

CA Aion[®] Rule Manager

Product Guide

r11



This documentation and any related computer software help programs (hereinafter referred to as the "Documentation") are for your informational purposes only and are subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be used or disclosed by you except as may be permitted in a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2009 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Aion® Rule Manager (CA Aion Rule Manager)
- CA Aion® Business Rules Expert (CA Aion BRE)
- CA Embedded Entitlement Manager (CA EEM)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA Technologies product documentation, complete our short customer survey, which is available on the CA Support website at <http://ca.com/docs>.

Contents

| | |
|---|---------------|
| Chapter 1: Introduction | 9 |
| About CA Aion Rule Manager | 9 |
| Audience | 10 |
| Components | 10 |
| Chapter 2: Installing CA Aion Rule Manager | 13 |
| Installation Prerequisites | 13 |
| Installation Considerations | 14 |
| Install Over the Network | 14 |
| Authenticate Shared EEM Server | 14 |
| Access CA Aion Rule Manager Database | 14 |
| Gather Information on CA Aion Rule Manager | 15 |
| Installation Log Files | 20 |
| Install CA Aion Rule Manager | 21 |
| Silent Installation | 21 |
| Variables Used in Installer Response File | 23 |
| Sample Response File | 26 |
| Chapter 3: Using CA Aion Rule Manager | 29 |
| Access CA Aion Rule Manager | 29 |
| CA Aion Rule Manager Portlet | 30 |
| Folder Portlet | 31 |
| The Notifications Portlet | 32 |
| The Checked Out Content Portlet | 32 |
| Search for Content | 33 |
| Chapter 4: Managing Business Rules | 35 |
| Rule Manager Organization | 36 |
| Projects | 36 |
| Business Terms | 36 |
| Rule Sets and Rules | 37 |
| Log In to the Rule Manager | 43 |
| User Names/Passwords | 44 |
| Browse Projects and Rule Sets | 44 |
| Icons list | 45 |
| Create a Project | 46 |

| | |
|--|-----|
| Upload a Compressed File | 47 |
| Business Vocabulary Manager | 47 |
| Create a Vocabulary | 48 |
| Create Business Terms | 48 |
| Edit a Term | 50 |
| Delete a Term | 50 |
| Create User Types | 50 |
| Define Values For Option Is Composed Of | 51 |
| Define Values for Option Is an enumeration with the following values | 52 |
| Edit a Type | 52 |
| Delete a Type | 53 |
| Create a Rule Set | 53 |
| How to Create a Rule | 54 |
| How to Create a Rule Manually | 55 |
| How to Create a Rule Automatically | 82 |
| Create Test Cases | 87 |
| How to Modify a Test Case | 92 |
| Validate a Project | 94 |
| Work with Rule Sets | 94 |
| Convert Rule Manager Project to BRE | 94 |
| Duplicate a Rule Set | 95 |
| Move a Rule Set's Location | 96 |
| Edit a Rule Set's Properties | 96 |
| Work with Rules | 97 |
| Check In and Check Out a Rule | 97 |
| Duplicate a Rule | 98 |
| Delete a Rule | 99 |
| Move a Rule's Location | 100 |
| Edit a Rule's Properties | 102 |
| Version History | 102 |
| Specify Access and Permission | 105 |
| Available Users | 105 |
| Selected for Sharing | 106 |
| Permissions | 107 |
| Apply Settings To Rule Set | 107 |

Chapter 5: Uninstalling CA Aion Rule Manager 109

| | |
|---|-----|
| Uninstallation Considerations | 109 |
| How to Uninstall CA Aion Rule Manager | 110 |
| Unregister CA Aion Rule Manager Instance of EEM | 110 |
| Uninstall CA Aion Rule Manager Database | 111 |

Appendix A: Configuring CA Aion Rule Manager for the TLP Insurance Tour 113

| | |
|--|-----|
| Introduction to the TLP Insurance Tour | 113 |
| Roles | 113 |
| TLP Insurance-Summary of Activities | 114 |
| Log In to CA EEM | 115 |
| Create Pricing Tier Rules | 116 |
| Edit Rule for TPL Insurance | 117 |
| Test Rules for TPL Insurance | 121 |
| Deploy Rules for TPL Insurance | 124 |
| User and Group Accounts | 128 |
| Groups | 129 |
| Create Groups | 129 |
| Associate Groups with Portal Policies | 130 |
| Add Rule Manager Users to CA Aion Rule Manager | 130 |

