementResult database table.

Unmatched requirement results can be matched with requirements on the Activity Requirement screen in OIPA. To do so, a user should search for a requirement and click on the magnifying glass icon in the desired requirement's Action column, which will open the Requirement Result Search screen. The user can then search for a requirement result and, if the desired result is found, click on the page icon to match it to a requirement. The Requirement Result Search screen can be configured using the RequirementResultSearchScreen business rule.

Requirement Definition

Requirement definition configuration controls the processing of a single

requirement. Requirement definitions are configured in the XML Source pane of the requirement editor, and the configuration is stored in the XMLDefinition column of AsRequirementDefinition.

For detailed information on the configuration of requirement definitions, navigate to **Requirements | Requirement Definition** in the XML Guide.

Supporting Configuration

Depending on the exact method by which a requirement is intended to process, some additional configuration may be necessary. An outline of additional configuration is provided below. Note that the NBU system will need to be enabled for the following configuration to be available.

- The Application screen may need to be configured to allow requirements for a particular role on a policy.
- Requirements whose availability should be dependent on the policy's status should be defined in the EligibleRequirementsByPolicyStatus business rule.
- If the addition or processing of requirements needs to be triggered by other events in the system, additional business rules may need to be attached to a transaction or requirement:
 - **AddRequirements:** This business rule is attached to a transaction or requirement in order to add requirements based on the configured requirement criteria.
 - **ProcessRequirements:** This business rule is attached to a transaction in order trigger the processing of requirements.
 - **CopyToPolicyFields:** This business rule allows one or more MathVariables to be copied from an activity to one or more policy fields.
 - **CopyToClientFields:** This business rule allows one or more MathVariables to be copied from an activity to one or more client fields when the activity to which this rule is attached is processed.
 - **CopyToRequirementFields:** This business rule is attached to a transaction to allow the results of an activity's math calculations to write out to a defined group of RequirementGUIDs.
 - **MatchRequirementResult:** This business rule matches a specified requirement result with the current requirement.
 - **ProcessActivities:** This business rule is attached to a requirement to set conditions for the automatic processing of activities.
 - **SpawnActivities:** This business rule is attached to a requirement or transaction to set conditions for the spawning of activities.
 - **StatusChange:** This business rule can use requirements as criteria for updating an application's status.

- The UnmatchedResultSearchScreen can be used to configure the Unmatched Requirement Result Search screen. While specific requirement results can be matched to specific requirements using the MatchRequirementResult attached rule, this screen allows the user to perform a search for unmatched requirement results across all available policies, and then attach the results to requirements.

You are here: New Business Underwriting > Impairments

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Impairments

The Assessment screen and Assessment Detail screen are used to display impairment data in OIPA, with the Assessment screen providing a summary of existing impairments and the Assessment Detail screen displaying the particulars of a single impairment record. Both of these screens are code-generated, meaning that the only configuration needed in the Rules Palette is the setup of the screen's security.

Impairment data is typically added to the system along with an application, via AsFile. However, Impairment data can also be added to an application manually by clicking the **Add Impairment** link on the Assessment screen.

When viewing an application in OIPA, the Impairment screen can be accessed by clicking on **Assessments** in the menu on the left side of the screen. The Assessment Detail screen can be accessed by clicking **Add Impairment** on the Assessment screen, or by clicking on the icon in the Action column of the impairment summary table on the Assessment screen.

Shadowing Impairments

The Assessment screen supports the ability to manually shadow impairments by clicking on the trash can icon in the "Action" column of the Impairments table. The trash can icon will only be available if the user has the proper [security settings](#) enabled. Shadowing an impairment in OIPA will update the impairment record in AsImpairment by changing the value in the StatusCode column from 01 ("Active") to 12 ("Shadowed"). These code values are set by the AsCodeImpairmentStatus [code name](#).

The Assessment screen also allows users to view previously shadowed Impairments by selecting the "Show Shadows" checkbox above the Impairments table. The user will need to have the proper [security](#) enabled in order to use this checkbox.

You are here: Release Management > Release Management Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Release Management Overview

The Release Management process allows users to control the flow of database information from a source environment to a target environment. It establishes a repeatable release process and provides structure to the configuration experience. Using Release Management also creates an audit trail for both past and future package releases. Changes to rules during this process can be tracked, eliminating the need for manual spreadsheets.

[Configuration Packages](#) are created to bundle rules. The use of a specific, defined naming system will help identify the purpose and content of the configuration packages. Once all of the configuration packages are created, they are marked as Ready to Migrate and a Configuration Lead will manually review them to make sure there are no conflicts.

[Migration Sets](#) are the second phase of the process. Multiple configuration packages are added to a single migration set. Once a migration set has all the necessary configuration packages, it is added to a release package. A release package can only contain one migration set.

[Release Packages](#) are the third phase of the process. A single migration set can be added to a release package. Stored procedures, DML scripts and DDL scripts and other non-Palette items can also be added at this time. A release manifest file is generated in the release package file directory to identify all the components in the package.

Detached Migration

When a source environment doesn't share an IVS environment with the target environment, a detached migration must be performed. The detached migration creates a Release Package that contains the XML data for each of the rules being migrated. This release package can be sent via FTP, email, etc., to any location and deployed to a 'detached' IVS environment.

Refer to the [Detached Migration](#) section for an explanation of this process.

Enable Release Management Functionality

Release Management is only available to a user when the Release Management functionality is turned on. The build manager determines if Release Management is needed. A user can check to see if this is turned on in a particular environment by right-clicking on the environment node, when logged in, and selecting the **Properties** option. Then scroll down and check the Release Management property. Only the Main Explorer or the Global Rules Explorer can be used to check the environment properties.

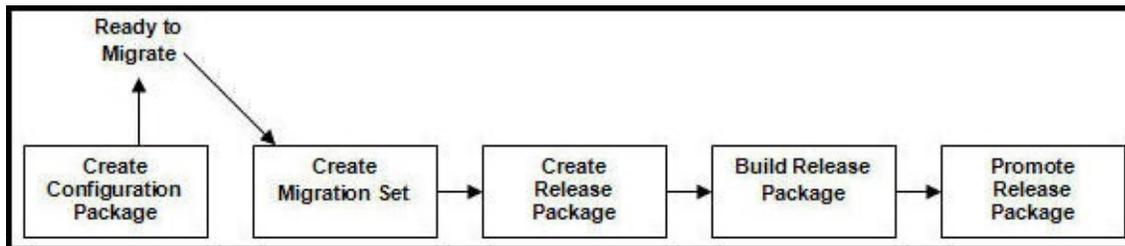
Once the Release Management functionality is on, a user must also have the proper security privileges.

Security

[Release Management security](#) is divided into three stages. A user typically has security privileges for one stage of the process, but this is not a limitation. The user can have any combination of privileges, according to what is specified in the [security role](#).

Process Overview

There are several steps that must be completed to move through the release process. First, [create a configuration package](#). Once it is complete, mark it ready to migrate and then a [migration set will be created](#). A [release package is then created](#). Next, [build the package](#) and [promote the package](#). After all these steps are complete, a release manager will deploy and promote the package in the target environment.



Overview of Release Management Process



Before making any changes and migrations, make sure all environments in the development track are synchronized with respect to the rules configuration.

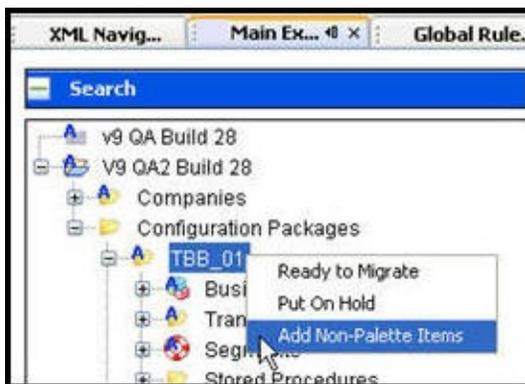
Create Configuration Package

After Release Management is set-up in the source environment, begin [creating configuration packages](#). A node for configuration packages will display under the environment name in the Main Explorer. If the node is not present, contact the Security Manager to make sure Release Management privileges were assigned to your role. Release Management will not be visible unless a user has the proper security privileges.

The configuration packages node on the Main Explorer holds the configuration packages that are created. Security controls whether users see their packages only or packages created by other users. When the release management process is initiated, this folder will be empty until the first package is created. Packages are removed from this folder once they have been labeled Ready to Migrate.

Configuration Package Right-Click Menu

- **Ready to Migrate:** select this option when the configuration package is complete. The rules and transactions associated with the package will unlock and be available for use.
- **Put on Hold:** select this option when the package is not ready to migrate but the rules and transactions associated with the package can be released so other users can access them.
- **Add Non-Palette Items:** select this option to attach stored procedures, DDL scripts, or SQL scripts to the configuration package.



Configuration Package Right-Click Options

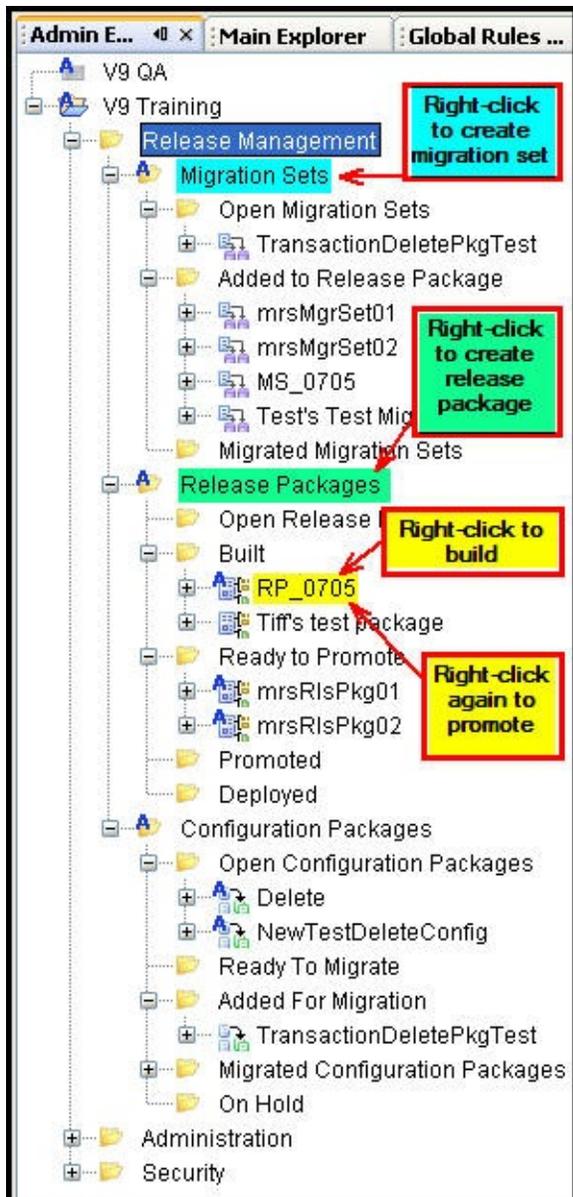
Create Migration Set

After a configuration package is marked Ready to Migrate, the package is moved to the Migration Sets folder in Release Management, which is located on the Admin Explorer tab. The Build Manager (or BA with administrative privileges) will [create a migration set](#) that includes the configuration package as well as any other packages that are relevant.

Release Management Folder Hierarchy

- **Migration Sets:** right-click to create a migration set. Select configuration packages to add to the migration set.
- **Release Packages:** right-click to create release package. Select one migration set for the release package.
- **Name of Release Package Created:** right-click on the name of the release package to build the package. Right-click again to promote the package. The right-click option for the stage that the release package is currently in will be visible. For example, if it has already been built, then the only right-click action available will be Promote.

Once the migration set is promoted, it will move from the Built folder to the Promote folder. The package will continue to move through the folder structure as it advances through the release process.



Release Management Hierarchy in Main Explorer

Create Release Package

After the migration set is created, [create a release package](#). The migration set will remain in the Migration Set folder.

Build Release Package

After the release package is created, the Build Manager will need to go back to the Release Management folder in Admin Explorer. From that folder the release package can be built by right-clicking the package and selecting **Build Release Package**. Once the package is built, the right-click menu option will change to Promote.

Promote Release Package

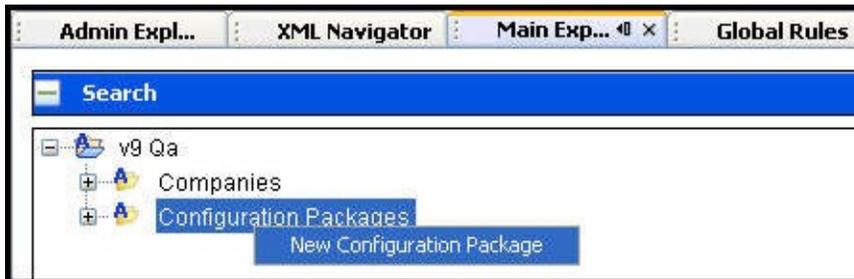
After the package is marked ready to promote, log-out of the source environment and log-in to the target environment. From the Admin Explorer, locate the Release Management node. Open the folder and look for Release Packages. The release package should be listed under the Release Package | Ready to Promote node. First, approve the package. The [Release Package-ApprovePromotion](#) security privileges must be granted to users who are responsible for approving release packages. Approved packages are moved to the Promoted folder. Open the Promoted folder and [deploy the package](#). After completing these three actions, the right-click menu for the release package will disappear and the package is moved to the Deployed folder. The migration is complete.

You are here: Release Management > Create Configuration Package

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create Configuration Package

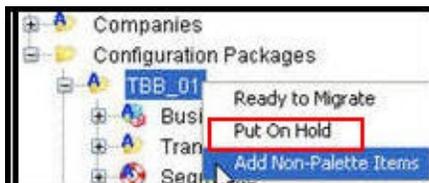
Configuration packages are created by checking-out rules, transactions or segments and assigning them to a configuration package. Additionally, they can be created from the Main Explorer when a user right-clicks on the Configuration Package node and selects **New Configuration Package**.



Configuration Package Right-Click Option

When an item is added to a Configuration Package, it is locked to all other users until the Configuration Package is pushed to Ready To Migrate or until the **Put on Hold** option is selected. If the **Put on Hold** option is selected, then ALL items in the Configuration Package are released.

⚠ Make sure to analyze rules that were Put On Hold after they are released and taken off hold, to ensure that any additional information added by another user doesn't conflict with the original intent of your rule configuration.



Put On Hold Option for Configuration Packages

Security privileges determine if all configuration packages display or only the ones created by the user. Configuration packages display under the

Configuration Package node. Each package contains folders for the items that can be included in the package. The items added to the configuration package will display in the respective folder. Business rules added to the package will display under the business rule folder. Stored Procedures added to the package will display under the Stored Procedures folder and so forth.

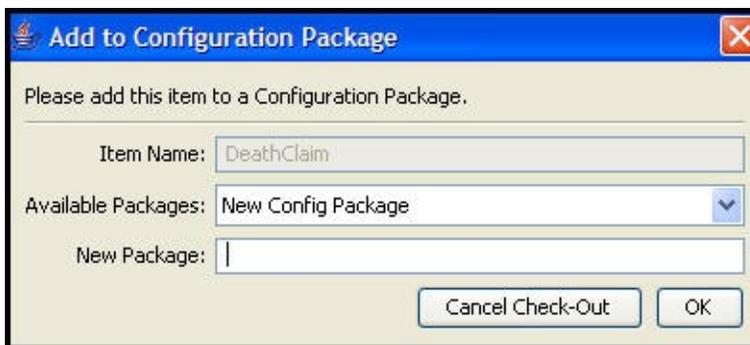


Configuration Package with Nodes for Each Included Item

After a configuration package is created, the items in the package can be [removed](#) or updated as needed. Select an item and right-click on the XML file to see the available options.

Steps to Create a Configuration Package

1. Check-out a business rule, transaction or segment. The **Add to Configuration** window will open.
2. Select a configuration package from the drop down list or select **New Configuration Package** and enter the new package name.
3. Click **OK**. If the item is already locked in another configuration package, then a Configuration Package Lock message will display. Contact the package owner or the system administrator to gain access.



Add to Configuration Window

4. Make any necessary updates to the item and save the changes.
5. Continue adding all necessary business rules, segments or transactions.
6. Open the **Main Explorer** tab.
7. Open the configuration packages folder.
8. [Right-click on the configuration package](#) and select **Add non-palette items**. If non-palette items are not needed in this package, then skip to step 10.
9. Enter the stored procedure information and click **Attach**.
10. Right-click on the configuration package and select **Ready to Migrate**. The configuration package will disappear from the configuration packages node because it has been moved up to the migration set level. Only users with security permissions to view migration sets will be able to view the package.

An empty configuration package can also be created from the Release Management node in the [Admin Explorer](#). Click the Admin Explorer tab and open the Release Management folder under Administration.

Next, [add the configuration package to a migration set](#).



A configuration package cannot be modified once it is migrated. If it contains something that needs to be deleted, the entire configuration package must be deleted and recreated.

Remove a Rule from a Configuration Package

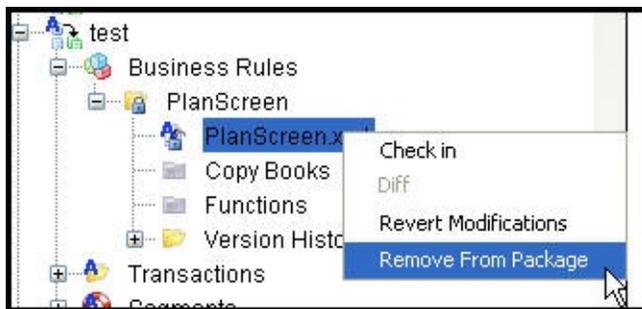
When a rule is removed from a Configuration Package, any ability to track or trace that rule is lost. A best practice to keep in mind when removing rules is to immediately add that rule to a new Configuration Package with a descriptive name like Orphan. This allows all removed rules to be further evaluated to ensure they are not needed in another Migration Set.

Steps to Remove an Item from the Package

1. Open the configuration package where the item resides from the Main Explorer or the Admin Explorer.
2. Open the folder that contains the item.
3. Open the specific folder that contains the XML file for the item.
4. Right-click on the XML file and select **Remove from Package**.
5. Select **Yes** when the confirmation message appears.
6. Add the rule that was removed to another Configuration Package immediately to prevent the rule from being lost. A best practice is to add it to a Configuration Package with a descriptive name like Orphan.



Items cannot be removed once the Configuration Package is migrated.



Configuration Package Right-Click Options

You are here: Release Management > Create Migration Set

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create Migration Set

Once a Configuration Package is marked as ready to migrate, it can be added to a Migration Set. Configuration Packages will no longer be visible under the Configuration Package node in the Main Explorer once they are ready to migrate and they cannot be updated or changed.



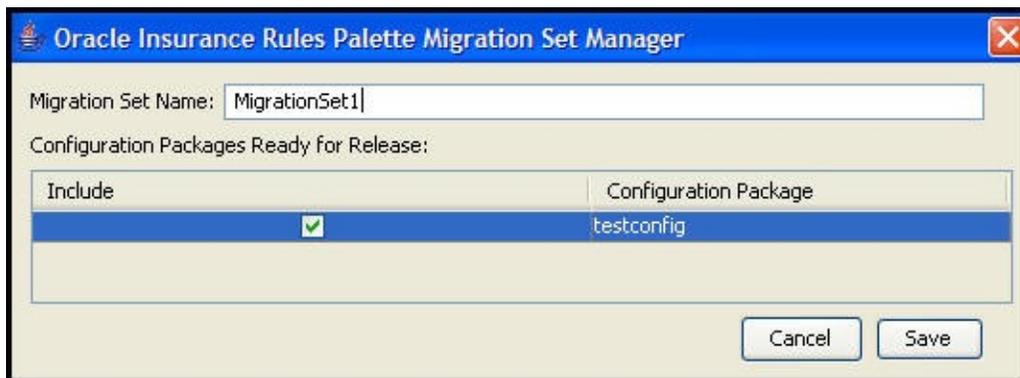
If the Release Management folder in Admin Explorer is not visible, then check the properties file. Make sure Release Management is on. If it is, then contact a supervisor to verify that the proper security privileges were granted to the role.

Typically a Configuration Lead will be responsible for Migration Sets. One important step in this process is the analysis of configuration. Each Migration Set should be reviewed to ensure the Configuration Packages do not contain conflicting configuration. This must be accomplished manually at this time.

Steps to Create a Migration Set

1. Open the Admin Explorer tab.
2. Open the Release Management folder.
3. Right-click on the Migration Sets node and select **New Migration Set**. The Migration Set Manager will open.
4. Enter a name for the migration set and check the configuration packages to include.
5. Select **Save**. The migration set will be listed under the Migration Set node in the Release Management folder.

Next, [create and promote the release package](#).



Migration Set Manager

You are here: Release Management > Create and Promote Release Package

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Create and Promote Release Package

After the migration set is created, the release package can be created and promoted.

Steps to Create a Release Package

1. Open the Admin Explorer tab.
2. Open the Release Management folder.
3. Right-click on the Release Package node and select **New Release Package**. The New Package Wizard will open.
4. Enter the following information, then select **Next**.
 - a. Select archive type. Can be either .zip or .jar
 - b. Enter package name.
 - c. Select the directory where the package will be stored. The Release Manifest is created here.
 - d. Enter Source Control Manager (SCM) label. This is optional.
 - e. Enter any comments. This is optional.



If this is a detached migration, then make sure to check the box at the bottom of the window indicating that files should be created for detached migration.

5. Check the migration set to include in the release package, then select **Next**.
6. Enter the location of the executable file, or check the option **Do not bundle the executable in this Release Package**. Select **Next**.
7. Enter the location of the configuration file, or check the option **Do not bundle the configuration file in this Release Package**. Select **Next**.
8. Check the stored procedures to include in the package and select **Next**.
9. Check the required DML scripts to include in the package and select **Next**.
10. Check the required DDL scripts to include in the package and select **Next**.
11. Select **Finish** to complete the release package.

Build and Promote the Release Package

1. Right-click the release package and select **Build Release Package**. When this is complete, right-click on the same release package and now the option says Ready to Promote.
2. Right-click the release package and select **Ready to Promote**.



You will now need to log out of the source environment and log into the target environment. Once in the target environment, you will be able to approve and [deploy the release package](#).

You are here: Release Management > Deploy Release Package

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Deploy Release Package

The final stage of the release management process happens in the target environment. If you have not already logged out of the source environment and into the target environment, you will need to do that now.



An environment is designated as a source or target in the Web Application Utility. The source environment has Release Management turned On and the Entry option set to True. The target environment has Release Management turned On and the Entry option set to False.

Before a release package can be deployed, it must first be approved. If a package is deployed to multiple environments, then it must be approved before each deployment.

[Security privileges](#) must be granted to anyone who has the responsibility of approving a release package before it is deployed.

Release Package Deployment Order

The Rules Palette enforces the deployment order of release packages in two ways:

- When a release package is being deployed to multiple environments, it will need to be deployed to each in a designated order. This order is specified by the [build manager during the environments' creation](#). An error message will display if a user attempts to deploy a release package in an incorrect order.
- When multiple release packages are being deployed from an environment, each must be deployed in a designated order. An error message will display informing the user that the release packages must be deployed in the proper order. This order enforcement also applies to [deployment rollbacks](#).

Steps to Approve and Deploy the Release Package

1. Log out of the source environment and log in to the target environment.
2. Navigate to the **Admin Explorer** window and open the **Release Management** folder.
3. Open the **Ready to Promote** folder.
4. Right-click on the release package and select **Approve Promotion**. The package will move to the **Promoted** folder. If the Approve Promotion option is not available, then security privileges have not been granted for the user.
5. Open the Promoted folder and right-click the package. Select **Deploy Package**.

Once the release package has been deployed in the target environment, the release package will move under the **Deployed** node of the Release Package folder in Admin Explorer. The release package will stay in **Ready to Promote** in any other environment(s) that share the IVS, until deployment of the package is done in the specific environment(s).

In the source environment, the release package will remain in the Ready to Promote folder.

Deployment Rollback

Release packages deployed to target environments can be "rolled back" if needed, reversing the deployment. This ability is useful if, for example, an error with the release package is found, or if the package is deployed to the incorrect environment. Release packages cannot be rolled back from source environments.

Just as the Rules Palette enforces the order in which release packages are deployed, it will enforce the order in which release packages are rolled back. Release packages must be rolled back in the reverse order that they were deployed. An error message will display if a user attempts to roll back a release package in an incorrect order.

Since rollbacks are not allowed in a source environment, detached migrations cannot be rolled back. However, a release package created from a detached migration can be rolled back if it has been deployed to other environments.

To rollback a release package:

1. In a target environment, navigate to the **Admin Explorer** window and open the **Release Management** folder.
2. Open the **Release Packages** folder.
3. Open the **Deployed** folder.
4. Right-click on the release package and select **Rollback**.
5. A message will display verifying that the release package should be rolled back. Select **Yes**. The release package will be removed from the Deployed folder and re-added to the **Promoted** folder.

You are here: Release Management > Delete Transaction Through Release Management

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Delete a Transaction Through Release Management

When a transaction is deleted during the release management process, the deleted transaction is added to a configuration package. All attached rules for that particular transaction are also deleted and added to the configuration package. Then, when the configuration package is promoted and deployed in the target environment, the transaction and all associated rules are removed from the target environment, synchronizing it with the source environment.

Steps to Delete a Transaction During Release Management

1. Expand the Company>Plan>Transaction folders until you locate the transaction to delete.
2. Right-click on the transaction and select **Delete**.
3. When the Question window appears, select **Yes**.
4. Give the Configuration Package a name.
5. Move the Configuration Package through the release management process and deploy it to the target environment.

You are here: Data Dictionary > Data Dictionary Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Data Dictionary Overview

The **Data Dictionary** is a tool that can be used to create a uniform set of fields and math variables for use during configuration. When a field or math variable is defined in the Data Dictionary, it can then be used in the configuration of a rule or transaction. The Data Dictionary also tracks the dependencies between variables and rules. This is done through the use of the **Used In** button in the Search tab.

All terms added to the Data Dictionary must be [validated](#). The use of the Data Dictionary can be [enforced](#) through the DataDictionaryEnforcement business rule.



Refer to the Data Dictionary [Best Practices](#) section before setting up and using the Data Dictionary.

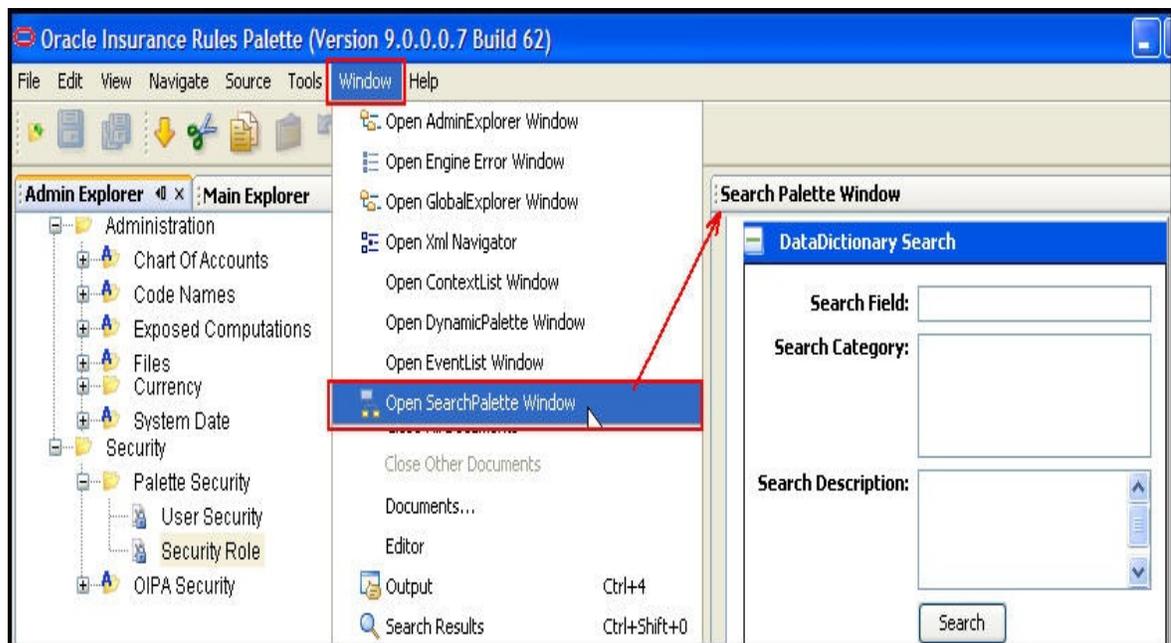
Data Dictionary Facts:

- Data Dictionary and its enforcement reside at the Global level.
- Data Dictionary saves configuration time by allowing users to drag and drop existing fields or math variables into the configuration.
- Data Dictionary prevents users from accidentally deleting active fields because they can see the rules and transactions that use it.
- Data Dictionary can be used as a dictionary of terms.
- Data Dictionary provides consistency during configuration by giving users access to a list of common names.
- The Rules Palette can warn end users that they are attempting to configure a field and/or math variable that is not defined in the Data Dictionary.

How to Open the Data Dictionary

There are two ways to access the Data Dictionary:

1. When configuring, access the Data Dictionary by clicking **Window** from the menu bar and selecting **Open SearchPalette Window**. Procedures for using the drag and drop functionality in the Configuration Area are described in the [SearchPalette Window](#) description.



Window Option on Tool Bar with OpenSearchPaletteWindow Option Selected

2. An additional option is to select the book icon found on the Tool Bar menu. The drag and drop functionality cannot be used from the Data Dictionary when this access option is chosen. Instead this option is used to manage and look-up variables and Data Dictionary categories.



Data Dictionary Icon on Tool Bar

[Security](#) is wrapped around the Data Dictionary, allowing only users with the proper permissions to create and edit dictionary terms. There are three levels of Data Dictionary security: check-in/check-out, manager and view.

How to Use the Data Dictionary

Open the Data Dictionary by clicking the Data Dictionary icon  on the Main Menu. There are three tabs in the Data Dictionary window: Search, Manage Term and Create New Category.

- **Search:** use this tab to search for existing terms or add new terms.
- **Manage Term:** use this tab to update existing terms or change the category a term is associated with.
- **Create New Category:** use this tab to add new categories to the Data Dictionary.



In the Data Dictionary the word *term* refers to either a field or math variable.



Data Dictionary Tabs

You are here: Data Dictionary > Data Dictionary Best Practices

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Data Dictionary Best Practices

Adhering to Data Dictionary best practices will help ensure the successful use of the tool. Before implementing the use of the Data Dictionary it is important to review the best practices that are listed below.

Best Practices:

1. When starting an implementation project:
 - Establish a team to define the categories, naming conventions and definitions for Data Dictionary terms.
 - Assign an individual as the Data Dictionary owner who will approve all terms added to the dictionary.
 - Turn on the Data Dictionary as soon as possible.
2. Use [security privileges](#) in the Rules Palette to control access to the Data Dictionary tool. Only configuration team members should have access to the Data Dictionary.
 - **DataDictionary-View:** this privilege restricts a user from adding new terms. The user is only able to select from existing terms defined in the Data Dictionary when configuring rules.
 - **DataDictionary-CheckIn/CheckOut:** user is able to add new terms to the Data Dictionary and can use the new term or existing terms in configuration. User is also able to add new categories to the Data Dictionary. Adding new terms means that the Rules Palette user will be able to name the term, define the term type, associate the term to one or more pre-defined categories and add a definition to the term.
3. Set the level of compliance for math variables in the [DataDictionaryEnforcement business rule](#) to **Ignore**.
4. Use standardized naming conventions for fields and math variables when defining Data Dictionary terms.
 - Minimize the use of abbreviations. Ex: NewEffectiveDate
NOT NewEffDate
 - Ensure the name is relevant to the data the field or variable is representing.
 - Do not use spaces between words in a name.
 - Do not use underscores or special characters in names.
 - Use an uppercase letter for each word in the name.
5. Use standardized naming conventions when defining categories for the Data Dictionary.
 - Minimize abbreviations when at all possible.

- Ensure the category name is meaningful to the configuration team as they will be using categories to filter the Data Dictionary terms.
- Spaces can be used in category names.
- Use an uppercase letter for each word in the category name.
- Special characters are supported in category names, such as dashes or underscores.

6. Manage the addition of new terms.

- Provide the DataDictionary-CheckIn/CheckOut privilege to a small number of configuration analysts. When adding a new term, use an agreed-upon prefix in the definition, such as 'FOR REVIEW-XXXXXXXX'.
- Provide a query to run against the Data Dictionary tables that will pull any entries that have the prefix to be reviewed by the Data Dictionary owner.
- If a new term is approved, then the prefix can be removed and will be available for the team to use.
- If the term is changed in any way, alert the team so any existing configuration can be changed to the approved name.

An alternate method for managing new terms is to use a spreadsheet rather than the online Data Dictionary in the Rules Palette. The spreadsheet should be filled out when a new term is deemed necessary and approved and should include all of the necessary data to populate the Data Dictionary tables. Scripts can be run to directly insert those terms into the database, making those newly added terms available in the Rules Palette for use by the configuration team.

You are here: Data Dictionary > Add or Edit Categories

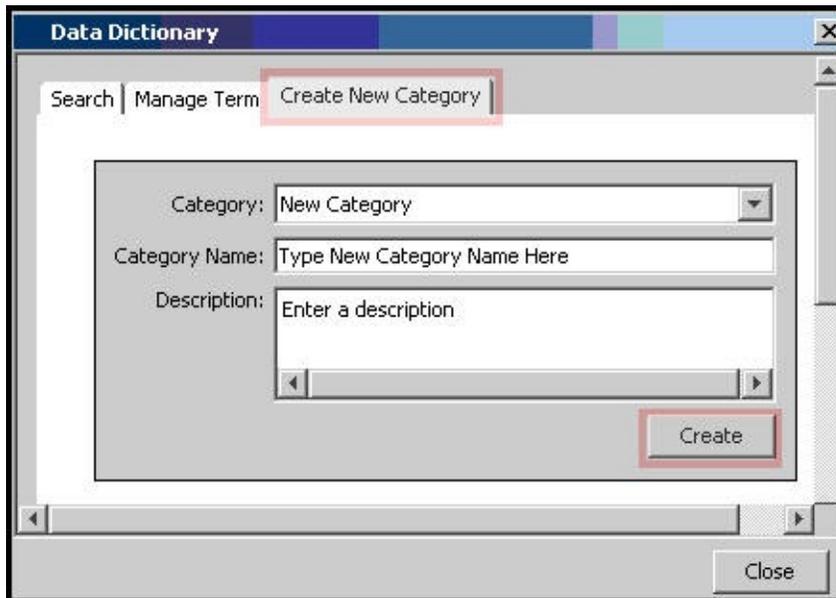
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Add or Edit Categories

The **Create New Category** tab is used to create and edit categories that can be associated with field and math variable terms. All terms must be associated with a category when they are first entered in the Data Dictionary.

Steps to Create a New Category

1. Open the Data Dictionary.
2. Click the **Create New Category** tab.
3. Select **New Category** from the Category drop-down box.
4. Type the name in the Category Name text box.
5. Type a description.
6. Select the **Create** button.



Create New Category Tab in Data Dictionary Editor

After a category is created, terms can be associated with that category through the Manage Term tab. If at any time a category needs to be updated, the changes can be made through the Create New Category tab. The category changes will be applied to any term associated with that category.

Steps to Edit a Category

1. Open the Data Dictionary.
2. Select the **Create New Category** tab.
3. Select the category from the Category drop-down box.
4. Edit the name or description.
5. Select the **Update** button.

You are here: Data Dictionary > Add or Edit Terms

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Add or Edit Terms

The Search tab is used to add new terms to the Data Dictionary. The Search tab has a New button, which when clicked will open the Manage Term tab. The Manage Term tab has a Create button that can be used to enter new terms. Once this button is clicked, it changes to Update so that any existing terms can be updated. Multiple terms cannot be entered one after the other on the Manage Term tab. You must go back to the Search tab and click New each time a new term is entered.



Make sure the [DataDictionaryEnforcement business rule](#) has the attribute Compliance="Yes" .

Steps to Create a New Field or Math Variable Term

1. Open the Data Dictionary by selecting the book icon from the Tool bar.
2. Select the **Search** tab.
3. Click **New**. The Manage Term tab will open.
4. Type the term name in the Term Name box (this will be the name that will be reflected in the configuration as the field or math variable name).
5. Select the data type from the **Term Type** drop-down box.
6. Highlight all categories that apply and select the right-facing arrow button. The categories will move over to the Selected Term Category box.
7. Type a description in the description box.
8. Select the **Create** button.
9. Add additional terms by selecting the **Search** tab and clicking **New**. Then complete steps 4 to 8.

Updating Terms Without Dependencies

A term can be edited through the Data Dictionary in the Manage Term tab. The term name can be updated as long as the term does not have dependencies. A term has dependencies when it is added to the field or math section of a transaction. All information relating to the term can be edited if the term is free of dependencies.

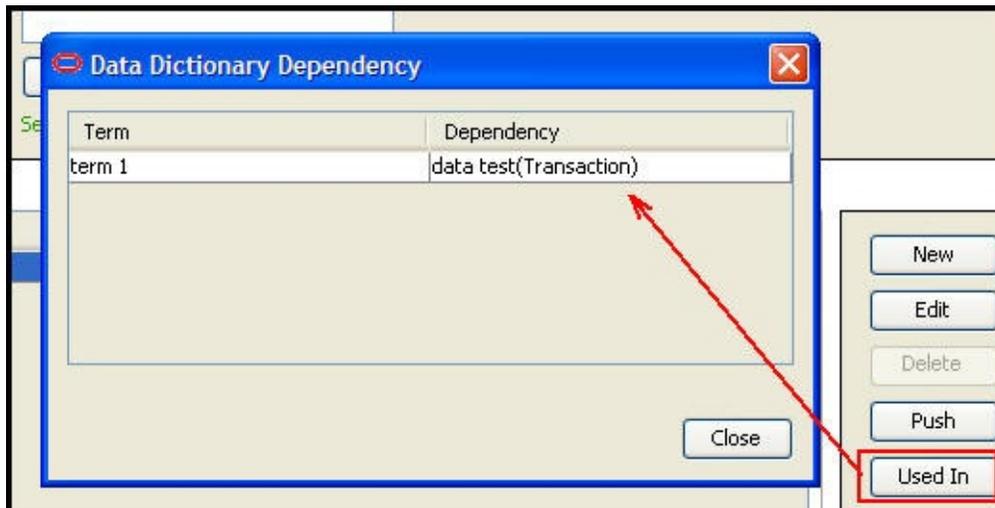
Steps to Update a Field or Math Variable Term

1. Search for the term.
2. Highlight the term in Return Results.
3. Select the **Edit** button, which toggles to the Manage Term tab.
4. Make necessary changes. You will not be able to change a term name if the term has dependencies.
5. Select the **Update** button.

Updating Terms with Dependancies

Term names can only be changed if the term is free of dependencies. A dependency exists if the term is used in the field or math section of a transaction. If dependencies exist, then the transaction should be updated so that the term is removed from the field or math section. Then the term can be updated and added back to the transaction if needed.

If a term name is greyed out and cannot be edited, but you aren't sure what dependencies exist for the term, then you can use the Used In button in the Search tab to find the dependencies. Search for the term, select the result from the search and click **Used In**. A window will open showing all dependencies for that term.



Used In Button in Data Dictionary Editor

Steps to Update Terms With Dependencies

1. Open the transaction where the term name has been added to the Field or Math section.
2. Remove the term from the transaction.
3. Check in the transaction to save the changes.
4. Open the Data Dictionary by selecting the book icon from the Tool bar.
5. Select the **Search** tab.
6. Type the name of the term that needs to be updated.
7. When the result is returned, select the term and click **Edit**. This will open the Manage Term tab.
8. Type the new name for the term. The term name should now be accessible since the dependency was removed.
9. Select the **Update** button.
10. Add the term back to the transaction if needed.

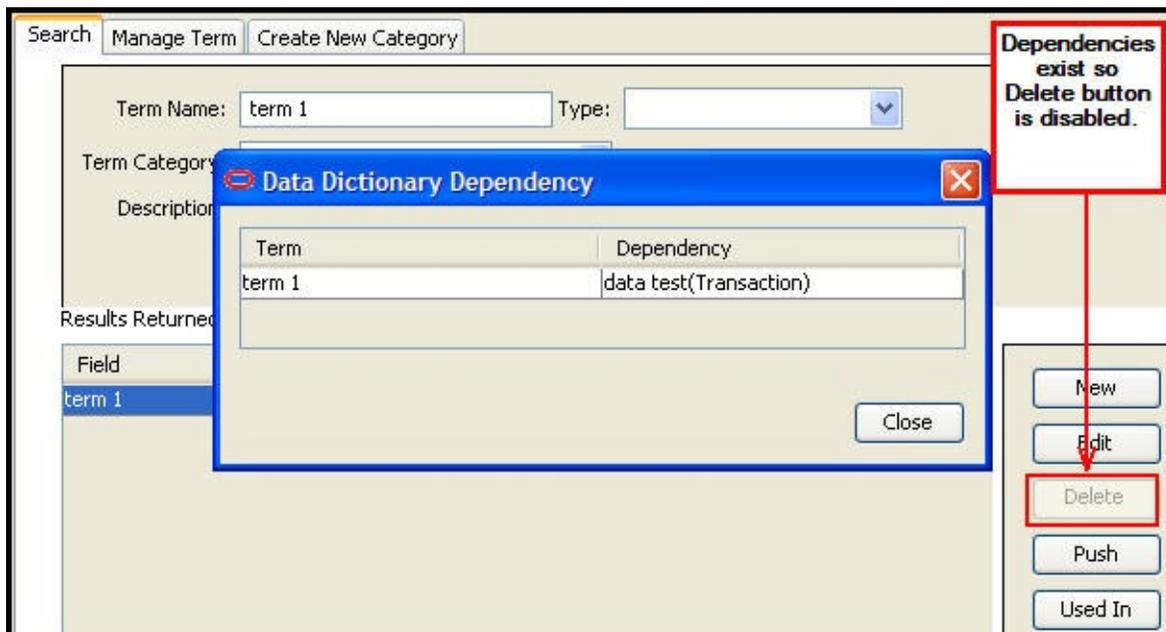
You are here: Data Dictionary > Delete Terms

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Delete Terms

Terms can be deleted from the Data Dictionary if no dependencies exist. The Delete button will be disabled if dependencies exist for a term. You can check to see what dependencies exist by searching for the term in the Search tab. Highlight the result and click **Used In** to see all dependencies.