Appendix B: Using Converted BRE Library 133

| | |
|--|-----|
| Library Structure | 133 |
| Entry Class | 134 |
| User Defined Classes | 135 |
| Rule Domain Classes | 135 |
| Data Type Classes | 137 |
| Mapping of Data Types | 138 |
| Data Type Differences | 138 |
| Number Type | 139 |
| Date and Time | 140 |
| Duration | 140 |
| String Operations | 141 |
| Engine Differences | 141 |
| When Unpending a Rule Premise | 141 |
| When Unpending a Rule Action | 142 |
| Null Instance Handling | 143 |
| Time Base in Client/Server Scenarios | 143 |
| Name Clashes for Iteration Variables | 144 |

Appendix C: Using and Creating Iterator 145

| | |
|---|-----|
| Create and Use Iterator in a Rule | 145 |
| Create Iterator Variable on Expression Builder Page | 146 |

Index 147

Chapter 1: Introduction

This section contains the following topics:

[About CA Aion Rule Manager](#) (see page 9)

[Audience](#) (see page 10)

[Components](#) (see page 10)

About CA Aion Rule Manager

CA Aion Rule Manager enables process users and IT professionals to design, execute, and administer core processes within your organization. Through the use of rules and automation technologies, processes become more consistent and more thorough. Also, it is a portal based solution which provides an interactive point of entry for web-based content and applications.

Users and groups have defined access permissions to all CA Aion Rule Manager services. Users manage rules. Content management functions of CA Aion Rule Manager provide access control and versioning for the relevant content and documents defining the business processes that are built using CA Aion Rule Manager.

CA Aion Rule Manager enables you to, without programming, define, edit, and test business rules and terms using intuitive web-based editors.

CA Aion Rule Manager provides the following:

- A web-based environment to step you through the phases of creating and testing a rule, as well as:
 - Access control to specified users and user groups
 - Rule check in and check out
 - Rule versioning
 - Rule search
- A rule repository to store the rules you create
- XML Based storage for rules created by Rule Manager
- An application to host the deployed rules created in Rule Manager as web services

Audience

Users of CA Aion Rule Manager are defined by the business role they perform, as explained below:

Rule Writer

Defines business terms, and writes and tests rules.

Business Analyst

Defines business terms and rules, and tests rules.

System Tester

Creates test plans, tests data, and runs tests.

Auditor

Verifies the rules that are in the system and keeps track of changes made to them.

Business User

Authors and reviews business documents.

IT Operations

Performs day-to-day operations of rules.

Each CA Aion Rule Manager feature benefits the following roles in your business process:

- Rule Writer
- Business Analyst
- System Tester

Components

CA Aion Rule Manager consists of multiple components that perform specific services and functions. The architecture of CA Aion Rule Manager is defined by its components and their interrelationships. To optimize performance you can choose to install these components across multiple computers.

CA Aion Rule Manager consists of the following main components (features):

Rule Manager

Enables you to build and maintain rules using its web-based interface. You can define business terms, organize rules into logical groups, edit rules, and test rules using the intuitive editors. The Rule Manager generates web services that execute the rules, which can be hosted within the same application server as the Rule Manager, or deployed to other web application servers. The Rule Manager is hosted on application server (JBoss).

CA Aion Rule Manager database

Defines tables, roles, privileges, procedures, and other database objects.

EEM

Manages defined users, groups, and access control policies. By default, eTrust Directory is selected as the directory and is installed along with EEM. However, you can configure EEM to work with an external directory such as Microsoft Active Directory after installing EEM. CA Aion Rule Manager supports EEM 8.4.

None of these components are optional. All the components are required for proper functioning of CA Aion Rule Manager.

Chapter 2: Installing CA Aion Rule Manager

This section contains the following topics:

[Installation Prerequisites](#) (see page 13)
[Installation Considerations](#) (see page 14)
[Installation Log Files](#) (see page 20)
[Install CA Aion Rule Manager](#) (see page 21)
[Silent Installation](#) (see page 21)

Installation Prerequisites

To ensure a successful installation

1. Read both the Release Notes and Readme File before you install CA Aion Rule Manager. These documents contain latest product news and information about installation considerations, operating system support, and CA Technical Support contact details. Do not install CA Aion Rule Manager until you read and understand these information.