Remove the term from the field or math section of the transaction shown in the dependency window. After the term is removed from the transaction, the Delete button will become enabled and the term can be deleted.



Delete Button in Data Dictionary Editor

You are here: Data Dictionary > Search the Data Dictionary

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Search the Data Dictionary

There are two ways to search for terms in the Data Dictionary.

- Click the Data Dictionary  icon in the Main Menu. The Data Dictionary window will open and you can follow the search steps listed below.
- Click **Tools** on the Main Menu and select **Data Dictionary Action**. This will also open the Data Dictionary window.

Steps to Search for Terms

1. Open the Data Dictionary using one of the methods described above.
2. Enter the search criteria and click **Search**. You can search using any method described below.
 - a. Enter the name of the field or math variable in the Term Name box and select the **Search** button. You can use the % sign as a wildcard.
 - b. Select a Term Category and select the **Search** button to see all terms that are in that category.
 - c. Select a Type and the **Search** button to search for terms of a particular type.
 - d. Enter keywords and the % wildcard to search through term descriptions.

Search results are displayed in the Results Returned section. Select the desired result and click one of the action buttons on the right. An explanation of each action button is provided below.

Buttons on the Search Tab

- **New:** Click New to open the Manage Term tab so that a new term can be created.
- **Edit:** Highlight a term from the search results and click Edit. The Manage Term tab will open so that the term can be edited.
- **Delete:** Highlight a term from the search results and click Delete. A warning message will ask for confirmation of the delete. Select **Yes** to delete the item or **No** to cancel the deletion.
- **Push:** Displays all the BusinessRules/Transactions/SegmentNames where the field/math variable is referenced. This will prevent the user from deleting any field if it is used in other places.
- **Used In:** Highlight a term from the search results and click Used In. This shows all dependencies. A dependency is a rule where the term is attached.

You are here: Data Dictionary > Data Dictionary Validation

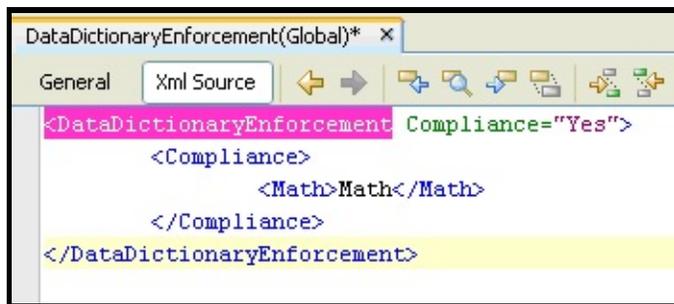
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Data Dictionary Validation

Data Dictionary Validation allows you to validate terms in the Data Dictionary and define a standard list of field names. You must validate any new term or field name that is added to the Rules Palette. In order to enable Data Dictionary Validation, you must first enable the DataDictionaryEnforcement business rule.

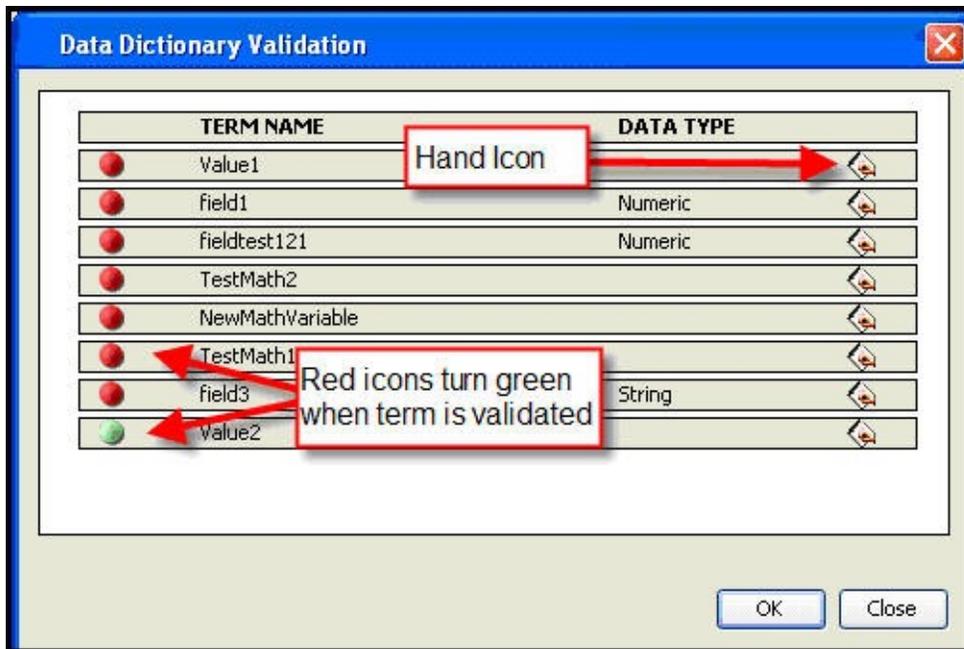
Steps to Enable DataDictionaryEnforcement Business Rule

1. Check-out the DataDictionaryEnforcement business rule in the Global Rules Explorer. The file is located in **Business Rules | System**.
2. Select the XML Source tab and modify the first line to read: `<DataDictionaryEnforcement Compliance="Yes">`.
3. Check the rule back in after making the modification.



DataDictionaryEnforcement XML

Enabling Data Dictionary enforcement also enables the **Data Dictionary Validation** pop-up window. This pop-up is invoked by selecting the **Save All**  button in the Tool bar after a field has been renamed in the [Field Properties](#) window of a transaction. It allows you to access the Manage Term tab, or to use an existing field/math variable name by changing the name. The **Data Dictionary Validation** pop-up displays all fields/terms that have been renamed in the **Term Name** column as well as the **data type** for each term in the Data Type column.

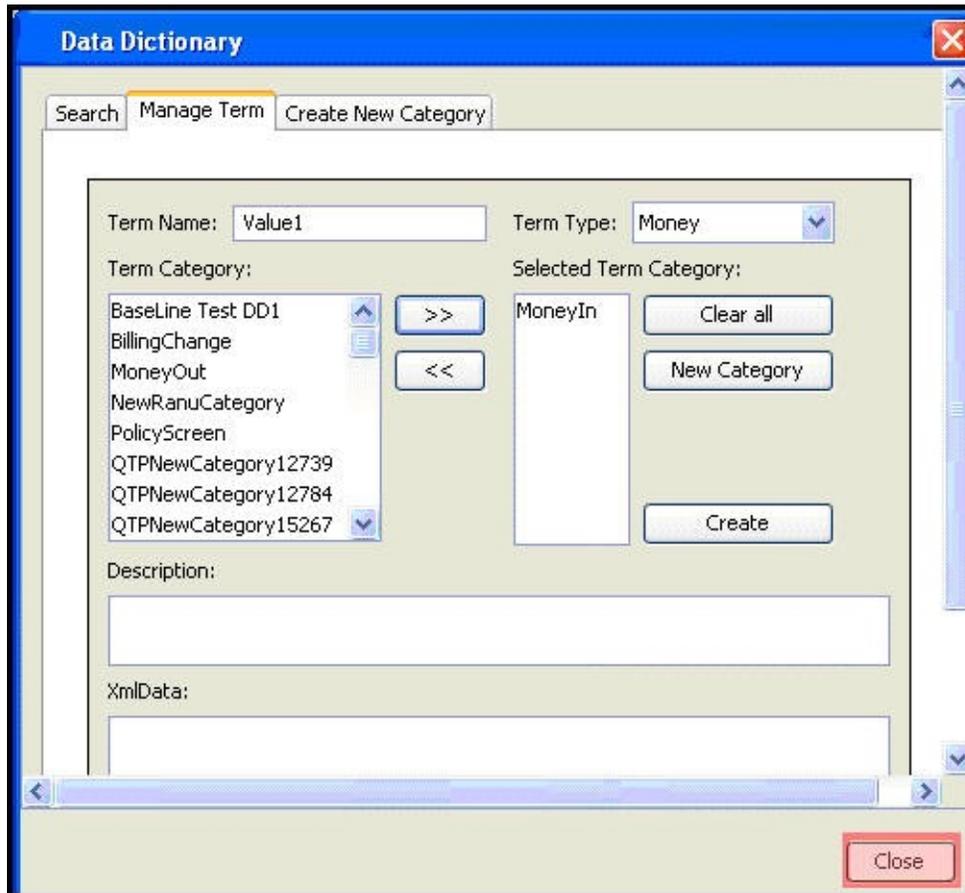


Data Dictionary Validation Pop-Up Window

A hand icon will display to the right of the Data Type column for each term listed. Clicking this icon invokes the Manage Term tab window. Each term also has a red or green circle icon located to the left of the term name. The icons are red when the term has not yet been validated in the system. The icon turns green once the term has been successfully validated.

Steps to Validate a Field/Term Name

1. Select the hand icon in the Data Dictionary Validation pop-up window. The Data Dictionary Manage Term tab window appears.



Data Dictionary Manage Term Tab

2. Add the term/field name in the Manage Term tab and select **Close**.
3. The new term/field name should now have a corresponding green icon in the Data Dictionary Validation pop-up window to indicate that it has been validated.
4. Close the Data Dictionary Validation pop-up window once all terms have been validated.



You cannot check in a business rule/transaction/segment until all fields/term/math variable names are validated.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Internationalization > Internationalization Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Internationalization Overview

The Oracle Insurance Policy Administration (OIPA) system can be configured for use in any language. The Rules Palette contains the tools that are used to configure the system for use in multiple languages and multiple currencies. The available tools are described below.

High Level Steps to Setting Up Translations

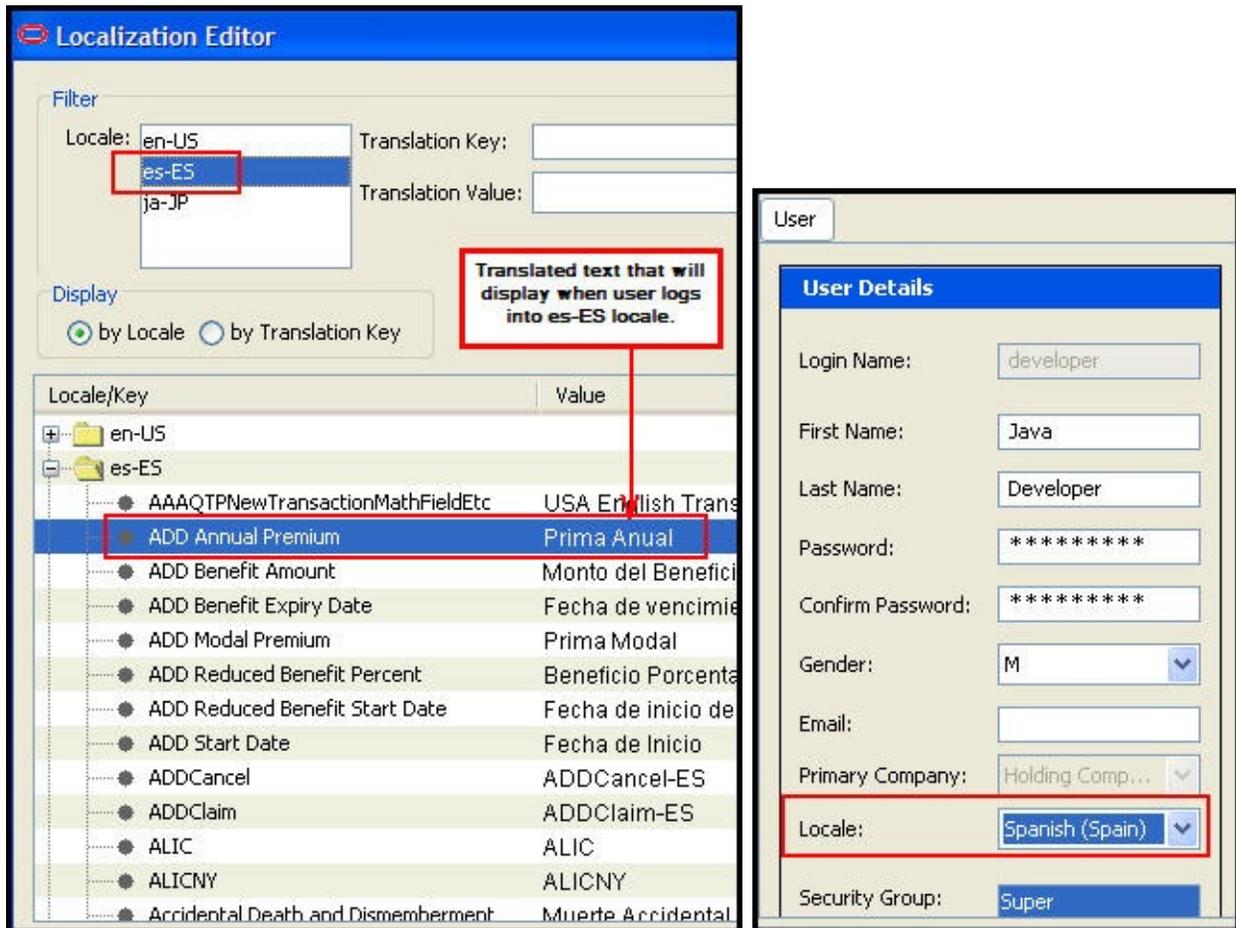
There are several steps involved in entering translations and associating users to those translations so they can be viewed in OIPA.

1. Upload the translations to the AsTranslation database table. The Localization Editor can be used to help with this. Each translation must be associated with a locale. For example, each English translation would be associated with the locale en-US.
2. [Create a new locale in AsCodeLocale](#) that corresponds to the locale the translations are associated with.
3. Create a new OIPA user. [The locale associated with the user](#) must be the same locale used in the two steps above.

Translations

The [Localization Editor](#) is used to enter translations for the underlying application displays, such as navigation and table displays, as well as [displays for transactions](#), dynamic field elements, and screen messages. When a user signs in to OIPA, the locale associated with that user's profile will determine the language that is used for navigation, tables, dynamic field elements, transactions and any messages displayed on the screen.

The Localization Editor also supports the [control of decimal displays for currencies](#). Each locale will have an associated decimal display that is stored in the AsTranslation table with the key Number.Decimal.



Localization Editor and User Details from Rules Palette User Security File

Currency Editor: Manage multiple currencies within the OIPA system with the Currency Editor. The currency information is displayed according to the international standards outlined in the ISO 4217 Currency Names and Codes Element Table. The Currency Editor allows currency names and rounding methods to be defined and updated. The three-letter currency code cannot be edited.



The screenshot shows a window titled "Currency" with an "Add" button and a table of currency configurations. The table has the following columns: Currency Code, Currency Name, Display Round Places, Currency Round Places, Currency Round Method, and Edit. The data rows are as follows:

Currency Code	Currency Name	Display Round Places	Currency Round Places	Currency Round Method	Edit
EUR	EURO	1	3	Round To Even	
GBP	GBP	7	7	Round Down	
INR	Indian Rupee	7	5	Round Down	
TWD	New Taiwan Dollar	2	2	Automatic Round	
USD	UNITED STATES DOLLAR	2	5	Automatic Round	
YEN	Japanese Yen	6	4	Round Down	

Currency Editor

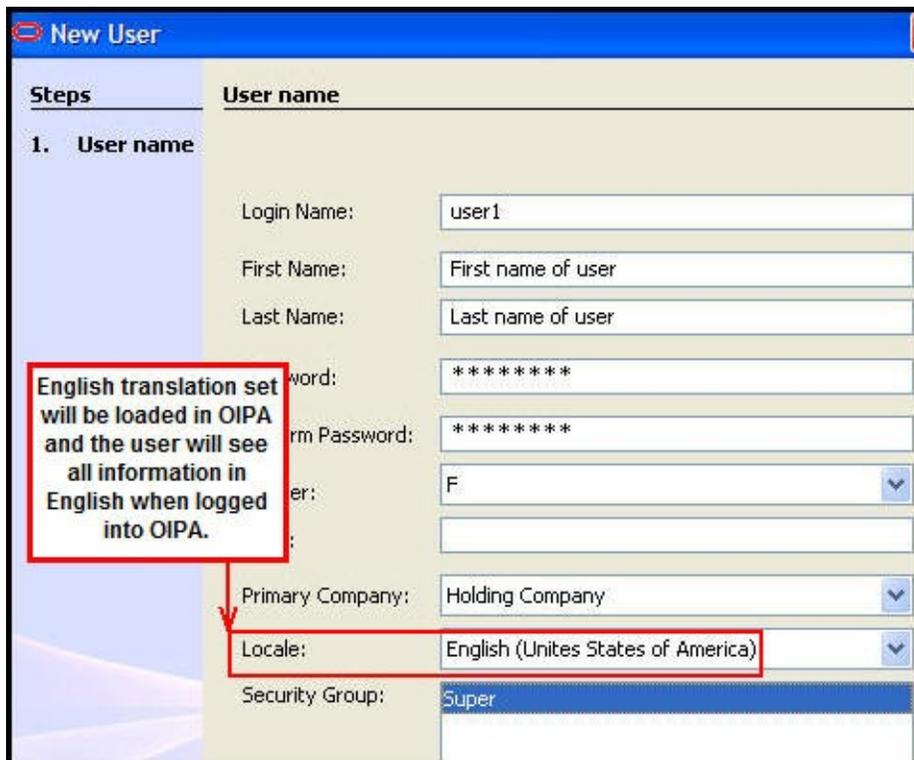
You are here: Internationalization > Associate a User with an OIPA Translation Set

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Associate a User with an OIPA Translation Set

The Rules Palette allows all OIPA fields, screens and messages to be translated into multiple languages. [When a new user account is created](#), the user is associated with a Locale. This locale determines the translation set that is loaded in OIPA when the user logs in. If the Locale is identified as Japanese, the japanese translation set will be loaded when the user signs into OIPA and all fields, screens and messages will be translated into Japanese.

Note: The Login screen displays by default in English. If a different translation for this screen is required, then the PAS.PROPERTIES file must be updated and the property **application.defaultLocale** must be set to the desired translation.



The screenshot shows a 'New User' form with the following fields and values:

Steps	User name
1. User name	
Login Name:	user1
First Name:	First name of user
Last Name:	Last name of user
Word:	*****
Form Password:	*****
er:	F
Primary Company:	Holding Company
Locale:	English (Unites States of America)
Security Group:	Super

English translation set will be loaded in OIPA and the user will see all information in English when logged into OIPA.

Translation Set Identified in User Account

You are here: Internationalization > Unit Linked Product > Configure Price Offsets

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Configure Price Offsets

Some unit linked activities require that one or more offset business days be applied to the activity effective date so that a future dated (unknown) exchange rate can be used to exchange between currencies. Once the Guaranteed Exchange Date is known, the exchange rates for the activity will be the next available rates on or after that date. An activity cannot proceed to value funds without the exchange rates.

Exchange Dates and Fund Price Offsets

There are multiple ways to configure price offsets. An explanation of each option is given below.

- Configure a field to hold an offset value. A field can be added to a transaction, or added to [Company Data](#) or [Plan Data](#). This field can then be referenced in a math variable and passed into the MathStatement function. Using fields at the plan or company level to hold offset values allows these values to be applied across multiple plans and companies.
- Configure a math variable that holds a value, which will be passed to the MathStatement function. This option does not allow the offset to be reusable across plans and companies.

The MathStatement will provide the exchange date based on activity effective date and the number of offset days specified in the mathvariable. A [MathStatement](#) function must be included in the Math section to tell Assignment how to retrieve prices and exchange rates.

OIPA will access the [AsSystemDate](#) table to verify if the day for the exchange is an actual business day. If the day is not an actual business day, then the date will advance to the next business day.

Exchange Dates Based on Time of Day

Unit dealing is a time sensitive process. A transaction may need to determine the currency exchange offset to use based on the local time of day when the activity is entered. Transactions added before a cutoff time may use 0 offset days while those issued after the cutoff may get a different number of offset days. Since the offset is based on the activity effective date the **TransactionTimes** rule can be used in conjunction with the configured price offset to dynamically to change the activity effective date that the offset is applied to. Syntax is also available for the Effective Date attribute of the transaction element so when the screen loads, it automatically advances the date to the next business day. Refer to the XML Configuration Guide for specific details concerning the Transaction element.

You are here: Internationalization > Unit Linked Product > RollingValuation

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Rolling Valuation

Valuation records are written to the database as the net asset values become available based on guaranteed price dates. OIPA can track the rolling valuation process by recording the system date when an individual record is written (Order Date) as well as shadowed (Shadow Order Date). The Shadow Order Date is a reversal date and is the same as the ActiveToDate of the parent AsActivity or related AsValuation record.

While these records are written as soon as their value is determined along with the rolling valuation process, accounting (Chart of Accounts) is only written when the valuation is [active](#).

A math variable is configured in each activity to hold the number of [offset days](#). The value of the math variable can be dynamically set in the configuration through any calculation process including retrieval from the plan fields.

Exchange Dates and Guaranteed Price Dates

Exchange Dates and Guaranteed Price Dates are calculated using the [MathStatement](#) function FindNextExchangeDate. OIPA will move through a series of steps to determine these dates and arrive at an Actual Price Date. The steps are described below.

1. OIPA compares the Activity Effective Date with the [Market Maker's](#) calendar to make sure the date is a business day. Two scenarios can occur:
 - If it **IS** a business day, then OIPA will add any offset business days to the date as identified in the Input parameter of the MathStatement function.
 - If it is **NOT** a business day, then OIPA will advance to the next business day for the calendar and then add any offset business days to that date.
2. The date OIPA calculated in step one is the Exchange Date for the activity. OIPA compares the Exchange Date to the Fund's calendar to make sure it is a business day.
 - If it **IS** a business day, then OIPA will add any offset business days to the date as defined on the AsPriceOffset entry for that calendar or fund if a fund override exists.
 - If it is **NOT** a business day, then OIPA will advance to the next business day for that calendar and then add any offset business days to that date.
3. The date OIPA calculated in step 2 is the Guaranteed Price Date for the fund. These three steps are repeated for each fund in the activity.
4. The Actual Price is the next price available for a fund on or after the Guaranteed Price Date.

Activity Advances to Active Status

The activity will remain in NUVPending status while the above steps are performed by OIPA when all the actual prices are known for all the funds in the activity.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

You are here: Utilities > Localization Editor > Localization Editor

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Localization Editor

The Localization Editor manages multiple translations for the display of transaction names and dynamic fields as well as any messages that appear on the screen in the Oracle Insurance Policy Administration (OIPA) system. The user can create [locales](#) and [add and edit existing translations](#). Currency [decimal displays](#) can also be set.

When a locale is created it should be associated with a particular country. Within those various locales, a translation value can be assigned to the transaction names, dynamic fields and screen messages. When a user is assigned a specific [OIPA security group](#), the user is also associated with a locale. The locale selected will determine what language the transaction names and dynamic fields will display in when populated in the OIPA system. Translation values are stored in the AsTranslation table.

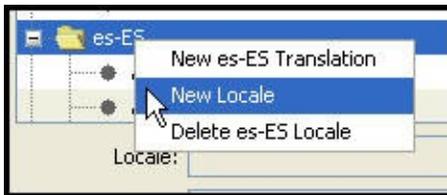
The Localization Editor is accessed from the Tools tab of the Main Menu.



Localization Editor Option on the Tools Drop Down Menu

Steps to Create a New Locale

1. Click **Tools** from the Main Menu.
2. Select **Localization Editor**. The editor will open.
3. Right-click on one of the locale folders and select **New Locale**.



Locale Right-Click Options

4. Click to select a country from the left side of the screen.
5. Click to select a language from the right side of the screen.
6. Click **OK**. The new locale will display with a folder icon. All translations associated with that locale will be listed beneath it.

Delete a locale by right-clicking on the locale folder and selecting **Delete Locale**. A delete message will appear next to the locale and it will be removed when the Localization Editor is closed.



Make sure to enter the new locale in [AsCodeLocale](#) so that the [translations can be associated to a user](#) for viewing in OIPA.

Steps to Add a Translation

1. Open the Localization Editor.
2. Right-click on the locale folder for the country where the new translation will be added and select **New Translation**.
3. Enter a mnemonic for the message in the Translation Key field. The mnemonic should be in latin characters and should be descriptive enough to indicate what message it represents. For example, MinPremViolation would be the mnemonic for the message for a minimum premium violation.
4. Enter the translation in the Translation Value field.
5. Click **OK**.

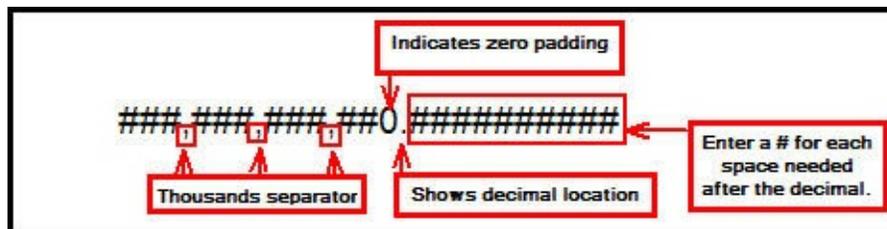
Note: In a development environment, you must restart the application server after adding a new translation in order to see the new translation in OIPA.

Steps to Edit Existing Translations

1. Click on the field that contains the translation to edit. The translation value will be displayed at the bottom of the Localization Editor.
2. Enter the new translation value.
3. Continue updating any other translation values.
4. Click **OK** when finished. The changes will be saved to the database.

Steps to Control Decimal Displays

1. Open the Localization Editor from the Tools window.
2. Right-click on a locale and select **New Translation**. The New Translation Unit window will open.
3. Enter **Number.Decimal** for the Translation Key.
4. Enter the value.
 - a. Use # for each number.
 - b. Use a comma to separate thousands.
 - c. Use a zero just before the decimal place to indicate zero padding. If zero padding is not needed, use a #.
 - d. Use a period to indicate the decimal place.
 - e. Use a # for each decimal place needed.



5. Click **OK** to save the new translation.

You are here: Utilities > Localization Editor > Translations for Screen Messages

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Translations for Screen Messages

OIPA uses the AsTranslation table to hold translations in multiple languages for system error messages and configured errors and warnings (ValidateExpressions and Events). All messages on the screen, either system or configured, have translations in the AsTranslation table.

Translations are entered into the AsTranslation table through the Rules Palette. The [Localization Editor](#), which can be found in the Tools Main Menu, provides the means for entering and editing translations.

Steps to Add a Translation

1. Open the Localization Editor.
2. Right-click on the locale folder for the country where the new translation will be added and select **New Translation**.
3. Enter the actual name of the screen message in the Translation Key field. These are always entered in English.
4. Enter the translation in the Translation Value field.
5. Click **OK**.

Steps to Edit Existing Translations

1. Click on the field that contains the translation to edit. The translation value will be displayed at the bottom of the Localization Editor.
2. Enter the new translation value.
3. Continue updating any other translation values.
4. Click **OK** when finished. The changes will be saved to the database.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

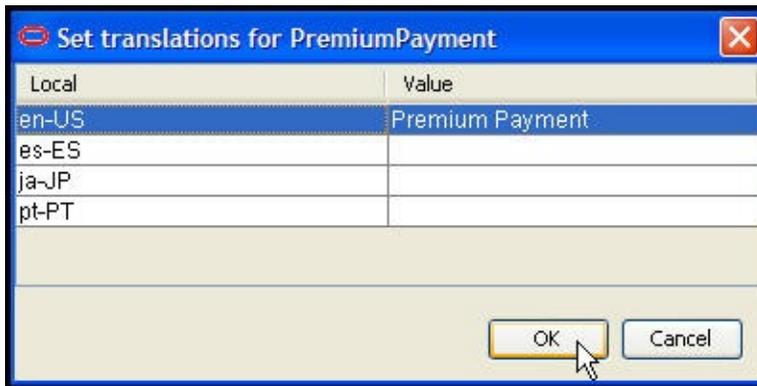
You are here: Utilities > Localization Editor > Translations for Transactions

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Translations for Transactions

Transactions have an additional method for assigning translation. When a new transaction is checked in, a translation window will display. It will list all of the locales created through the Localization Editor and will provide a place to enter the translation of the transaction name for each locale. If a translation is not entered for each locale, this window will display every time the transaction is checked in. The information entered in this window can be updated through the [Localization Editor](#).

Note: If a transaction is not assigned a translation value, then it will not appear in the activity list in OIPA.



Translation Window

You are here: Utilities > Plan Copy > Plan Copy Overview

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Plan Copy Overview

The Plan Copy utility allows plans to be copied from a source environment to a target environment. An entire plan can be copied or specific rules can be selected to create a new plan. The source and target environments must share the same Internal Versioning System (IVS).

The **source** environment is where the original plan resides. The **target** environment is where the new plan will be copied.

A copied plan may be renamed in order to distinguish it from the original plan. The utility creates a copy of a plan and all of its associated records. Selected rules for a plan can be copied leaving others not copied. New GUIDs will also be generated for a plan (if copying to the same environment) and effective and expiration dates can be modified, which is useful when creating a new similar plan. All new copied business rules, transactions and segment names are versioned. These copied items will be version one.

Once the source and target environments are selected, a list of plans that are available to be copied from the source environment is presented to the user. The user selects a plan from the list. The user then selects the level the plan should be copied to in the target: Company, Subsidiary Company or Product. A new level must be the same level as the originating plan. A company level plan must copy as a company level plan. New levels cannot be created at the time of plan copy. Any new levels must be created in the target environment prior to the plan copy. A check will be performed to ensure that the appropriate levels exist before the plan copy executes.



The level of the plan must be the same in the source and target environments. For example, a Primary Company level plan in the source should be copied only to a Primary Company level in the target. A Subsidiary Company level plan should be copied to Subsidiary Company level in the target and a Product level plan

should be copied to a Product level in the target.

Products

If [Products](#) are turned ON in the source environment, but turned OFF in the target environment (or vice versa), then the plan copy will not go past the first screen and a validation error will display. This occurs because there are no valid Product parents where the plan can be copied. If Products are turned on in both environments, then all plans must be a part of a group based on current functionality.

If a user wants to copy a plan that is part of a Product in the source environment, but the Product doesn't exist in the target environment, then a two-step process must be followed.

1. Create the new Product in the target. This can be accomplished through the Plan Copy utility.
2. Copy the plan to the newly created Product in the target environment. This will ensure that the Product and plan match in the source and target environments. A plan that exists in one Product in the source may be copied into a different Product in the destination environment.

 Additional work must be completed in the target environment after the Plan Copy has completed in order to bring up and process the newly created plan. This work includes updating items such as rules security (transaction security), translations, Administration table updates, global rules missing, screen rules, Segment Calculate General rules, and Sequence table updates.

Preserve GUIDS

This option is only available if the source environment and the target environment are different. If the environments are the same, then the Preserve GUIDs option will be unchecked and disabled.

If preserve GUIDS is selected then the existing Plan name will be preserved. Also the plan name field will populate with the existing plan name and the field will become disabled. The GUIDs for all copied rules (PlanGUID, SegmentGUID, BusinessRuleGUID and TransactionGUID) will also be preserved.

Release Management

[Release Management](#) is only available if the functionality is turned on in the target environment. If it is turned on, then the copied plan can be added to a configuration package. When the configuration package is deployed to the target environment it can be copied to numerous environments without having to repeat the plan copy process each time.

Security

Security was added to the Plan Copy tool. The menu option will only be enabled if the user has **Utilities-Plan Copy** added to his [Palette security role](#).

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

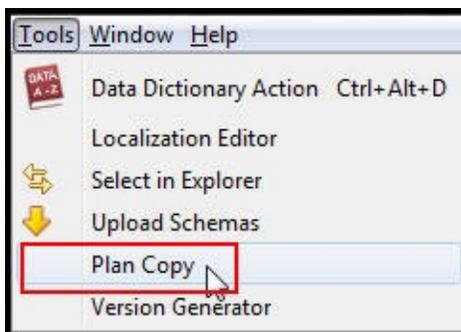
You are here: Utilities > Plan Copy > Plan Copy Wizard

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Plan Copy Wizard

The OIPA Plan Copy Wizard steps through the process of copying a plan from one IVS environment to another. The Plan Copy utility can only be used in development environments. When a plan is copied, no valuation settings will be changed or transformed.

To open the Plan Copy Wizard, click **Tools | Plan Copy**.



Plan Copy Option on Tools Menu

Steps to Copy a Plan

1. Click **Tools | Plan Copy** on the Main menu. This will open the OIPA Plan Copy Wizard.
2. Click the name of the target environment and then click **Next**. Only IVS environments that have been set up by the user and share the same IVS as the source environment are listed in this section. By default, the source environment is whatever environment the user is logged into when Plan Copy is invoked.
3. Define the source properties.
 - **Source Environment:** This is the environment the user is logged into when Plan Copy is invoked. This is not a selectable field.
 - **Company:** Select the company that contains the plan or Product that is being copied.
 - **Product:** Select the Product that is being copied. This option will be disabled unless Products are enabled.
 - **Plan:** Select the plan that is being copied.



If a user wants to copy a plan that is part of a Product in the source environment, but the Product doesn't exist in the target environment, then a two step process must be followed. First, create the new Product in the target environment. This can be accomplished by using the Plan Copy utility to copy the Product to the Target environment. Then, copy the plan to the newly created Product in the target environment. This will ensure that the Product and plan match in the source and target environments. A plan that exists in one Product in the source may be copied into a different Product in the target.

4. Define the target environment properties.
 - **Target Env:** This is the environment that was selected in step one of the wizard. This is not a selectable field.
 - **New Plan Name:** The name of the plan that is being copied will

appear here but can be edited if a plan with the same name doesn't exist in the target. If the a plan with the same name exists in the target, then this field will be a blank field.

- **Effective Date:** Select the date when this plan is available.
 - **Expiration Date:** Select the date when this plan is no longer available.
5. Determine if GUIDs should be preserved. If selected, the Plan Name field will populate with the existing plan name and the field will become disabled. The GUIDs for all copied rules (PlanGUIDs, BusinessRuleGUIDs, TransactionGUIDs, SegmentNameGUIDs) will also be preserved. If the target environment selected is the same as the source environment (the one the user is logged into), then Preserve GUIDs will be unchecked and disabled.
 6. Determine if all rules in the plan should be copied. If this checkbox is selected, then the wizard will skip to the Release Management section. If this checkbox is not selected, then the wizard will step through adding business rules, transactions, segments, funds and additional plan data.
 7. Click **Next** to advance to the next screen of the utility.
 8. Select any business rules that should be copied in the new plan and click **Next**. If **Copy All Rules** was selected on the first screen of the wizard, then this step will not be available.
 9. Select any transactions that should be copied in the new plan and click **Next**. If **Copy All Rules** was selected on the first screen of the wizard, then this step will not be available.



A check will need to verify if the global rule exists in the target environment so a duplicate is not added.



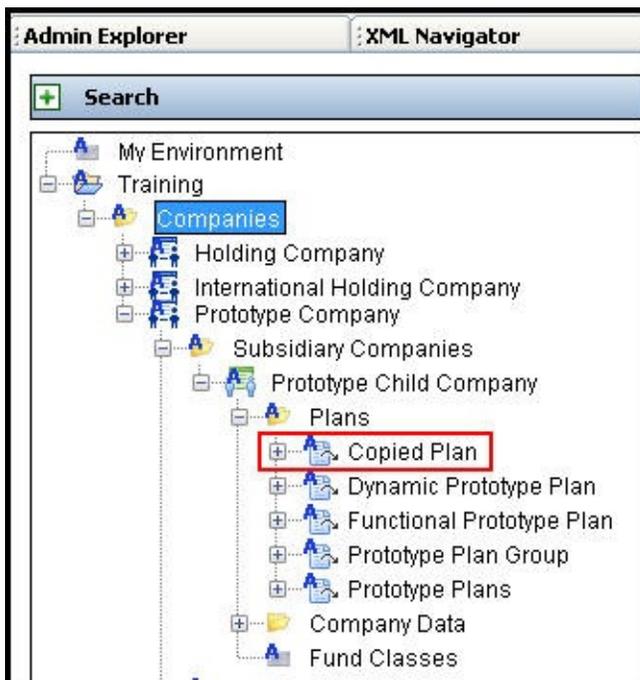
MarketMaker information will be automatically copied if at the Plan level a MarketMakerGUID exists. The MarketMakerGUID must not be null in AsPlan where the PlanGUID is equal to the Source PlanGUID.

10. Select any segments that should be copied in the new plan and click **Next**. If **Copy All Rules** was selected on the first screen of the wizard, then this step will not be available.
11. Select any plan funds that should be copied in the new plan and click **Next**. If **Copy All Rules** was selected on the first screen of the wizard, then this step will not be available. A child or benefit fund cannot be copied unless the parent fund is also selected. If a parent fund has multiple child or benefit funds, then any combination of the child or benefit funds can be selected, leaving some unchecked if needed.
12. Select the plan model definitions that should be included in the copied plan then click **Next**. If **Copy All Rules** was selected on the first screen of the wizard, then this step will not be available.
13. Select the plan program definitions that should be included in the copied plan then click **Next**. If **Copy All Rules** was selected on the first screen of the wizard, then this step will not be available.
14. Select the segment program definitions that should be included in the copied plan then click **Next**. If **Copy All Rules** was selected on the first screen of the wizard, then this step will not be available.
15. Click the box next to any additional plan data that should be included such as plan allocations, plan withholding and plan and segment state approvals.
16. Select any comment templates that should be included in the copied plan, then click **Next**. If **Copy All Rules** was selected on the first screen of the wizard, then this step will not be available.
17. Select any activity filters that should be included in the copied plan, then click **Next**. If **Copy All Rules** was selected on the first screen of the wizard, then this step will not be available.
18. Select the requirements that should be included in the copied plan, then click **Next**. If **Copy All Rules** was selected on the first screen of the wizard, then this step will not be available. If any of the selected requirements have rules attached, the attached rules will need to be selected from the Choose Business Rules screen.
19. Click **Add to Configuration Package** if the plan should be added to a configuration package for release management. Release

Management must be enabled in the target environment to use this feature.

20. Review the new plan. The top portion contains the plan set-up information and the bottom portion contains a navigation tree that displays the contents of the plan. If changes need to be made, you must move back to the screen where the original information was entered to make the changes.
21. Click **Finish** to complete the copy. The new plan should appear in the navigation tree of the environment where it was copied.

If a plan is copied into the same environment as the source, log out and back into the environment to see the copied plan.



Copied Plan in Navigation Tree

You are here: Utilities > Plan Copy > Additional Steps to Finalize Plan Copy

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Additional Steps to Finalize Plan Copy

Once a new plan is copied into the target environment, a series of steps must be completed in order to finalize the plan copy.

- Enter any map information into the map database tables:
AsMapGroups, AsMapCriteria, AsMapValue.
- Enter any rate information into the rate database tables:
AsRateGroup, AsRate.
- Enter any new code values into the AsCode database table or use the [Code Name editor](#) in the Admin Explorer..
- Add [security](#) to the plan and the new transactions.
- Enter [translations](#) for the new plan information.
- Perform any administration table updates.
- Create [screen rules](#).
- Create segment [Calculate General](#) rules.
- Perform any sequence table updates.
- Create [copybooks](#) to share common configuration.
- Create any [functions](#) that must exist at override levels other than the plan level.
- Enter unit values into the AsNetAssetValue database.
- Enter [market calendar](#) information.

You are here: Utilities > Version Generator > Version Generator

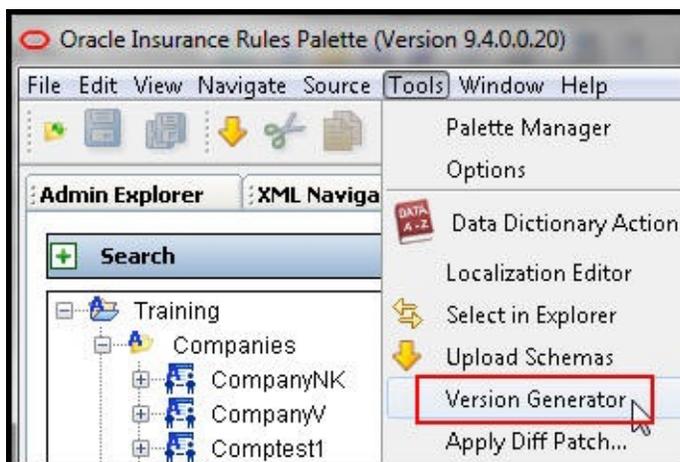
Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Version Generator

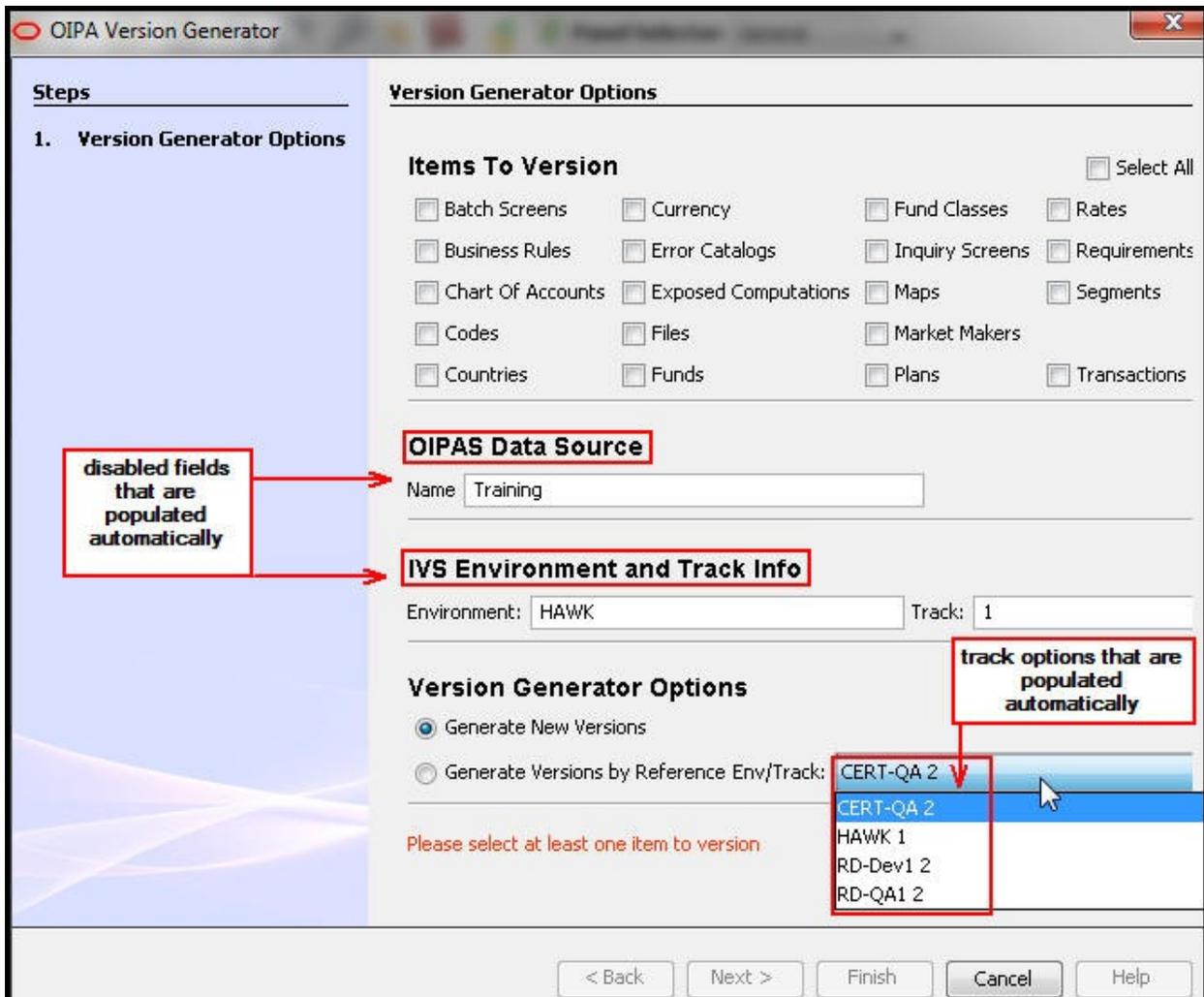
The Version Generator is a utility that manages versions of data in the database. There are two actions that can be performed with the Version Generator:

1. New versions of data can be created.
2. Existing data can be restored to a different version from a variety of database tracks.