Note: You can find the most current version of the Release Notes and Readme at <http://ca.com/support>.

2. Install JDK 1.5 and JBoss 4.0.3 SP1 on system where Rule manager installer will be run, so that you can install any web application component.

Note: Currently, JBoss is the only supported application server.

- To install the JDK, browse to `jdk-1_5_0_01-windows-i586-p.exe` file, located in the Prerequisites directory of the installation media, or to `<CD_root>\Prerequisites\jdk-1_5_0_01-windows-i586-p.exe` on the installation image.
- To install JBoss, copy the `jboss-4.0.3SP1` directory from the Prerequisites directory of the installation media, or from the `<CD_root>\Prerequisites` directory on the installation image, to a local directory that you choose on the computer running the installation.

3. Install Microsoft SQL Server 2005 or Microsoft SQL Server 2008 on your network.

Note: Microsoft SQL Server 2005 and Microsoft SQL Server 2008 database types are supported.

4. If you have installed Microsoft SQL Server 2005 on your network, install Microsoft SQL Server 2005 Client on system where Rule manager installer will be run. If you have installed Microsoft SQL Server 2008 on your network, install Microsoft SQL Server 2008 Client on system where Rule manager installer will be run.

Note: Rule manager installer uses SQL Server command line utility sqlcmd.exe which is provided with the SQL Server 2005 client and SQL Server 2008 client. Make sure environment variable PATH contains path of sqlcmd.exe. SQL client version should be compatible with the SQL Server version.

Installation Considerations

For a successful installation, read the following considerations before you install CA Aion Rule Manager.

Install Over the Network

You *cannot* install CA Aion Rule Manager components to a mapped network drive. The installation destination folder you specify must be on the computer that is running the installation.

Authenticate Shared EEM Server

You can install CA Aion Rule Manager to authenticate against an existing EEM server. To plan for this type of installation, you must have the server host name and EiamAdmin password. You must also decide whether to install CA Aion Rule Manager using existing application instance (in which case you need the application instance name, certificate file for the application instance, and the password for the certificate file) or into a new one. You can also have CA Aion Rule Manager install a new instance of EEM.

Access CA Aion Rule Manager Database

CA Aion Rule Manager accesses the CA Aion Rule Manager database through a specific database account, which is specified during the installation. CA Aion Rule Manager can connect to an existing SQL Server database or install a new database to an existing SQL Server.

If you plan to use an existing CA Aion Rule Manager database, you must have the database server's hostname, connection port number for the SQL Server, and credentials for accessing the database. During the installation, you will be asked for the account name and password.

If you plan to create a new CA Aion Rule Manager database, you must have server's hostname, connection port number for the SQL Server and SQL Server user with administrative credentials. In addition to this, you will be asked for the database account name (SQL Server login) and password, during the installation. The installation can create the SQL Server login (if requested) or use an existing one.

Gather Information on CA Aion Rule Manager

To help you organize and plan for the installation of CA Aion Rule Manager, gather the following information.

Note: If you are installing CA Aion Rule Manager components across multiple computers and are connecting to existing shared installations of certain components (such as EEM or CA Aion Rule Manager database), the product installation uses this information to connect to the distributed components. Otherwise, the installation uses this information to configure the components being installed.

You must select all three features for new installation of CA Aion Rule Manager. For subsequent installations (to configure CA Aion Rule Manager database and/or EEM) unless you uninstall CA Aion Rule manager, you can select CA Aion Rule Manager database and/or EEM features for installation.

If you are performing new installation, you should specify Installation directory where Rule manager files will be deployed otherwise this value is not required.

You cannot install CA Aion Rule Manager on a computer where CA Aion Rule Manager feature is already installed. However, you can configure CA Aion Rule Manager database and EEM with existing CA Aion Rule Manager installation, that is, you can install CA Aion Rule Manager database and EEM features. Remove the existing CA Aion Rule Manager installation using Add or Remove Programs to reinstall the CA Aion Rule Manager feature.

The following registry key determines whether Rule manager is installed or not:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AionRM

If this key is present, Rule manager is installed otherwise Rule manager is not installed.