The Version Generator can be accessed from the **Tools** menu. When the Version Generator opens, it is automatically populated with the appropriate database information and track options.



Version Generator Option on Main Menu

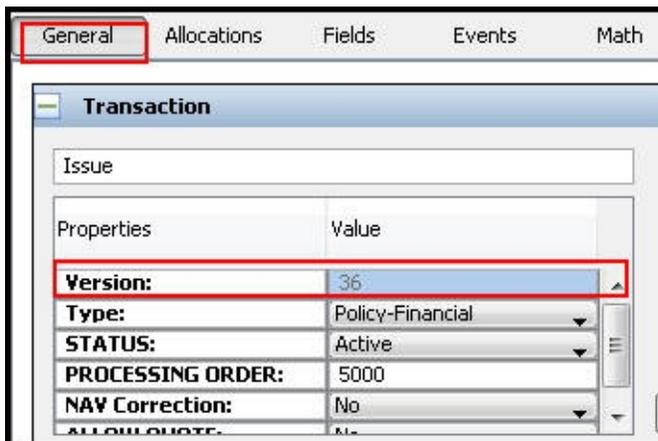


Version Generator Window in Rules Palette

Steps to Create a New Version of Data

1. Select **Tools** from the Main Menu and click the **Version Generator** option. The Version Generator window will open.
2. Click the checkbox next to the type of data that requires a new version. Multiple boxes may be checked, or the Select All box may be checked to include all data types.
3. Click **Generate New Versions** in the Version Generator Option section at the bottom of the window.
4. Click **Finish**.

The data types that were selected in the first section of the window should have their version numbers incremented by one. You can check this by opening one of the data types and checking the version number on the General Pane.



Version Number on General Pane of a Transaction

Steps to Restore a Different Version

1. Select **Tools** from the Main Menu and click the **Version Generator** option. The Version Generator window will open.
2. Click the checkbox next to the type of data that requires a new version. Multiple boxes may be checked, or the Select All box may be checked to include all data types.
3. Click **Generate Version by Reference Env/Track** in the Version Generator Option section at the bottom of the window.
4. Select the Track containing the version you want to use to overwrite the existing version.
5. Click **Finish**.

The data types that were selected in the first section of the window should have their version numbers restored to the version number of the data referenced in the Track in Step 3. You can check this by opening one of the data types and checking the version number on the General Pane.

You are here: Utilities > Diff Reports > Detailed Diff Reports

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Detailed Diff Reports

The Rules Palette allows for the comparison of the entities (i.e. business rules, segments, transactions, etc.) in one environment to those in another, and for the creation of reports that summarize these differences. The comparison of two environments may be needed in order to determine why one is behaving differently from another, among other reasons.

Two types of diff reports are available in the Rules Palette: summary diff reports and detail diff reports. This page explains detailed diff reports, which compare a single entity (e.g. a single transaction, business rule override or code name) across two environments. See the instructions below for information on creating and interpreting detailed diff reports.

Creating a Detailed Diff Report

1. Along with the "Show Detail" button in the Summary Report, a right-click **Compare To** option on an entity node is available. See the [Nodes Available for Detailed Diff Reports](#) section below for a complete list of entities that can be compared in this manner.
2. This will display a sub-menu containing the names of the other environments. Select one to compare its entity of the chosen type to that entity already selected in the current environment already selected.
3. There are three possible outcomes:
 - If the entity does not exist in the target environment, a message will display that reads, "Rule does not exist in the Target Environment."
 - If the entities are the same in both environments, a message will display that reads, "No differences were found."
 - If the entities differ across the two environments, the Detailed Diff Report window will launch.

Detailed Diff Report Window

The Detailed Diff Report window illustrates the specific differences between two of the same entity in two different environments. The top section of the screen displays information relating to the entity being compared—it's type, name, GUID, etc.—while the bottom section displays the XML for the entity in both environments, with visual highlights illustrating the exact differences.

If a section of configuration exists in the source environment, but not in the target environment, the section will be highlighted in red, with a line extending into the target environment XML indicating where the difference occurs.

<SegmentStateApproval>	2	2	<SegmentStateApproval>
<StateApprovalGuid>CC76	3	3	<StateApprovalGuid>CC76
<SegmentNameGuid>31839C	4	4	<SegmentNameGuid>31839C
<EffectiveDate>11/02/12	5	5	<EffectiveDate>11/02/12
<ExpirationDate></Expir	6	6	<ExpirationDate></Expir
<StateCode>01</StateCod	7	7	<StateCode>01</StateCoc
</SegmentStateApproval>	8	8	</SegmentStateApproval>
<SegmentStateApproval>	9	9	</SegmentStateApprovals>
<StateApprovalGuid>2073	10		
<SegmentNameGuid>31839C	11		
<EffectiveDate>01/01/10	12		
<ExpirationDate>02/01/1	13		
<StateCode>07</StateCod	14		
</SegmentStateApproval>	15		
</SegmentStateApprovals>	16		

XML that exists in the source environment, but not the target environment, highlighted in red

If a section of configuration exists in the target environment, but not in the source environment, this scenario will be reversed, and the XML section will be highlighted in green.

<Math>	18	19	<MathVariables>
<MathVariables>	19	20	<MathVariable VARIABI
<MathVariable VARIABLEN	20	21	<MathVariable VARIABI
</MathVariables>	21	22	</MathVariables>
</Math>	22	23	</Math>
</Transaction>]]</XMLData>	23	24	</Transaction>]]</XMLData>
<ProcessingOrder>1</ProcessingOrder>	24	25	<ProcessingOrder>1</ProcessingOrder>
<PriceCorrectionIndicator>Y</PriceCorrec	25	26	<PriceCorrectionIndicator>Y</PriceCor
</Transaction>	26	27	</Transaction>

XML that exists in the target environment, but not the source environment, highlighted in green

If the two environments have the same sections of XML, but have differences within those sections, the sections will be highlighted in blue in both environments' XML.

<AccountingTypeCode>03</AccountingTypeCode>	20	20	<AccountingTypeCode>03</AccountingTypeCode>
<AccountingAmountField>COIPaid</AccountingAmountField>	21	21	<AccountingAmountField>COIPaid</AccountingAmountField>
<GainLossFlag>0</GainLossFlag>	22	22	<GainLossFlag>0</GainLossFlag>
<FlipOnNegativeFlag>0</FlipOnNegativeFlag>	23	23	<FlipOnNegativeFlag>0</FlipOnNegativeFlag>
<EffectiveFromDate>12/31/79</EffectiveFromDate>	24	24	<EffectiveFromDate>01/01/80</EffectiveFromDate>
<EffectiveToDate></EffectiveToDate>	25	25	<EffectiveToDate></EffectiveToDate>
<DoReversalAccountingFlag>1</DoReversalAccountingFlag>	26	26	<DoReversalAccountingFlag>1</DoReversalAccountingFlag>
<OriginalDisbursementStatusCode></OriginalDisbursementStatusCode>	27	27	<OriginalDisbursementStatusCode></OriginalDisbursementStatusCode>
<FundTypeCode></FundTypeCode>	28	28	<FundTypeCode></FundTypeCode>

An XML section that differs between environments highlighted in blue

The Detailed Diff Report window also gives users the ability to export the XML for the source environment and/or the target environment. To do so, click the **Export Source** or **Export Target** button at the bottom of the window. The XML will save as a .txt file by default.

Nodes Available for Detailed Diff Reports

The Compare To option is available on the following entity nodes:

- Activity Filters
 - On each activity filter XML file
- Allocation Models
 - On each model's XML file under the primary company node (*Primary company | Models | Model name | Model name.xml*)
 - On the model name node under the subsidiary company node (*Primary company | Subsidiary Companies | Subsidiary company | Plan | Model name | Model name.xml*)
- Fund Asset Classes
 - On the company XML file above the fund names
- Batch Screens
 - On the main Batch Screens node
- Business Rules
 - On each business rule's XML file
- Chart of Accounts
 - On each Chart of Accounts name node
- Codes
 - On the XML file under each code name
- Companies
 - On the Company Data XML file
- Countries
 - On the Countries XML file
- Currency
 - On the Currency XML file
- Cycle Sequence
 - On the Cycle Sequence XML file
- Error Catalog
 - On the Error Catalog XML file

- Exposed Computations
 - On the XML file under each Exposed Computation name node
- Files
 - On the XML file under each File name node
- Funds
 - On the XML file under each Fund name node
 - On the XML file under the Fund Asset Classes node for each fund
 - On the XML file under each ChildFunds node
- Inquiry Screens
 - On each Inquiry Screen node
- Map Groups
 - On the XML file under each Map Group name node
- Market Maker
 - On the MarketMaker.xml file
- Masks
 - On each mask XML file
- Plans
 - On the Plan Data XML file under the Plan Data node
- Plan Models
 - Under each plan name node for the model name
- Plan State Approvals
 - On the State Approval node under each Plan
- Programs
 - On the main Program Definition node



Note: The Compare To option is not available for Plan or Segment program nodes.

- Rate Groups
 - On each rate group XML file
- Requirements

- On the requirement XML file under each company
- Segments
 - On each segment XML file
- Segment State Approvals
 - On the state approvals XML file under each segment
- System Date
 - On the System Date XML file.
- Transactions
 - On each transaction XML file
- Web Services
 - On each Web Services XML file

You are here: Utilities > Diff Reports > Summary Diff Reports

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Summary Diff Reports

The Rules Palette allows for the comparison of the entities (i.e. business rules, segments, transactions, etc.) in one environment to those in another, and for the creation of reports that summarize these differences. The comparison of two environments may be needed in order to determine why one is behaving differently from another, among other reasons.

Two types of diff reports are available in the Rules Palette: Summary Diff Report and Detail Diff Report. This page explains Summary Diff Reports, which allow for the comparison of multiple entity types across the two environments. See the instructions below for information on creating and interpreting Summary Diff Reports.

Creating a Summary Diff Report

1. Click on the **Tools** option in the main menu at the top of the screen and select **Summary Diff Report**.
2. The OIPAS Full Environment Difference Report wizard will open. In the Items to Compare section, check the **Select All** box to compare all the entities within the two environments, or check the boxes next to the individual entity types to compare only those types of entities. Any number of entity types may be selected.
3. The **OIPAS Source Data Source** field will be disabled and populated with the name of the environment that the user is logged in to. Choose the second environment from the **OIPAS Target Data Source** drop-down box.
4. Click **Finish**. The Summary Report will generate, and the Full Environment Difference Report window will display, summarizing the differences between the two environments.

Full Environment Difference Report Window

The Full Environment Difference Report window presents all of the differences found between the two environments. See the descriptions of each of this window's columns below.

- **Entity Type:** The type of entity (e.g. business rule, segment, transaction, etc.) for which the difference is found.
- **Entity Name:** The name of the entity. An entity's name will be pulled from the appropriate column of the entity's database table.
- **Entity GUID:** If applicable, the entity's GUID. An entity's GUID will be pulled from the appropriate column of the entity's database table.
- **Override Level:** If applicable, the entity's override hierarchy.
- **Difference Type:** The way that the entity differs between the two environments. There are three types of differences.
 - **Source Only:** The entity is only found in the source environment.
 - **Target Only:** The entity is only found in the target environment.
 - **Matched Diff:** The entity is found in both environments, but differs.
- **Difference Column:** The column in the entity's database table where the difference is found. If differences are found in multiple database columns, the column names will be separated by commas.
- **Source VersionNumber:** The version number of the source environment. This column will only be populated with data if the environments are connected to the same IVS database.
- **Source LastModifiedBy:** The last user to modify the entity in the source environment. This column will only be populated with data if the environments are connected to the same IVS database.
- **Source LastModifiedGMT:** The time (in GMT) that the entity was last modified in the source environment. This column will only be populated with data if the environments are connected to the same IVS database.
- **Target VersionNumber:** The version number of the target environment. This column will only be populated with data if the environments are connected to the same IVS database.

- **Target LastModifiedBy:** The last user to modify the entity in the target environment. This column will only be populated with data if the environments are connected to the same IVS database.
- **Target LastModifiedGMT:** The time (in GMT) that the entity was last modified in the target environment. This column will only be populated with data if the environments are connected to the same IVS database.

The Full Environment Difference Report window also offers two options for further analyzing and utilizing diff report data: the **Show Detail** button and the **Export to CSV** button. To use these buttons, select a row in the Full Environment Difference Report window and click the desired button.

- **Show Detail** launches a Detailed Diff Report illustrating the differences between two entities. Detailed Diff Report can only be generated for entities with the Matched Diff difference type. See the [Detailed Diff Report Window](#) section below for more information on using and interpreting the Detailed Diff Report window.
- **Export to CSV** allows the user to save the summary diff report as a comma-separated values (CSV) file, complete with the column headings from the Summary Diff Report. This file may be imported into Excel and manipulated/sorted if necessary.

Entities Available for Summary Diff Report

The following entities are able to be compared using Summary Diff Report, each with its own checkbox in the OIPAS Full Environment Difference Report wizard. Also included below are the database tables that are compared for each entity in order to generate the diff reports.

- Activity Filters
 - **Database tables compared:** AsFilter
- Allocation Models
 - **Database tables compared:** AsModelDefinition, AsModel, AsModelFund, AsModelAssetClass, AsModelAssetClassFund
- Asset Classes
 - **Database tables compared:** AsAssetClass, AsAssetClassFund
- Batch Classes
 - **Database tables compared:** AsBatchScreen
- Business Rules
 - **Database tables compared:** AsBusinessRules
- Chart of Accounts
 - **Database tables compared:** AsChartOfAccounts, AsChartOfAccountsEntity, AsChartOfAccountsEntry, AsChartOfAccountsMoneyType, AsChartOfAccountsCriteria, AsChartOfAccountsResults
- Codes
 - **Database tables compared:** AsCode
- Countries
 - **Database tables compared:** AsCountry
- Currency
 - **Database tables compared:** AsCurrency
- Companies
 - **Database tables compared:** AsCompany, AsCompanyField
- Comment Templates
 - **Database tables compared:** AsCommentsTemplate

- Error Catalogs
 - **Database tables compared:** AsErrorCatalog
- Exposed Computations
 - **Database tables compared:** AsExposedComputation
- Files
 - **Database tables compared:** AsFile
- Funds
 - **Database tables compared:** AsFund, AsFundField
- Fund Relations
 - **Database tables compared:** AsFundRelation, AsFundRelationField
- Inquiry Screens
 - **Database tables compared:** AsInquiryScreen
- Map Groups
 - **Database tables compared:** AsMapGroup, AsMapValue, AsMapCriteria
- Market Makers
 - **Database tables compared:** AsMarketMaker, AsMarketMakerCurrency
- Masks Details
 - **Database tables compared:** AsMaskDetail
- Plans
 - **Database tables compared:** AsPlan, AsPlanField, AsWithholding (RelatedGUID = PlanGUID), AsWithholdingField, AsPlanDefaultAllocation, AsAllocation (if using old Allocation method), AsPlanDefaultAllocationModel, AsPlanDefaultAllocationFund

Note: If Model Supported Plan Allocation is used, then AsPlanDefaultAllocation, AsPlanDefaultAllocationModel, and AsPlanDefaultAllocationFund will be compared instead of AsAllocation.

- Plan Funds

- **Database tables compared:** AsPlanFund, AsPlanFundStatus
- Plan Models
 - **Database tables compared:** AsPlanModelDefinition, AsPlanModelDefintionStatus
- Plan State Approvals
 - **Database tables compared:** AsPlanStateApproval
- Program Definitions
 - **Database tables compared:** AsProgramDefinition, AsPlanProgramDefinition, AsSegmentProgramDefinition
- Requirements
 - **Database tables compared:** AsRequirementDefinition, AsRequirementGroup, AsRequirementCriteria
- Segments
 - **Database tables compared:** AsSegmentName
- Segment State Approvals
 - **Database tables compared:** AsSegmentStateApproval
- System Dates
 - **Database tables compared:** AsSystemDate
- Transactions
 - **Database tables compared:** AsTransactions
- Web Services
 - **Database tables compared:** AsAuthWebService

You are here: References > Rules not listed in TransactionBusinessRulePacket

Oracle® Insurance Rules Palette 10.1.2.0 E59346-01

Rules Not Listed in the TransactionBusinessRulePacket

The following business rules should **not** be listed in the TransactionBusinessRulePacket business rule because they do not process data. All other business rules that are an [Attached rule type](#) should be listed in the order in which they should be processed.

1. FundListForAllocation
2. GenerateDocument
3. CopyToScheduledValuationFields
4. ReportFile
5. ReportDetails
6. TransactionCosmetics
7. DisbursementNumber
8. ReverseScreen
9. ActivitySummary
10. SummaryFields
11. TransactionBusinessRulePacket
12. ConfirmationScreen
13. QuoteScreen
14. VerificationScreen

Note: FundListForAllocation is a system rule, but it can sometimes be overridden at the transaction level so that is why it is included in this list.

Fund Asset Classes

After funds are created they can be organized into asset classes. This must be completed in order to use the funds later in allocation models. [Allocation models](#) give policy owners a way to invest their money in different categories of assets, typically cash equivalents, bonds and stocks, which then diversifies their portfolio.

In order to organize funds into fund asset classes, fund asset class groups must first be created and then associated to funds. Once that step is complete, allocation models can be configured.

High Level Steps to Create Fund Asset Classes

1. Create the Fund Asset Class.
2. Associate funds with a Fund Asset Class.

Steps to Create Fund Asset Classes

1. Create the Fund Asset Classes.
 - Navigate to the **Admin Explorer**.
 - Expand the **Administration | Fund Asset Classes | Company Name** folder.
 - Check out the Company.xml file.
 - Select the **Add** button in the Asset Classes pane to create a new asset class.
 - Type the name of the asset class in the Asset Class column.
 - Select the ellipses button to enter a translation for the asset class.
 - Check in the Company.xml file to save the changes.
2. Associate a fund with an asset class.
 - Navigate to **Administration | Fund Asset Classes | Company Name Fund | Fund Asset Classes**.
 - Check out the Fund Asset Classes.xml file.
 - Select the **Add** button in the Fund Asset Classes pane.
 - From the drop-down box select the asset class the fund belongs to.
 - Check in the Fund Asset Classes.xml file to save the changes.

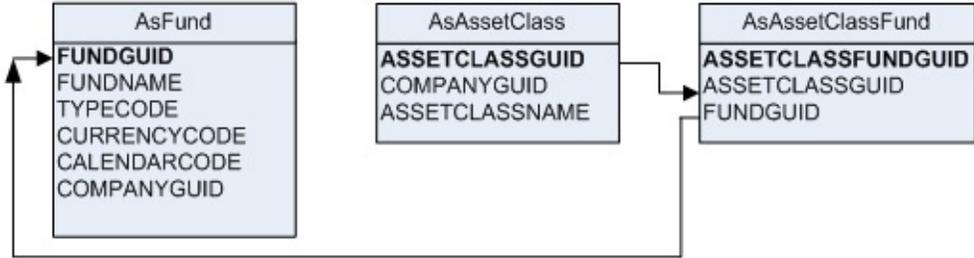
When finished you can review the funds that have been associated with the various fund asset classes.

Steps to Review the Funds Associated with Fund Asset Classes

1. Navigate to the Admin Explorer and open **Administration | Fund Asset Classes | Company | Fund Asset Class**.
2. Double click the **Fund Asset Class.xml** file.
3. View the funds listed in the Fund Asset Class.

This is the only way to view the funds in an asset class. There is no functionality to add or delete funds.

Fund Asset Class Database Tables



Fund Asset Class Database Tables

State Approvals Overview

Domestic Insurance companies are required to obtain approval for products that are available for sale in each state. States may approve the product, but with conditions that exclude a particular feature (segment) of that product.

Insurance companies may begin to sell a product in some states before it has been approved for sale in all states. The Rules Palette provides a State Approval editor to record the states that have approved sale of an insurance product by plan and segment. The state approval records are assigned a starting effective date and end or expiration date. The system uses these state approval records to control whether or not a plan or segment is available in the UI for a particular state.

State Approval Navigation Tree

Plans must be set up to support state approval. This is done using the [PlanScreen business rule](#). Each plan that is set up to support state approvals will have a state approval node, Version History node and Segment State Approvals node.

- **State Approval:** There are three right-click options that can be performed on this node: Check-in/Check-out, Revert Modifications and Delete Rule. When Delete Rule is selected, all state approval records for the plan will be deleted. Selecting the Check-out right-click option opens the State Approval editor.



The Delete action will not be allowed if segments within the plan contain state approval records.

- **Version History:** Right-click on a particular version record listed under this node to revert to that version.
- **Segment State Approvals:** Each segment that is set up to support state approval will be listed under this node. The [SegmentScreen rule](#) must be overridden at the plan level and state approval support must be added for every segment that requires state approval support. If segments have been granted support, then when the Segment State Approval folder is opened, the segments will display. Open the segment folder to expose the State Approval editor for that segment. The right-click options are the same as those for State Approval at the plan level.

Refer to the [State Approval editor](#) section for additional information on creating state approval records.

Security

The State Approval node in the Admin Explorer tab is governed by the Rules Palette's Administration privileges, "Administration-All Non-Security Administration-CheckIn/CheckOut" and "Administration-All Non-Security-View". If a user is granted one of these privileges, then access will be granted to the State Approval editor.

OIPA users may view information loaded into the State Approval table when the **Tables | State Approval** option is selected from the OIPA Main menu. This information will only be visible to the user if security privileges are granted through the Security folder in the Admin Explorer.

To allow OIPA users to view the State Approvals table in OIPA, click the Admin Explorer tab and navigate to **Security | Application Security | Security Groups | *Name of the Security Group* | Company Security | Company Pages | *Name of the Company***. The State Approval option is listed under the Company. Check out the file and click the box to grant access to the State Approval table.



State Approval Page in Company Pages Security

Prepare Spreadsheet for Rate Upload

When a spreadsheet is the source for uploading rates, a very specific format must be used. All required metadata and rate table values must be provided in the spreadsheet and the file type must be XLS (*.xls). Multiple rate groups that have the same rate description can be entered using the same workbook.

The first worksheet in the workbook is called the **Rate Legend**. This is where all of the metadata is provided. The information in the legend worksheet and the individual rate sets will vary depending on the type of rate table that is uploaded: Aggregate, Select, or Ultimate. One worksheet is required to upload one rate set under a rate group.

Rate Legend

The legend worksheet contains the metadata of the rate table in the form of two columns. The data needs to be presented exactly as described below.

The Legend key will begin in cell A2, with the corresponding Legend values beginning in B2. The Legend key provides the user with the literal name of the expected value. The Legend worksheet should be structured as follows:

- **A2** – "Group Name"
- **B2** – The name of the rate group. This value provides the "RateDescription" for the rate table. There may be multiple rate sets with the same rate group name. If this is the case, the rate sets will be differentiated by the Rate Activation, Active From and Active To dates.
- **A3** – "Table Type"
- **B3** – "Select", "ultimate" or "aggregate." Some of the following legend information is dependent on the table type specified here.
- **A4** – "Primary Index"
- **B4** – The value here is the "IntegerCriteria" for the rate table.
- **A5** – "Primary Index Orientation"
- **B5** – "Column" or "row." This is the orientation of the integer criteria on the rate set worksheets.
- **A6** – "Secondary Index"
- **B6** – A value is required here for Select and Ultimate table types. The Secondary Index is one of the 10 optional criteria. It is used to determine when to stop loading.
- **A7** – "Secondary Index Data Type"
- **B7** – The value here is the data type of the secondary index mentioned above.

- **A8** – "Select Period/Duration"
- **B8** – A value is required here for the Ultimate table type. A value is required for the Select and Aggregate table types if a Maximum Age is not provided.
- **A9** – "Maximum Age"
- **B9** – A value is required here for the Ultimate table type. A value is required for the Select and Aggregate table types if a Select Period/Duration is not provided.
- **A10 to A19** – Criteria 1 to Criteria 10.
- **B10 to B19** – The criteria names for each of the criteria. These are not mandatory, but need to be in sync with the rate set worksheets. In the case of an existing rate description, the criteria cannot differ from the criteria in AsRateGroup table.

After the legend is complete, a worksheet must be created for each rate set. Refer to the [Rate Set Worksheet](#) page for additional information.

A	B
1 Legend	
2 Group Name	FEVUL_Commission_Target_Premiums
3 Table Type	Aggregate
4 Primary Index	IssueAge
5 Primary Index Orientation	column
6 Secondary Index	
7 Secondary Index Data Type	
8 Select Period/Duration	
9 Maximum Age	81
10 Criteria1	Gender
11 Criteria2	UWClass
12 Criteria3	
13 Criteria4	
14 Criteria5	
15 Criteria6	
16 Criteria7	
17 Criteria8	
18 Criteria9	
19 Criteria10	

Sample Aggregate Rate Legend

	A	B
1	Legend	
2	Group Name	FEVUL_Mortality_Charges
3	Table Type	Select
4	Primary Index	Duration
5	Primary Index Orientation	column
6	Secondary Index	IssueAge
7	Secondary Index Data Type	int
8	Select Period/Duration	
9	Maximum Age	99
10	Criteria1	
11	Criteria2	Gender
12	Criteria3	UWClass
13	Criteria4	FaceAmountBand
14	Criteria5	
15	Criteria6	
16	Criteria7	
17	Criteria8	
18	Criteria9	
19	Criteria10	

Sample Select Rate Legend

	A	B
1	Legend	
2	Group Name	FEVUL_Mortality_Charges
3	Table Type	Ultimate
4	Primary Index	Duration
5	Primary Index Orientation	column
6	Secondary Index	IssueAge
7	Secondary Index Data Type	int
8	Select Period/Duration	99
9	Maximum Age	99
10	Criteria1	
11	Criteria2	Gender
12	Criteria3	UWClass
13	Criteria4	FaceAmountBand
14	Criteria5	
15	Criteria6	
16	Criteria7	
17	Criteria8	
18	Criteria9	
19	Criteria10	

Sample Ultimate Rate Legend

Retrieve Rates Using Stored Procedure

When OIPA encounters a [RATE](#) or [RATEARRAY](#) math variable type, it executes one of three access methods. Accessing a stored procedure is one of those retrieval methods. The property **NamedStoredProceduresRateBll.config.filePath** allows a stored procedure to be used to access rates. The property identifies an XML file that is parsed for the rate group's description, the name of a stored procedure that returns a rate and the name of a stored procedure that returns a rate array. If the named stored procedure does not exist, an exception is returned to the system log. Execution will continue by using the default access to OIPA's rate data tables. During execution, OIPA passes parameter information from the RATE or RATEARRAY configuration to the designated stored procedure and expects an appropriate return. If the appropriate result is not returned, an exception will be thrown.

Refer to the 10.1.2.0 Documentation library on [OTN](#) for a complete explanation of this property.



Modifications to the System Properties file requires a restart of the application.

Configuration Steps

1. Define [rate table](#). Rate tables can be user-defined tables stored in the AsRate and AsRateGroup database tables. Alternatively, they can be custom tables stored internally on OIPA or externally and accessed by stored procedures.
2. Make sure the System Properties file has the stored procedure property **NamedStoredProceduresRateBll.config.filePath** properly defined.



RATE and RATEARRAY can be used with stored procedures and/or extension framework.

- 3.
3. Create the stored procedure. An example of each type is provided at the bottom of this page.
4. Create an XML file. An example of the XML is provided at the bottom of this page and names the file rateRetrieverProcedures.xml.
5. Modify or add the rateRetriever property in the System Properties document. In the example it should read as follows:
`#NamedStoredProceduresRateBll.config.filePath=/opt/Oracle-SIT/conf/rateRetrieverProcedures.xml`

Execution errors

If a stored procedure is not found when OIPA attempts to process rates and the System Properties file indicates that a stored procedure should exist, then an error will be logged and processing will continue using the [default method](#).

If a bad parameter is passed, then an exception will be generated.

Retrieve Rate directly through Stored Procedure XML

If custom tables are set up outside of Oracle's OIPA data structure, a service request must be created to allow Oracle's integration team to help facilitate the process.

Stored Procedure Examples

Rate

```
CREATE OR REPLACE PROCEDURE
OIPA.GETTERM20PREMIUMRATE
(v_Gender IN VARCHAR2 DEFAULT NULL,
v_Tobacco IN VARCHAR2 DEFAULT NULL,
v_Risk IN VARCHAR2 DEFAULT NULL,
v_IssueAge IN NUMBER DEFAULT NULL,
v_Band IN VARCHAR2 DEFAULT NULL,
v_Duration IN NUMBER DEFAULT NULL,
refCursor1 OUT sys_refcursor)
AS
BEGIN
  OPEN refCursor1 FOR
  SELECT Rate FROM AsRate
  WHERE
RateDescription='Term_20_Premium'
AND Criteria1=v_Gender
  AND Criteria2=v_Tobacco
  AND Criteria3=v_Risk
  AND Criteria4=v_IssueAge
  AND Criteria5=v_Band
  AND IntegerCriteria=v_Duration;
END;
```

Rate Stored Procedure Example

RateArray

```

CREATE OR REPLACE PROCEDURE
OIPA.GETTERM20PREMIUMRATEARRAY
(
  v_Gender IN VARCHAR2 DEFAULT NULL,
  v_Tobacco IN VARCHAR2 DEFAULT NULL,
  v_Risk IN VARCHAR2 DEFAULT NULL,
  v_IssueAge IN NUMBER DEFAULT NULL,
  v_Band IN VARCHAR2 DEFAULT NULL,
  v_Duration IN NUMBER DEFAULT NULL,
  v_startIndex IN NUMBER DEFAULT 10,
  v_endIndex IN NUMBER DEFAULT 30,
  refCursor1 OUT sys_refcursor)
AS
BEGIN
  OPEN refCursor1 FOR
  SELECT Rate, IntegerCriteria FROM AsRate
  WHERE RateDescription =
'Term_20_Premium'
  AND Criteria1=v_Gender
  AND Criteria2=v_Tobacco
  AND Criteria3=v_Risk
  AND Criteria4=v_IssueAge
  AND Criteria5=v_Band
  AND IntegerCriteria >=v_startIndex
  AND IntegerCriteria <=v_endIndex;
END;

```

Rate Array Stored Procedure Example

Rate Retriever XML Example

```

rateRetrieverProcedures.xml placed in ../opt/Environments/QA-Oracle-SIT/conf/rateRetrieverProcedures.xml

<rate-retrievers>
  <rate-retriever>
    <rate-group-description>Term_20_Premium</rate-group-description>
    <rate-procedure-name>getTerm20PremiumRate</rate-procedure-name>
    <ratearray-procedure-name>getTerm20PremiumRateArray</ratearray-procedure-
name>
  </rate-retriever>
</rate-retrievers>

```

Rate Retriever XML

Retrieve Rates Using Extension Framework

When OIPA encounters a [RATE](#) or [RATEARRAY](#) math variable type, it executes one of three access methods. Accessing rates through an extension framework is one of those retrieval methods. The property **rateRetriever.className** allows an extension to access rates. The property identifies the extension by its fully qualified class name. OIPA passes parameter information to the designated extension. The extension may return one rate or multiple rates as a result of the RateGroup's structure and the parameters provided by the RATE or RATEARRAY math variables. If the appropriate result is not returned, an exception will be thrown.

Refer to the 10.1.2.0 Documentation library on [OTN](#) for a complete explanation of this property.



Modifications to the System Properties file requires a restart of the application.

Configuration Steps

1. Define [rate table](#). Rate tables can be user-defined tables stored in the AsRate and AsRateGroup database tables. Alternatively, they can be custom tables stored internally on OIPA or externally and accessed by stored procedures.
2. Make sure the System Properties file has the stored procedure property **rateRetriever.className** properly defined.



RATE and RATEARRAY can be used with stored procedures and/or extension framework.

- 3.
3. Use a fully qualified class name.
4. Define Input and Output parameters.

Execution errors

If a class name does not exist when OIPA attempts to process rates, then an error will be logged and processing will continue using the [default method](#).

If a bad parameter is passed, then an exception will be generated.

Product Overview

OIPA allows plans to be arranged together into Products for easy navigation and selection. Rules can be overridden at the plan level as well as the Product level.

When plans are grouped, they fall into two categories: Parents and Children. Parent plans are the grouping entities. The plans that typically exhibit the features of the insurance products are the Child plans. However, Parent plans may also define general features of insurance products that the Child plans may or may not override.

Activating Products

Products are controlled through an application setting. They are not controlled by configuration. The build manager will determine if Products are used by controlling the setting in the Web Application Utility. If Products are turned on, then a Product field will appear on the following screens in OIPA:

- Policy screen
- Create New Policy screen
- Unit Values screen
- Disbursement Search screen



Transactions and segments are not supported at the Product level.

Plan and Product Database Tables

There are three database tables that store plan information.

1. **AsPlan**: stores the information entered when a new plan is created.
2. **AsPlanFields**: stores values for the plan. These values can be edited through the [Plan Data](#) folder in the Main Explorer.
3. **AsPlanGroup**: stores GUIDs of the parent and child plans.

Security

Security is available at the child plan level. It is not available at the Product level.

Deleting Products

A Product can be deleted if all child plans are removed from the Product and placed in another Product. Once all child plans are removed from the Product, the user may right-click on the Product and select the delete option.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

Calculate Rules

Calculate rules are business rules that calculate various segment and/or policy values necessary for a policy. Calculate rules consist of an input math and calculation section, MathVariable validation and a mapping output section.

The math section pulls in any necessary input values and performs required calculations.

The output section of a Calculate General rule includes a mapping section that can write calculated values to AsPolicyField, AsSegmentField, AsClientField, and AsRoleField.

- When calculating a segment the segment fields should be mapped to the segment being calculated.
- Any field included in the output mapping must be defined in the destination screen rule such as the [PolicyScreen](#) or Segment business rule.



A calculate button or a save button must be configured on the desired screen with the RULE attribute equaling the name of the Calculate rule to be invoked.

Validations

[Validations](#) can be configured to validate any math variables right after they are processed. Error messages can be configured within the Validations section to provide the user with information in the event that a validation fails.

A math variable (from the MathVariables section) or Segment Field substitution may be made in the validation message. If an actual integer value from a math variable or segment field is needed, it must be surrounded by \$\$\$\$. If a Segment field name is used in the Calculate Segment rule, the prefix 'Segment:' is required in the substitution.

The only supported data types are 'TEXT', 'INTEGER', 'CURRENCY', 'DATE' or 'DECIMAL'. Therefore any other data types will need to be converted to TEXT, INTEGER or DECIMAL before being passed into the validation message. Special characters such as '\$' or '%' need to be added in the validation message and will not be included in the substitution.

Validation Examples

The following example shows the error message configuration a user would see if the base issue age is greater than the maximum issue age on a policy.

```
<Expression TYPE="ErrorOnTrue" MESSAGE="Issue Age exceeds the maximum age of 65 (60 if in New York State). Segment will not calculate.">BaseIssueAge &gt; MaximumIssueAge</Expression>
```

If an actual integer value from a Segment Field or MathVariable name BaseIssueAge was needed for the validation message, then the configuration could be constructed as:

```
<Expression TYPE="ErrorOnTrue" MESSAGE="Issue Age of $$$Segment:BaseIssueAge$$$ exceeds the maximum age of 65 (60 if in New York State). Segment will not calculate.">BaseIssueAge &gt; MaximumIssueAge</Expression>
```


RoleShading

The Role Shading feature enables the display of role status in the [Role screen](#) in color. If the company is configured with the [RoleShading](#) feature enabled (in a company override of the CompanyCosmetics rule), then the role view will display each role in appropriate color and shading. The color of the text in the role listing matches the role status.

- **Black**— Active (01)
- **Gray** — Inactive (98)
- **Gray** — Deleted (99)

Create Function Override

After a [global function rule is created](#), it can be overridden at a variety of levels. The override levels in order from highest to lowest level are listed below:

- Company
- Product (if Product functionality is turned on)
- Plan
- Transaction
- State (this level can only be used in conjunction with another override level.)
- Fund (this level can only be used if funds exist on the Plan chosen for the override.)

Steps to Create a Function Override

1. Navigate to **Global Explorer | Business Rules | Functions**.
2. Double-click the Functions folder to reveal a list of existing function rules.
3. Right-click on the name of the function rule to override and click the option to override the function rule.
4. Click **Next**, as the name of the override must be the same as the original function rule.
5. Select the Company where the function rule override will apply. Selecting the Company will enable the available override levels for plans, transactions, states and funds associated with that particular company.
6. Select any additional override levels.
7. Click **Finish**. The new function rule override will display in the Configuration Area.
8. Configure the override using the steps in the section below.

Steps to Configure a Function Override

1. Make sure the Function is open in the Configuration Area. If not, double-click the .xml file to open it.
2. Click the Parameters tab.

General **Parameters** Math Xml Source Debug

Name:

Return:

Data Type:

Is Array
 Yes No

Input Parameters

Name	Data Type	Is Array
pEffectiveDate	DATE	<input type="radio"/> Yes <input checked="" type="radio"/> No
pPremiumPaymentArray	DECIMAL	<input checked="" type="radio"/> Yes <input type="radio"/> No
pPremiumDateArray	DATE	<input checked="" type="radio"/> Yes <input type="radio"/> No
pSurrenderAmount	DECIMAL	<input type="radio"/> Yes <input checked="" type="radio"/> No
pFreeAmountPercent	DECIMAL	<input type="radio"/> Yes <input checked="" type="radio"/> No
pMaxSurrenderYears	INTEGER	<input type="radio"/> Yes <input checked="" type="radio"/> No

Enter Input and Output Parameters

Output Parameters

Name	Data Type	Is Array
oFreeAmount	DECIMAL	<input type="radio"/> Yes <input checked="" type="radio"/> No
oPremiumsOutsideOfSurrenderPeriod	DECIMAL	<input type="radio"/> Yes <input checked="" type="radio"/> No

Function Parameters Tab

3. Enter the name of the function.
4. Enter the return value. It is a best practice to begin the value with the letter *r*.
5. Select the Data type.
6. Click the radio button for **Yes** or **No** if function is an array.
7. Click **Add** and click in the blank field to enter the input parameters.
 - input parameter name (best practice is to begin with the letter *p*)
 - select the data type from the drop down box.
 - specify whether it is an array.
8. Click **Add** and click in the blank field to enter the output parameters.
 - output parameter name (best practice is to begin with the letter

- o).
- select the data type from the drop down box.
 - specify whether it is an array.
9. Click the Math tab and configure the math section. Use all functionality available in the MathVariables section of Math with the exception of LOG.



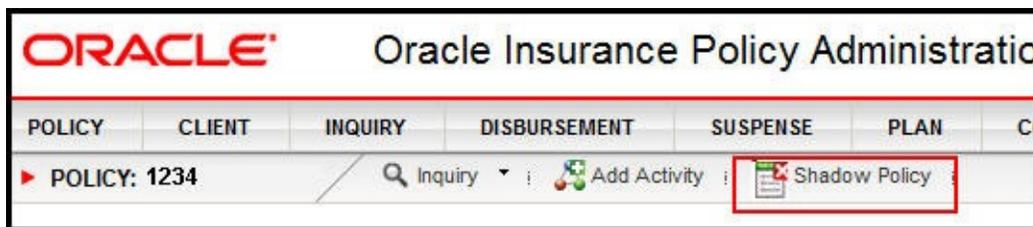
Refer to the [Math Pane](#) section for additional information on configuring in the Math Pane.

10. Check-in the XML file to save your changes to the database.

The Rules Palette validates that the signature of the function override matches that of the global version before the override may be checked in. This is to enforce that the same signature used by the global function is applied across all overrides.

Shadow Policy Pane

The OIPA application allows users to shadow (soft delete) policy records in certain defined statuses. Policy status codes that represent a shadowed status are configurable in the Policy screen business rule. When configured, a **Shadow Policy** link on the Policy screen in OIPA is available for users with the proper security privileges. This button allows the user to shadow a policy based on the current status of the policy. This is typically used to remove pending policies added in error.



Shadow Policy Link on Secondary Menu in OIPA

Shadow Policy Pane

The Shadow Policy pane in the PolicyScreen business rule visual editor provides a method for visually configuring shadow criteria. Configuration can also be added directly in the XML Source pane. There are three main sections to the Shadow Policy pane: the Shadow Status Code drop down box, the Allow Shadow section and the Validations section.