Information on Installation

Consider the following information for CA Aion Rule Manager:

- Will this installation involve multiple computers?
 - You may want to consider using multiple computers for performance, maintenance, security, and resiliency reasons. CA Aion Rule Manager may span across multiple computers with one computer for the EEM server and other for the CA Aion Rule Manager database, and another for Rule Manager application. By separating components, you can minimize the effect of a single point of failure and maximize performance.
- Multiple CA Aion Rule Manager instances sharing one database.
 - You may want to install separate CA Aion Rule Manager instances that share the same database. Each CA Aion Rule Manager instance will operate independent of all others.
 - If you have multiple CA Aion Rule Manager instances that will share the same database, be sure to designate different instance IDs when you install each CA Aion Rule Manager instance.
 - CA Aion Rule Manager instances can use the same EEM server, but must use different EEM application instances.
- Do you have an existing directory with existing users and groups that you want to use, such as eTrust Directory or Microsoft Active Directory?
 - The installation of EEM will automatically install eTrust Directory along with the EEM component. To use a different external directory for your users and groups, you must configure EEM after the installation of EEM is complete.
- Do you want to use an existing EEM application instance to store all CA Aion Rule Manager related management information?
 - During the installation of the CA Aion Rule Manager, you can use an existing EEM application instance. If you are using an existing application instance, you must have the EEM server host name, EiamAdmin user password, the name of the application instance, the location and name of the certificate file on the file system, and the password for the certificate file.

Information on CA Aion Rule Manager Database

Check that you have the following information:

Database name

Specifies the name of the existing CA Aion Rule Manager database, or a name you select if the CA Aion Rule Manager database needs to be created.

Database username

Specifies the database username. If you are connecting to existing CA Aion Rule Manager database, this user must exist with access to this database. If you plan to create new CA Aion Rule Manager database, this can be an existing user or a new user that will be created during the installation.

Database password

Specifies the password for the database user name.

Hostname

Specifies the name of the SQL Server host. It can be local or remote SQL Server host.

Port

Specifies the port number of the SQL Server installation which contains, or will contain, the CA Aion Rule Manager database. The default port number for SQL Server is 1433.

Note: To enable CA Aion Rule Manager to connect to a SQL Server named instance, you must specify the port number of the named instance in this field. The named instance must have previously been set up, and the port number assigned, during the SQL Server installation.

Admin username

Specifies the SQL Server user with administrative privileges, if you plan to create new CA Aion Rule Manager database. If you are connecting to existing CA Aion Rule Manager database, this field is not required.

Admin password

Specifies the password for the Admin username, if you plan to create new CA Aion Rule Manager database. If you are connecting to existing CA Aion Rule Manager database, this field is not required.

Information on EEM

Check that you have the following information:

Hostname

Specifies the host name. If you are connecting to existing EEM, name of EEM server host. If you plan to install new EEM server, this field is not required if installer is run in GUI mode.

Note: New EEM server can be installed only on the computer that is running the installation. New EEM server cannot be installed on Windows Vista and Windows 2008. If installer is run on Windows Vista or Windows 2008, existing remote EEM host must be used.

Admin Password

Specifies the password for EEM user EiamAdmin (existing if using existing EEM, or one you choose, if EEM should be installed).

Application Instance

Specifies the application instance. We recommend that a separate application instance be created for each product using the EEM installation. Therefore, the default value of RuleManager can be used when installing the CA Aion Rule Manager. This application instance will be created in EEM if it does not already exist. This is the application instance name that should be selected when logging in to EEM to manage user accounts for CA Aion Rule Manager.

Certificate File

Specifies the fully-qualified path to the encrypted certificate file for the application instance in EEM. When reusing an application instance that already exists, you must have a copy of the certificate file which was created for this application instance during a previous installation.

Search for a file with the extension .p12. The certificate file typically has the following naming convention:

EEM Server Name-Application Instance Name.p12

The password for this certificate file must be available. When creating a new application instance, a certificate file will be created for this application instance. Note the location, name, and corresponding password for future use of the new application instance.

Certificate Password

Specifies the password for the certificate file of the application instance in EEM.

Information on Application Server

Check that you have the following information:

Configuration

Specifies the configuration of the application server.

Note: An application server (JBoss) is a prerequisite for CA Aion Rule Manager. For information about how to install JBoss, see [Installation Prerequisites](#) (see page 13). For JBoss, Configurations are directly under JBOSS_HOME\server.

HTTP Port

Specifies the HTTP port of the application server.

HTTPS Port

Specifies the HTTPS port of the application server.

RMI Port

Specifies the RMI port of the application server.

JDK

Specifies the Directory location of the JDK installation.

Note: A JDK is a prerequisite for CA Aion Rule Manager. For information about how to install the JDK, see [Installation Prerequisites](#) (see page 13).

This field is populated with JAVA_HOME environment variable, if it is present. If JAVA_HOME is not present, an error message is displayed when user starts installation and installation is terminated. JAVA_HOME can be User variable or System variable.

Configuration, Ports information and JDK path are required only if CA Aion Rule Manager feature is selected. Check that the ports specified are not already used by other services, such as, JBoss.

Information on Content Management

Check that you have the following information:

Instance ID

Specifies the instance ID of a CA Aion Rule Manager installation connecting to the CA Aion Rule Manager database. The Instance ID must be numeric and unique for each separate CA Aion Rule Manager installation sharing the same CA Aion Rule Manager database.

You can reuse existing Instance ID if that Instance ID was created on this system (system where you are running Rule manager installation) by the previous Rule manager installation. In this case, make sure that you use same CA Aion Rule Manager database and Library Location which were used earlier with this Instance ID.

The following query can be run in the CA Aion Rule Manager database to check which Instance IDs are already in use:

```
select ueporinstanceid from cmscpmconfig;
```

Library Location

Specifies the location where CA Aion Rule Manager should store its Library content. This can be any directory location.

Username

Specifies the EEM user name to log in to CA Aion Rule Manager with all the administrative rights (if the specified user does not exist, installation can create it in EEM).

Password

Specifies the password for the administrative CA Aion Rule Manager user.

Note: Instance Id and Library Location are required only if CA Aion Rule Manager database feature is selected.

Username and Password are required only if EEM feature is selected.

Installation Log Files

During CA Aion Rule Manager installation, the following log files are created depending on features selected for installation:

- AionRM_InstallLog_Date-Time.log: <install_dir>\AionRM\logs

This log file contains operation-wise log of entire installation process.

This file name is appended with installation date and time.

For example, If the installation is started at 3:25 PM on 19-April-2010, the log file name will be

```
AionRM_InstallLog_19_4_2010-15_25.log
```

- AIONDB_sqlserver.log: <install_dir>\AionRM\logs

This log file contains log of database creation and updating process.

- eem_install.log: <install_dir>\AionRM\logs

This log file contains log of EEM server installation and Application instance registration processes.

- eem.log: <install_dir>\AionRM\logs

This log file contains log of Rule Manager user creation process.

- AionRM_InstallMessages.log: <temp_dir> (%temp%)

This log file contains log of pre-install operations and validations. This file is useful in case of silent installation to check pre-install errors.

Install CA Aion Rule Manager

The installation wizard installs each feature that you have selected. From a single installation source, you can install multiple features of CA Aion Rule Manager.

Note: You cannot install CA Aion Rule Manager through Unicenter Software Delivery.

To install CA Aion Rule Manager

1. Log in to the target installation computer using an account with administrator privileges.
2. If JBoss is running, stop it.
3. Insert the installation media into the appropriate drive or access the product installation image through a mapped drive.

The CA Aion Rule Manager installation starts automatically when you insert the installation media.

Note: To start the installation manually, double-click <Drive:\>Windows\VM\setup.exe on the installation media.

4. Follow the on-screen instructions to complete the installation.

Note: For more information about a field or option during the installation, click the Help button.

Silent Installation

In silent mode, installer has no end-user interaction, and runs by providing a response file from which the installer retrieves the values for various installer variables used to control the install.

Enter following command at command prompt to run installer in silent mode:

Setup -i silent -f <<fully qualified path of response file>>

To generate response file

1. Enter the following command at the command prompt:

`Setup -r <<fully qualified path of response file>>`

This command starts installer in GUI mode.

2. Provide inputs and proceed with installer pages.
3. Click Cancel to exit installer.

Response file will be generated at the specified location. This file utilizes a simple key=value format.

Note: To generate response file, it is not necessary that you complete the entire installation process. The variables defined till you exit the installer are saved in the response file.