- **Shadow Status Code:** This drop down box contains the policy status codes that are listed within the <ShadowPolicyStatus> element of the [CompanyCosmetics business rule](#). The status that is selected here is the status that will be applied to the policy once it is shadowed (deleted). If this section is not in the CompanyCosmetics rule or if the rule does not exist, then no options will populate this box and the Shadow Policy section will not be included in the Policy screen business rule configuration.
- **Allow Shadow:** The options in the Available box of the Allow Shadow section are pulled from the AsCodeStatus table. Any status from this list can be moved to the Selected box. Policies in OIPA that are in the statuses listed in the Selected box are the only policies that will have the Shadow Policy link on the Secondary menu. If a policy is not in one of the Selected statuses, then no link is provided.
- **Validations:** OIPA will validate that the items in the Selected box in the Validations section do not exist on a policy that is being shadowed (deleted). If a policy is being shadowed with activities, disbursements, roles or segments, and the Validation section has any of those items selected, then OIPA will not allow the policy to be deleted.

General Fields Buttons Events Roles **Shadow Policy** Xml Source

Shadow Status Code :

Allow Shadow

Available		Selected
<div style="border: 1px solid red; padding: 2px;">Options are pulled from CompanyCosmetics rule</div> <div style="border: 1px solid red; padding: 2px; margin-top: 10px;">This box is populated from AsCodeStatus</div> <ul style="list-style-type: none">Active - FreeLook PeriodActiveAnnuitizedApprovedCancelledDeath - AnnuitizedDeath - NotifiedDeath Claim Pending - AnnuitizedDefault Segment	<input type="button" value=" >>"/> <input type="button" value=" <<"/>	<ul style="list-style-type: none">Closed

Validations

Available		Selected
<ul style="list-style-type: none">ActivitiesDisbursementsRolesSegments	<input type="button" value=" >>"/> <input type="button" value=" <<"/>	

Shadow Policy Pane in PolicyScreen Rule Visual Editor

Steps to Visually Configure Shadow Criteria

1. Check-out the PolicyScreen rule for the plan needed. If an override does not exist for the plan, [create the company override](#), then continue with step 2.
2. Click the **Shadow Policy** pane.
3. Select an option from the **Shadow Status Code** drop down box. If no options are available, add them to the [CompanyCosmetics business rule](#), then select the appropriate option from the drop down box when it populates.
4. In the **Allow Shadow** section, click an available status in the Available box then click the right facing arrow to move it to the Selected box.
5. Move any unwanted statuses out of the Selected box by clicking the left facing arrow. This will move the status back into the Available box.
6. In the **Validations** section, select the criteria that must **not** exist on a policy when it is shadowed then click the right facing arrow to move the criteria to the Selected box.
7. Move any unwanted criteria out of the Selected box by clicking the left facing arrow. This will move the criteria back into the Available box.
8. Check-in the rule to save the changes to the database.

DisbursementApprovalScreen

The DisbursementApprovalScreen provides the option to approve pending disbursements after they are created and processed as activities in OIPA.

Search Requirements

Search fields that are configured on the Disbursement Approval screen must define the group that is being searched on. The available groups are DisbursementApproval, Disbursement, DisbursementField, Activity, ActivityField, Transaction, Role, RoleField, Policy, PolicyField, Client and ClientField. The groups will support search on all columns in the corresponding table using the column name as the field name. For field tables like ActivityField, PolicyField, ClientField and DisbursementField, a search can be performed on the fieldname.

Totaling Currency Amounts

The Table section of the Disbursement Approval screen defines the columns that will display in the list of results when a search is performed. Configuration is supported to total the DisbursementAmount column. Currencies must be the same type. If they are mixed, then the amounts will not total.

 Please see the **XML Configuration Guide** topic in this help system for a list of all elements, attributes and values needed for configuration. View **Business Rules | Screen Rules | DisbursementApprovalScreen**.

General Pane

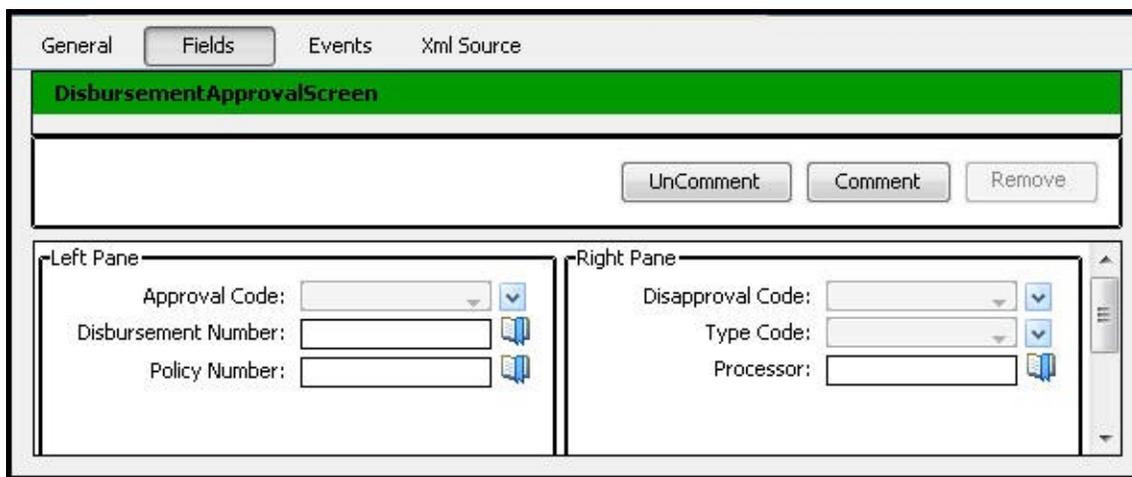
The DisbursementApprovalScreen General pane displays the name of the rule, the properties of the screen and their associated values.



General Pane

Fields Pane

The Fields pane is used to create fields that will display on the DisbursementApprovalScreen. The top section of the visual configuration area is for fixed fields. Drag and drop the available fixed fields from the Palette window on the right side of the screen. Custom fields cannot be created for this section, but the display names of fixed fields can be changed as needed. Configure dynamic fields in the section under fixed fields. The functionality is exactly the same as the configuration for other fields panes. Please see the [Fields Pane](#) for more information.



Fields Pane Visual Editing in Rules Palette for DisbursementApprovalScreen

Events Pane

This pane is used to create validations that will display on the screen. The functionality is exactly the same as described in the transaction section. Please see [Events](#) for more information.

XML Source Pane

Configuration for this rule is done in XML via the XML Source pane. The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the [XML Source Editing](#) section.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

DisbursementSearchScreen

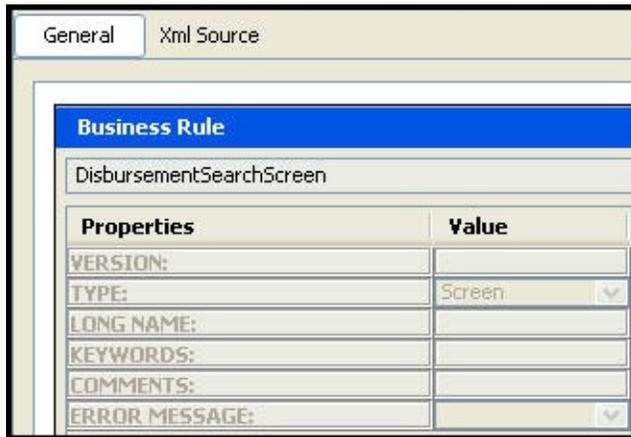
This business rule is used to configure the dynamic fields in the DisbursementSearchScreen to allow the user to search for disbursement records that match the specified criteria. If the DisbursementSearchScreen business rule is not used the Above the line Fields (FixedFields) will be displayed by default and used for searches. For example Company, Plan, Start Date and End Date will be displayed.

Note: DisbursementScreen and DisbursementSearchScreen business rules together constitute the Disbursement screen. Configuration for the Disbursement Search section is done in the DisbursementSearchScreen business rule; whereas the configuration for the Disbursement details section is done in the DisbursementScreen business rule.

 Please see the **XML Configuration Guide** topic under **Help** on the Main Menu for a list of all elements, attributes and values needed for configuration. View **Business Rules | Screen Rules | DisbursementSearchScreen**.

General Pane

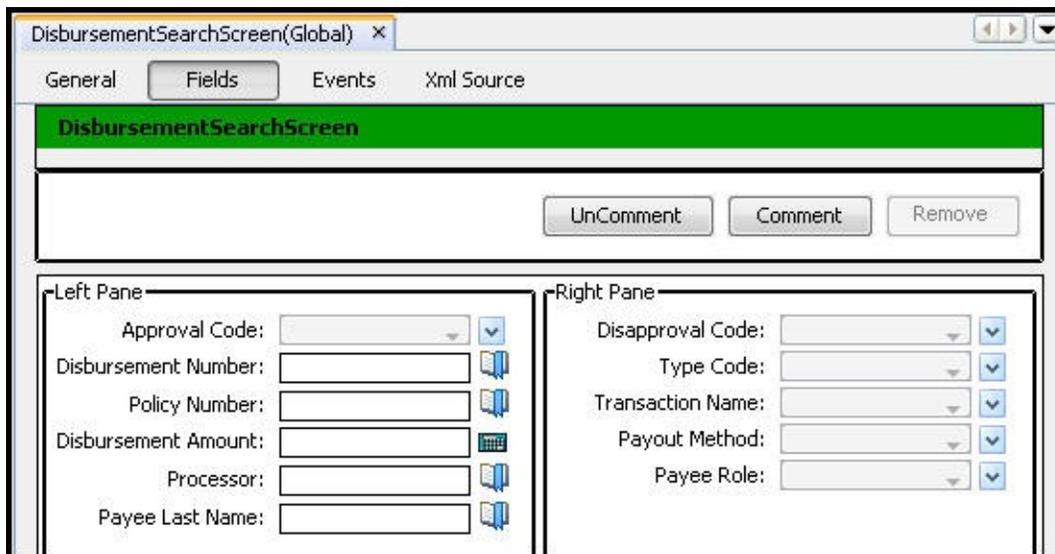
The Disbursement Search screen General pane displays the name of the rule, the properties of the Disbursement Search screen and their associated values.



General Pane of DisbursementSearchScreen

Fields Pane

The Fields pane is used to create fields that will display on the DisbursementSearchScreen. The top section of the visual configuration area is for fixed fields. Drag and drop the available fixed fields from the Palette window on the right side of the screen. Custom fields cannot be created for this section, but the display names of fixed fields can be changed as needed. Configure dynamic fields in the section under fixed fields. The functionality is exactly the same as the configuration for other fields panes. Please see the [Fields Pane](#) for more information.



Fields Pane Visual Editing in Rules Palette for DisbursementSearchScreen

Events Pane

This pane is used to create validations that will display on the screen. The functionality is exactly the same as described in the transaction section. Please see [Events](#) for more information.

XML Source Pane

The XML Source pane includes XML Editor functionality that enhances the configuration experience. More detailed information on configuration using the XML Source pane can be found in the [XML Source Editing](#) section.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

CompanyCosmetics Business Rule

The **CompanyCosmetics** business rule:

- Identifies the background to display on all screens except the login screen.
- Specifies statuses that can be assigned to a policy when it is deleted in OIPA.
- Identifies the role code that should get displayed in the policy header section for a individual policy record and group policy record.
- Enables or disables the [ActivityShading feature](#).
- Enables or disables the Role Shading feature.
- Enables or disables the Requirement Shading feature.

Steps to Configure the CompanyCosmetics Rule

1. In the **Global Rules Explorer**, browse to **Business Rules | System** and right-click on **CompanyCosmetics**.
2. Select **NewCompanyCosmeticsOverride**.
3. In the Configuration area, click on the **XML Source** tab. The XML source is displayed, with only the start and end tags for the CompanyCosmetics rule.
4. If a background is defined in this rule, add a sub-element `<CompanyCosmetics>`.
5. Under the sub-element `<CompanyCosmetics>` add another sub-element: `<Tile>`. The name of the file that contains the background for all screens except the login screen should appear between the opening and closing tags of this element.
6. If shadowing is allowed on the policy, then `<ShadowPolicyStatus>` should be added as a sub-element to the parent tag, at the same level as the `<CompanyCosmetics>` sub-heading.
7. The `<ShadowStatusCode>` element should appear as a sub-heading to `<ShadowPolicyStatus>` and should contain all valid status codes that can be assigned to a policy once it is deleted. Status codes are defined in the AsCodeStatus table. These defined statuses will populate the Shadow Status Code drop down box on the Policy Screen Shadow Policy visual editing pane.
8. If any additional `<ShadowStatusCode>` elements are needed, then add one for each code.
9. Use the `<Substitutions>` element along with the `<Owner>` and/or `<PrimaryMember>` element to define the role code that should get displayed in the policy header section.
10. If Activity Shading is required, add the `<ActivityShading ENABLE="Yes">` element after the `</CompanyCosmetics>` sub-element, the `</ShadowPolicyStatus>` element or the `</RoleShading>` element. The `<ActivityShading>` element must not be a sub-element of the `<CompanyCosmetics>` sub-element.
11. If Role Shading is required, add the `<RoleShading ENABLE="Yes">` element after the `</CompanyCosmetics>` sub-element, the

</ShadowPolicyStatus> element or the </ActivityShading> element. The <RoleShading> element must not be a sub-element of the <CompanyCosmetics> sub-element.

12. If Requirement Shading is required, add the <PolicyRequirementShading> element. Also add the <Shading> sub-element with a value specifying the color to shade the requirement, as well as a STATUS attribute of the <Shading> element specifying the status to which the shading should be applied.
13. Save and check in the rule.

Role Shading

The Role Shading feature enables the display of role status in the [Role screen](#) in color. If the company is configured with the Role Shading feature enabled (in a company override of the CompanyCosmetics rule), then the role view will display each role in appropriate color and shading. The color of the text in the role listing matches the role status.

- **Black**— Active (01)
- **Gray** — Inactive (98)
- **Gray** — Deleted (99)

Changing OIPA's Background Image

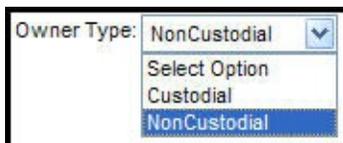
The CompanyCosmetics business rule allows a configurator to change the background image that displays in OIPA. To do so:

1. Unzip the OIPA .war file.
2. Within the .war file, navigate to **Themes | Oracle | images**.
3. Copy the desired image into the folder.
4. Close the program used to unzip the .war file and save the changes if prompted.
5. Open the Rules Palette and check out the Company Cosmetics business rule.
6. Change the value of the <Tile> element to the new image's filename.
7. Deploy the OIPA .war file.

Field Options

The <Field> element has two sub-elements that use an <Options> tag. Both <Query> and <Calculate> can use <Options> to populate fixed values in a [Combo](#) box. Multiple [radio buttons](#) can also be defined using <Options>.

Usually Field Options are configured on screens and in transactions within the <Field> configuration. There are two parts required for configuring <Options>. First, an <OptionValue> must be defined. This is the value that is stored in the database in the TextValue column. Then, an <OptionText> must be defined. This is the text a user will see displayed in the Combo box or radio button in OIPA. Option Text is stored in the OptionText column.



Combo Box Displaying Options

```
<Field>
  <Name>OwnerType</Name>
  <Display>Owner Type</Display>
  <DataType>Combo</DataType>
  <Query TYPE="FIXED">
    <Options>
      <Option>
        <OptionValue>01</OptionValue>
        <OptionText>Custodial</OptionText>
      </Option>
      <Option>
        <OptionValue>02</OptionValue>
        <OptionText>NonCustodial</OptionText>
      </Option>
    </Options>
  </Query>
  <DefaultValue>02</DefaultValue>
</Field>
```

Field Option XML Example

Configuration Consideration

Saving the option text in the database saves processing time because it allows the screen to display the option text from the database without having to execute the configuration. While it saves processing time, it also makes it impossible for a rule to fully update the value of a field on a screen that used option text.

This occurs because a rule (CopyTo, for example) can only update one value in the database. If it updates the option text, then the user will see a correctly updated field in OIPA, but the actual value will be unchanged. If it updates the option value, then the correct value will be stored in the database, but the option text the user sees will remain unchanged, giving the impression that the value is also unchanged.



Do not use field options if a rule needs to update a screen field value.

It is also important to keep in mind any impact OnChange events could have on configuration if a combo box is used. OnChange events can cause combo boxes to be reloaded.

Screen Math Considerations

Screen Math is a bit different from typical transaction math, and therefore, configuration can vary. Below are some important configuration considerations to keep in mind when working in screen math.

Math and Currency Considerations

Screen Math performs arithmetic and comparison operations on currency values. Some common operations are comparison of math variables in the Math section, addition for like currency codes and multiplication for like currency codes.

Keep in mind the following points when working with currency in ScreenMath.

- The Currency Code of a field can be found using the syntax of "FieldName:CurrencyCode".
- The Currency Code of a math variable can be found using the function "GetCurrencyCode ()". Comparisons of named fields or math variables will use the attached currency codes in addition to their numeric values. The associated currency code values must be the same or the system will throw a stack trace indicating the currency codes are incompatible.
- Type = FUNCTION supports both ToCurrency(), which allows a decimal or integer value to be given a currency code and "GetCurrencyCode()", which returns the currency code of a math variable.
- Type = FUNCTIONCALL parameters accept input Money/Currency values.

Suspense Considerations

- Screen Math supports the SUSPENSEFIELD math variable. Configuration will only have access to the suspense fields for the single suspense item from the activity details of that activity being processed. Suspense multifields will not be available using the SUSPENSEFIELD math variable type.
- Suspense fields can be accessed in both local and global screen math.
- Configuring a math variable with a type equal to SUSPENSEFIELD in a transaction that does not use suspense will result in a null pointer exception. As a result the SUSPENSEFIELD math variable should be contained within an IsEmpty check to prevent a system error.
- Action configuration can access the SUSPENSEFIELD screen math variables configured in screen math in order to perform validations on the suspense data for the activity.
- The value of the SUSPENSEFIELD math variable should be the name of the suspense field to be returned from the activity detail. The following suspense fields are available in screen math.
 - SuspenseGUID
 - SuspenseNumber
 - TypeCode
 - StatusCode
 - Amount
 - AttachedAmount
 - CompanyGUID
 - PolicyNumber
 - EffectiveDate
 - EffectiveFromDate
 - EffectiveToDate
 - ClientNumber
 - FirstName
 - LastName

- AccountNumber
- BankName
- BankNumber
- CheckNumber
- BatchNumber
- CurrencyCode
- All dynamic suspense fields configured for the suspense item.

Copyright © 2009, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)

Address Pane

The Address pane is used to visually configure address information for transactions. This pane contains two drop-down boxes (Source Address and Default Address Type), which represent each of the two available <Address> element attributes. The **Address Types** selector windows are available to visually identify available address types. They are identified in the XML with the <AddressTypes>sub-elements.

When the <Address> element is added to a transaction, it pulls information from the AddressScreen business rule in order to display Address fields on the Activity Address tab. Once the transaction that contains the <Address> element is processed, records are written into the address set of tables: AsAddress and AsAddressField.

The screenshot shows a software interface for configuring a transaction. At the top, there is a tabbed menu with options: General, Allocations, Address (highlighted with a red box), Fields, Events, Math, Assignment, Spawn, and Xml Source. Below the menu is a 'CopyBook' section. The main area is titled 'Address' and contains two dropdown menus: 'Source Address:' and 'Default Address Type:'. Below these is an 'Address Types:' section with a list of address types: Billing, EFT, Home, Insurance Company, Internal Account, Mailing, and Residence. To the right of this list are two buttons: '>>' and '<<'. A 'Clear' button is located to the right of the 'Address Types:' label.

Address Pane of a Transaction

Steps to Configure Addresses

1. Navigate to a transaction or create a new transaction.
2. **Check-out** the transaction's XML file.
3. Open the **Address** Pane.
4. If a source address will be used, select a source address from the **Source Address** drop down box. Available values for this attribute are Activity fields capable of reflecting GUIDs (i.e. those with Text, Combo and Radio data types) configured within the transaction. This drop-down box is pre-populated with all such fields. When a non-blank selection is made in this box, the <Address> element is automatically added above the <Fields> section in the XML Source pane, with the COPYSOURCEADDRESSGUID attribute set equal to a configured transaction field (including "Activity:" prefix). Using this drop down box will disable the Default Address Type drop down box.
5. If a default address type needs to be identified, then use the **Default Address Type** drop-down box to select the address type.
6. If a source address is NOT used, then select an address type from the left window and move it to the right window to designate it as an available address type. Available address types are designated (and will also be available in the Default Address Type drop-down box) via address role codes in the AsCode table, but are limited to the address types configured within the AddressScreen business rule.
7. Check in the transaction to save the changes to the database.

Link Program to a Plan

A program can be linked to a plan. When this occurs, a PolicyProgram rule is created. This rule identifies the policy statuses when programs can be added or updated.



Refer to the XML Configuration Guide in the Help menu for a complete explanation of all element and attributes mentioned in the steps below.