This command records all of the variables defined in your installation except CHOSEN_INSTALL_FEATURE_LIST variable in response file. The CHOSEN_INSTALL_FEATURE_LIST variable should be added manually. If this variable is not present in response file at the time of installation, all features will be installed.

Also, this command does not record USER_INSTALL_DIR and APPSERVER_CONFIGURATION variables in response file if all the features are not selected for installation. In this case, you should add these variables manually.

On Windows Vista and Windows 2008, Command prompt should be started with Run as administrator option. If the above- mentioned commands are run from within batch (.bat) file on Windows Vista and Windows 2008, batch file should be run with Run as administrator option.

Variables Used in Installer Response File

The following variables are used in Installer Response File:

USER_INSTALL_DIR

Specifies the Installation folder.

Note: USER_INSTALL_DIR is required if CA Aion Rule Manager feature is selected for installation.

CHOSEN_INSTALL_FEATURE_LIST

Defines the list of short names of features to be installed. The names do not contain any spaces and are separated by commas.

Valid Values: The following table consists of valid values:

| Feature | Short name |
|-------------------------------|------------|
| CA Aion Rule Manager database | RMDB |
| EEM | EEM |
| CA Aion Rule Manager | CARM |

CREATE_RMDB

Creates a database. Type 1 to create a database, otherwise type 0.

CONFIGURE_RMDB

Configures the existing database. Type 1 to connect to the existing database, otherwise type 0.

DATABASE_NAME

Specifies Database name.

RMDB_USER

Specifies Database user.

RMDB_USER_PASSWORD

Specifies Database user password.

CREATE_RMDB_USER

Creates a database user. Type 1 to create a database user, otherwise type 0 to use the existing database user.

RMDB_USER_PASSWORD_CONFIRM

Confirms database user password if CREATE_RMDB_USER is 1. If CREATE_RMDB_USER is 0, this field is not required.

DB_HOSTNAME

Specifies SQL Server host name.

DB_PORT

Specifies port number of the SQL Server.

DB_ADMIN

Specifies SQL Server user with administrative privileges.

DB_ADMIN_PASSWORD

Specifies password for the admin user name.

Note: DB_ADMIN and DB_ADMIN_PASSWORD are required only if CREATE_RMDB is 1. CREATE_RMDB, CONFIGURE_RMDB, DATABASE_NAME, RMDB_USER, CREATE_RMDB_USER, RMDB_USER_PASSWORD, RMDB_USER_PASSWORD_CONFIRM, DB_HOSTNAME, DB_PORT, DB_ADMIN, DB_ADMIN_PASSWORD fields are required if CA Aion Rule Manager database feature is selected for installation.

INSTALL_EEM

Installs a EEM server. Type 1 to install a EEM server, otherwise type 0.

CONFIGURE_EEM

Configures the existing EEM server. Type 1 to connect to the existing EEM server, otherwise type 0.

EEM_HOSTNAME

Specifies EEM server host name.

EEM_ADMIN_PASSWORD

Specifies password for EEM user EiamAdmin.

EEM_ADMIN_PASSWORD_CONFIRM

Confirms password for EEM user EiamAdmin if INSTALL_EEM is 1. If INSTALL_EEM is 0, this field is not required.

APPLICATION_INSTANCE

Specifies application instance in EEM.

USER_CERT_PATH

Specifies fully-qualified path of certificate.

CERT_PASSWORD

Specifies certificate password.

CREATE_CERT

Creates a certificate. Type 1 to create a certificate, otherwise type 0.

CERT_PASSWORD_CONFIRM

Confirms certificate password if CREATE_CERT is 1. If CREATE_CERT is 0, this field is not required.

Note: INSTALL_EEM, CONFIGURE_EEM, EEM_HOSTNAME, EEM_ADMIN_PASSWORD, EEM_ADMIN_PASSWORD_CONFIRM, APPLICATION_INSTANCE, USER_CERT_PATH, CERT_PASSWORD, CREATE_CERT, CERT_PASSWORD_CONFIRM fields are required if EEM feature is selected for installation.

APPSERVER_CONFIGURATION

Specifies configuration of the application server.

HTTP_PORT

Specifies HTTP port of the application server.

HTTPS_PORT

Specifies HTTPS port of the application server.

RMI_PORT

Specifies RMI port of the application server.

APPSERVER_JDKPATH

Specifies directory location of the JDK installation.