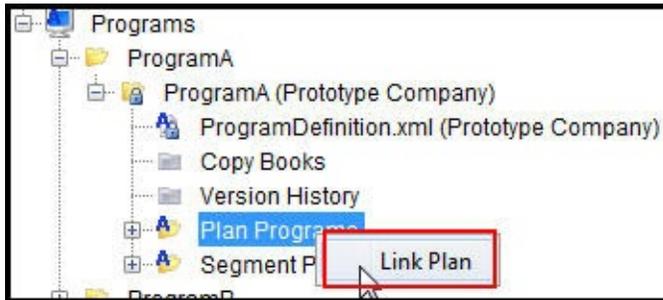
Steps to Link a Program to a Plan or Segment

1. Navigate to the **Global Explorer** tab.
2. Open the environment folder and then open the **Programs** node.
3. Right-click the Plan Programs folder and click **Link Plan**.
4. Select the Subsidiary Company from the drop down box that the plan is associated with. The three fields above are automatically populated and cannot be changed.
5. Select the plan from the Plan drop down box.
6. Click **Finish**.
7. Configure the **PolicyPrograms.xml** file.
 - Add an <Eligibility> section that contains the status codes a policy must be in to create or update a program. The <PolicyStatusCode> element holds the values of the policy status codes, which are in AsCodeStatus.
 - Define the operations that can be performed on the program when the policy is in the appropriate status. The **Add** operation allows the Add button to appear when the policy is in the proper status. The **Maintain** operation allows the Save button to appear when the policy is in the proper status. The **Both** operation allows both the Save and Add button to appear when the policy is in the proper status.



Security must also be assigned to the ProgramScreen through

Plan Security. If this rule provides access to the Add and Save buttons, but security has not be granted, then the buttons will not be available to the user in OIPA.



Link Plan Option on Plan Program Node

Link Program to a Segment

A program can be linked to a segment. When this occurs, a SegmentProgram rule is created. This rule identifies the policy statuses when programs can be added or updated. The PolicyStatusCode values can be found in **AsCodeStatus**.



Refer to the XML Configuration Guide in the Help menu for a complete explanation of all element and attributes mentioned in the steps below.

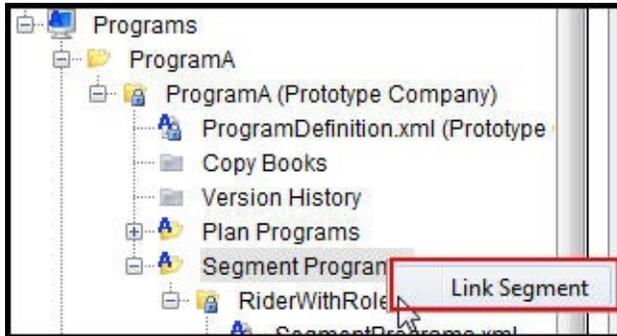
Steps to Link a Program to a Segment

1. Navigate to the **Global Explorer** tab.
2. Open the environment folder and then open the **Programs** node.
3. Right-click the **Segment Programs** folder and click **Link Segment**.
4. Select a Subsidiary Company from the drop down box.
5. Select a plan from the drop down box.
6. Select a segment from the drop down box.
7. Click **Finish**.
8. Configure the **SegmentPrograms.xml** file.
 - Add an <Eligibility> section that contains the status codes a policy must be in to create or update a program. The <PolicyStatusCode> element holds the values of the policy status codes, which are in AsCodeStatus.
 - Define the operations that can be performed on the program when the policy is in the appropriate status. The **Add** operation allows the Add button to appear when the policy is in the proper status. The **Maintain** operation allows the Save button to appear when the policy is in the proper status. The **Both** operation allows both the Save and Add button to appear when the policy is in the proper status.



Security must also be assigned to the ProgramScreen through

Plan Security. If this rule provides access to the Add and Save buttons, but security has not be granted, then the buttons will not be available to the user in OIPA.



Link Segment Option for Programs

Lateral Funds

Lateral funds are used when a Unitized fund is offered with different variations. The Unitized [Parent fund](#) is always created first. Lateral funds are then created with Effective and Expiration dates and inherit their fund fields and plan fund statuses from their parents.

[Child funds](#) are another type of fund that can be associated with a parent fund. A parent fund can have a child fund or a lateral fund, but not both. Lateral funds and child funds are mutually exclusive.

High Level Steps to Generate Lateral Funds

1. Configure the [FundScreen](#) business to support lateral funds and any extra fields needed for the fund information.
2. Make sure [Parent funds](#) were created using the Fund wizard in Admin Explorer.
3. Use the LateralFunds.xml subnode to create the lateral funds. This subnode is located in the Admin Explorer in the Funds folder under the Parent fund designated as eligible to have associated Lateral funds.
4. Associate parent funds with plans.
5. Associate lateral funds with plans.
6. Set fund Parent fund status.

Step 1: Configure the FundScreen business rule

The FundScreen business rule needs to be configured to indicate to the system that lateral funds will be allowed for a particular parent fund. A <LateralFund> section is included at the beginning of the rule indicating whether lateral funds are supported.

The <ParentFundFields> section holds fields for additional information for the fund, which carry over to any child, lateral or benefit funds as appropriate.

1. In the **Global Explorer**, navigate to **Business Rules | Screen | FundScreen**.
2. Locate the **FundScreen** business rule in the Company Overrides folder or create a new override of this rule for the applicable company.
3. Configure the <LateralFunds> element.
 - Include the ALLOW="Yes" attribute.
 - Add a <Fund TYPE="[fundtypecode]">Yes</Fund> element to indicate the type of lateral fund supported.
4. Configure the <LateralFundFields> element.

- Configure the <Field> elements for all fields needed to capture lateral fund information.

Step 2: Create Parent funds via fund wizard

Create the parent fund that the lateral funds will be associated with. This can hold general information, but clients will never be enrolled into this as a fund.

1. Navigate to the **Admin Explorer**.
2. Expand the **Administration** folder.
3. Expand the **Funds** folder.
4. Right-click on the Company Level folder and select **Add New Fund**. The New Fund Wizard will appear.
5. Type the fund information in the fields provided. Each field is described below.
 - **Fund name:** The name of the fund being entered.
 - **Fund Type:** Select a fund typecode that was designated as lateral fund eligible in <Fund TYPE="[fund type code]">Yes</Fund> sub-element of <LateralFund> element.
 - **Currency code:** This is the currency of the fund, which defaults to the currency code of the Primary Company or System Properties. Populated by AsCurrency, which can be edited in the Currency editor in the Admin Explorer.
 - **Calendar code:** This is the calendar that the fund is traded on. Populated by AsCodeCalendar. An Override Offset checkbox is available to selected if the fund type is Unit Linking. If this box is checked, Fund Offset Days can be selected directly below.
 - **Fund Offset Days:** This drop-down box is only enabled if the Fund Type is Unit Linked and the Override Offset is checked so it does not apply to parent funds eligible for lateral funds.
6. Select **Finish**. If the Finish button is not enabled, required fund information may be missing. When complete, the Fund Detail pane opens for the new fund in the Configuration Area.

Step 3: Use the LateralFunds.xml subnode to create Lateral Funds

The LateralFunds.xml file is used to actually create the lateral funds. It takes the <LateralFundField> information from the FundScreen configuration, lists all possible permutations of field values and provides a default lateral fund name. Using the check boxes, select the lateral funds to create.

1. Navigate to the **Admin Explorer** and open the **Company | Unitized Parent Fund** folder.
2. Expand the **LateralFunds** folder.
3. Check out the LateralFunds (name of fund).xml file. The Editor will open. There are two sub panes: Fund Detail and Funds.
4. Select or enter a Relation field value. The default value is configured in the <DefaultValue> tag. If the <DefaultValue> tag is not configured, then blank is set as the default.
5. Click **Add**. This button is disabled until all relation fields have been entered. A Remove button is available to remove newly added Lateral Fund record. This button will be disabled until at least one new lateral fund record(s) is added and then selected in the Funds sub-pane. If multiple records are added, they may be removed one at a time by selecting the record and clicking **Remove**.
6. Enter a unique fund name.
7. Check-in the lateral funds subnode.

Step 4: Associate Funds with Plans

To associate lateral funds with a plan, use the Plan to Fund method. The parent fund must first be associated with the plan, before lateral funds can be associated. Navigate to the plan in the Funds node and make the associations there.



A lateral fund cannot be deleted once it is associated with a plan. In addition, any lateral fund that has been run through the valuation process cannot be deleted.

1. In the **Admin Explorer**, expand **Administration | Funds**.
2. Expand the **Primary Company Name | Subsidiary Companies | Subsidiary Company Name | Plans | Plan Name**.
3. Expand the Parent Funds folder.
4. Check out the Parent Fund subnode file.
5. Select the funds from the **Available Funds** box and select the arrow to move them to the **Attached Funds** box.
6. Check in the file.
7. Expand the Lateral Funds folder.
8. Right-click on the Lateral Funds xml subnode and select **Check out**.
9. Select the Lateral Fund Relation fields that are listed in the top box under Lateral Funds that should be available for the plan. Remember the Lateral Fund Relation Fields are the fields that were configured to define the type of class or band that the fund is going to be.
10. Select the **Apply Relation Filter**, which loads the lateral funds with matching relation field values selected in the Available Funds box. The list is an alphabetized multi-select list box of available lateral funds related to the parent funds associated to the plan.
11. Use the arrow buttons between the Available Funds and Attached Funds box to define the lateral funds that will be associated with the plan.
12. Check in when complete.

Step 5: Set Plan Fund and Plan Fund Status information

Plan fund information such as removal method, removal precedence and deposit level tracking are handled at the parent fund level. Plan fund status records are not created in the AsPlanFund table for lateral funds. All lateral funds adhere to the plan fund status information set for the

parent fund.

Editing Lateral Funds

Lateral funds can be edited at any time; however, they can only be deleted before they are associated with a plan. To edit a lateral fund, locate it in the Admin Explorer, in the **Fund | Company | Parent Fund** folder. Check out the lateralfund.xml file and make updates through the editor. Keep in mind the following information when editing lateral funds:

- lateral funds ordered by Effective Date (latest –to-earliest, top-to-bottom).
- characteristics that can be edited are name, related field(s), effective date, and expiration date, as long as the fund has not yet been valued.
- a new lateral fund with same set of relation fields' values cannot be added unless the “earlier” lateral fund has an expiration date.

Suspense Refund Number

Transactions that refund suspense monies can be configured to automatically generate a suspense refund number when processed in OIPA. This number is fully configurable, and is generated via a dynamic field with the [Identifier](#) <DataType>. Through the use of its <Parts><Part> sub-elements (with “SEQUENCE”, or “VALUE” attributes), the suspense refund number has the capability to contain Prefix, Sequential Value and Suffix components.

Saving the Suspense Refund Number

Once generated, the suspense refund number may be passed to a variety of destinations, such as screens, spawned activities, field tables, etc., through the use of available CopyTo..., MaintainSuspense, GenerateSuspense, AddSuspense and other such rules or elements.

Note: The [Suspense Refund Prototype](#) demonstrates how configuration is used to pass the automatically generated suspense refund number to screens in OIPA.

Using Suspense Refund Number in Searches

Suspense refund number fields may be used as search criteria in various search screens (Policy Search, Suspense Search, Disbursement Search, Client Search, etc...).

A field must be configured on any screen that needs to support a suspense refund number search. For example, the Disbursement and Suspense Search screens containing the following XML configuration will display a field that allows the user to search for suspense or disbursement records containing a specific suspense refund number.

```
<Field>
  <Name>SuspenseRefundNumber</Name>
  <Display>Suspense Refund Number</Display>
  <Group>DisbursementField</Group>
  <DataType>Text</DataType>
</Field>
```

Field for Suspense Refund Number

The screenshot shows a web form titled "DISBURSEMENT SEARCH CRITERIA". The form contains several search criteria fields: Company (dropdown), Plan (dropdown), Start Date (calendar icon), End Date (calendar icon), Approval Code (dropdown), Disapproval Code (dropdown), Disbursement Number (text), Type Code (dropdown), Policy Number (text), Transaction Name (dropdown), Disbursement Amount (text) with a USD label, Payout Method (dropdown), Processor (text), Payee Role (dropdown), and Payee Last Name (text). The "Suspense Refund Number" field is located at the bottom right of the form and is highlighted with a red rectangular border. A "Find" button is located at the bottom center of the form.

Suspense Refund Number Field in OIPA

Configure Point-In-Time Valuation

Prior to configuring Point-in-Time valuation, ensure that basic fund valuation is already fully working in the environment. Funds, allocations, and assignments must already be setup and configured before attempting to enable Point-in-Time valuation.

Point-in-Time valuation requires the configuration of two business rules:

- [PointinTimeValuation](#)
- [WriteValuationElements](#)

Point-in-Time valuation must also be enabled at the Plan level for each plan that will use its functionality. This can be configured in the Rules Palette through the [New Plan editor](#) by setting the Point-in-Time Valuation field to **Yes**. The action can be confirmed by checking the PointInTimeValuation column of the AsPlan table. The value of the column should be **Y**. As soon as policies have been added to a plan, the Point-in-Time valuation field cannot be switched to traditional valuation by setting the Point-in-Time valuation field to **No**.

In addition, each of the plan's funds should have deposit level tracking set to **Yes** or **No**. This can be configured in the Rules Palette in the [New Fund editor](#) by setting the Deposit Level Tracking field. It can be confirmed in the database by examining the DepositLevelTracking column of the AsFund table. The value of the column must be **Y** or **N**.

Transitioning Existing Plans to Point-in-Time Valuation

If an existing plan transitions from Traditional valuation to Point-in-Time valuation, then a mixed methods indicator is used and the transition date determines how valuation records are written. This indicator can be found on the [Plan Maintenance editor](#). When the mixed valuation methods indicator is set to **Yes**, the transition date will determine how valuation should be written. When the mixed valuation methods indicator is set to **No** or null, any activity that is processed on the policy will write Point-in-Time valuation and no transition date will be required. This includes activities that are reprocessed, which previously calculated valuation using Period of Time.



The transition date is set by a utility when traditional valuation is converted to point in time valuation. To request this utility, go to My Oracle Support at <https://support.oracle.com> and file a service request. A support specialist will facilitate the process.

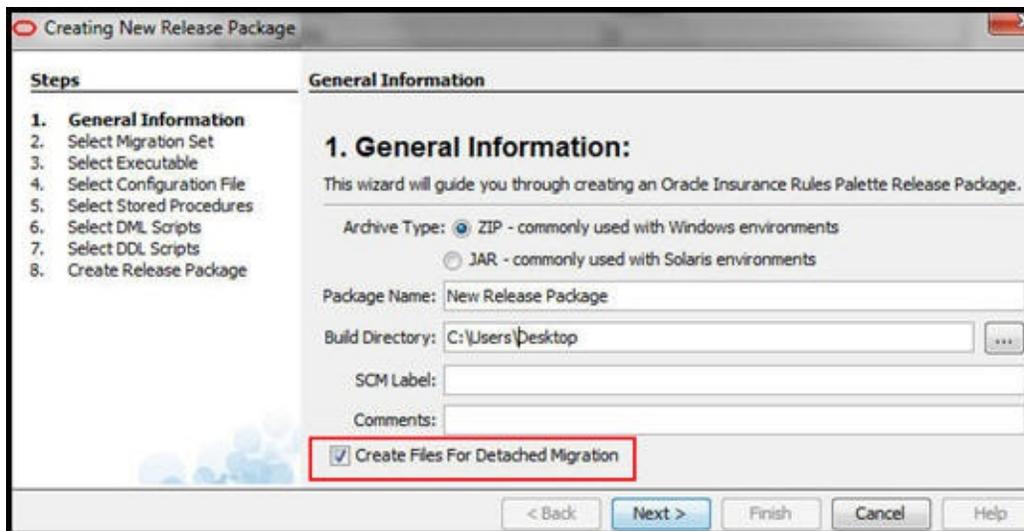
Once a plan transitions to Point-in-Time valuation, it cannot switch back to Traditional valuation.

Detached Migration

A detached migration begins in much the same way as a typical migration. The [configuration packages](#) are created and marked ready to migrate. They are then added to [migration sets](#).

When a release package is created, the first screen has a check box option to create files for detached migration. Clicking this box will allow the release package to migrate outside of the IVS environment of the source environment.

After the release package is created, it is [built](#) exactly as it is in a typical migration.



Detached Migration Checkbox On Release Package Wizard

Deploying a Detached Migration

The Rules Palette has a tool that manages the deployment of a detached migration. This tool can only be used in the target environment, after a release package has been built.

Steps to Deploy Detached Migration

1. Log out of the source environment and log into the target environment using a user ID and password with [detached migration security privileges](#).
2. Click **Tools | Detached Migration** from the Rules Palette main menu. The Detached Migration Wizard will open.
3. Type the location of the release package in the **Release Package** field. This is the package that was created, built and marked ready to promote in the source environment. A Browse button is also available to locate the package.
4. Type a name for the release package in the **Package Name** field. A package cannot already exist in the target environment with the same release package name. Each name must be unique. When the detached package is deployed in the Target environment, a new non-detached package is created and named here that can be used to migrate the items to another environment by regular migration.
5. Identify the location where the package will reside once it is built. A Browse button is available to help identify the location.
6. Click **Deploy**. If the deployment is successfully completed then a message will appear. You must log out of the environment and then log in again to view the deployed rules.

When the deployment is initiated, OIPA performs conflict validations on each rule prior to writing the rule to the database. If a conflict occurs, you will be presented with a warning message, along with action buttons to support user overrides or a cancel button to abort the detached migration.

Once the deployment is successfully completed, a confirmation message

is presented.

Trouble Shooting the Detached Migration

The IVSRELEASEPACKAGE table has a new column called DETACHEDFLAG. A value of Y indicates that the package was marked for detached migration.

The location of the detached migration files can be found by referencing the <BuildLocation> tags in the RELEASEPACKAGEMANIFEST column of the IVSRELEASEPACKAGE table.

The ReleaseManifest.xml file contains a list of all items included in the detached migration. Items are grouped by type and are separate by start and end tags that reflect the item type. For example, all business rules are grouped with a start and end tag named <Business Rules>.

Rate Set Worksheet

For each rate set, a new work sheet must be created. Each rate set must have its own rate effective date, active from date and active to date. A new RateGroupGUID will be generated for each combination of rate description, rate effective date and active from date. For any existing combination, the system will automatically pick up the existing RateGroupGUID and populate the data. If the same set of criteria exists in the AsRate table, then a warning message will be provided. If the warning is overridden, then the existing rate values will be changed to the new values.

Worksheet Definitions

- **Rate Activation Date:** The rate activation date for the rate set in the worksheet.
- **Active From Date:** The transaction From date for the rate set in the worksheet.
- **Active To Date:** The transaction To date for the rate set in the worksheet.
- **Criteria1 to Criteria10:** The criteria names for each of the criterion along with the criteria values for the specific rate set.

Example Legend

	A	B	C
1	Legend		
2	Group Name	CurrentCOIRate	
3	Table Type	Select	
4	Primary Index	Duration	
5	Primary Index Orientation	column	
6	Secondary Index	IssueAge	
7	Secondary Index Data Type	int	
8	Select Period/Duration	11	
9	Maximum Age	31	
10	Criteria1		
11	Criteria2		
12	Criteria3	Gender	
13	Criteria4		
14	Criteria5	UWCode	
15	Criteria6		
16	Criteria7		
17	Criteria8		
18	Criteria9	TobaccoBasis	
19	Criteria10		

Example Worksheet 1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	GoBack													
2	Rates must start in cell B10													
3														
4														
5				Criteria1	Criteria2	Criteria3	Criteria4	Criteria5	Criteria6	Criteria7	Criteria8	Criteria9	Criteria10	
6	Rate Activation Date	Active From Date	Active To Date		Gender			UWCode				TobaccoBasis		
7	1/1/2000	1/1/2000			01			01				01		
8														
9		20	21	22	23	24	25	26	27	28	29	30		
10	1													
11	2													
12	3													
13	4													
14	5													
15	6													
16	7													
17	8													
18	9													
19	10													
20	11													

Example Worksheet 2 (Note Criteria Value Change)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	GoBack												
2	Rates must start in cell B10												
3													
4													
5				Criteria1	Criteria2	Criteria3	Criteria4	Criteria5	Criteria6	Criteria7	Criteria8	Criteria9	Criteria10
6	Rate Activation Date	Active From Date	Active To Date			Gender		UWCode				TobaccoBasis	
7	1/1/2000	1/1/2000				01		01				02	
8													
9		20	21	22	23	24	25	26	27	28	29	30	
10	1												
11	2												
12	3												
13	4												
14	5												
15	6												
16	7												
17	8												
18	9												
19	10												
20	11												

Example Worksheet 3 (Note Date Range Change)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	GoBack												
2	Rates must start in cell B10												
3													
4													
5				Criteria1	Criteria2	Criteria3	Criteria4	Criteria5	Criteria6	Criteria7	Criteria8	Criteria9	Criteria10
6	Rate Activation Date	Active From Date	Active To Date			Gender		UWCode				TobaccoBasis	
7	4/1/2005	4/1/2005				01		01				01	
8													
9		20	21	22	23	24	25	26	27	28	29	30	
10	1												
11	2												
12	3												
13	4												
14	5												
15	6												
16	7												
17	8												
18	9												
19	10												
20	11												
21													