Note: APPSERVER_CONFIGURATION, HTTP_PORT, HTTPS_PORT, RMI_PORT and APPSERVER_JDKPATH are required if CA Aion Rule Manager feature is selected for installation.

INSTANCE_ID

Specifies instance ID of a CA Aion Rule Manager installation.

LIBRARY_LOCATION

Specifies location where CA Aion Rule Manager should store its Library content.

Note: INSTANCE_ID and LIBRARY_LOCATION are required if CA Aion Rule Manager database feature is selected for installation.

The following variables are required if EEM feature is selected for installation.

RM_EEM_USER

Specifies EEM user name to log in to CA Aion Rule Manager with all the administrative rights.

RM_EEM_USER_PASSWORD

Specifies password for the administrative CA Aion Rule Manager user.

CREATE_EEM_RM_USER

Creates a user. Type 1 to create a user, otherwise type 0.

RM_EEM_USER_PASSWORD_CONFIRM

Confirms password for CA Aion Rule Manager user if

CREATE_EEM_RM_USER is 1. If CREATE_EEM_RM_USER is 0, this field is not required.

If response file is manually created, path values should contain double slashes.
For example,

USER_INSTALL_DIR=C:\\Program Files\\CA\\AionRM

Sample Response File

```
CHOSEN_INSTALL_FEATURE_LIST=RMDB,EEM,CARM
#Choose Install Folder
#-----
USER_INSTALL_DIR=C:\\Program Files\\CA\\AionRM
#Get User Input
#-----
CREATE_RMDB=0
CONFIGURE_RMDB=1
#CA Aion Rule Manager - Aion Rule Manager database Information (1 of 2)
#-----
USER_INPUT_RESULT_1=1
DATABASE_NAME=RMDB_14909_4
RMDB_USER=rmdba
RMDB_USER_PASSWORD=0B0B5154405A616F
CREATE_RMDB_USER=0
RMDB_USER_PASSWORD_CONFIRM=
#CA Aion Rule Manager - Aion Rule Manager database Information (2 of 2)
#-----
DB_HOSTNAME=localhost
DB_PORT=1433
#EEM information
#-----
EEM_HOSTNAME=localhost
EEM_ADMIN_PASSWORD=0B0B5154405A616F
APPLICATION_INSTANCE=RuleManager_16909_2
USER_CERT_PATH=C:\\Program Files\\CA\\aphaw01-xp2-16909-2-cert.p12
CERT_PASSWORD=0B0B5154405A616F
CREATE_CERT=1
CERT_PASSWORD_CONFIRM=0B0B5154405A616F
```

```
#Application Server information
#-----
APPSERVER_CONFIGURATION=C:\\jboss-4.0.3SP1\\server\\modified
HTTP_PORT=8080
HTTPS_PORT=8443
RMI_PORT=1098
APPSERVER_JDKPATH=C:\\Program Files\\Java\\jdk1.6.0_04
#Content management information
#-----
INSTANCE_ID=10
LIBRARY_LOCATION=C:\\Program Files\\CA\\AionRM\\Library_17909
RM_EEM_USER=rm admin_14909_4
RM_EEM_USER_PASSWORD=0B0B5154405A616F
CREATE_EEM_RM_USER=0
RM_EEM_USER_PASSWORD_CONFIRM=
```


Chapter 3: Using CA Aion Rule Manager

CA Aion Rule Manager provides a common, personalized workplace where users can retrieve rules and content.

This section contains the following topics:

[Access CA Aion Rule Manager](#) (see page 29)

[CA Aion Rule Manager Portlet](#) (see page 30)

Access CA Aion Rule Manager

The following procedure allows you to access CA Aion Rule Manager.

To access CA Aion Rule Manager

1. Click Start, Run, and enter the following:

`http://hostname:8080/rmportal`

Note: You must enter your host name rather than “hostname”. The port may differ, depending on the HTTP port on which you installed CA Aion Rule Manager.

The CA Aion Rule Manager Log In page displays.

2. Enter the User Name and Password that you specified for the rm admin user during the install, or a user you have added since, and click Log In.

More information:

[User Names/Passwords](#) (see page 44)

CA Aion Rule Manager Portlet

CA Aion Rule Manager installs specific content in the portlet to enable product functionality.

The CA Aion Rule Manager contains portlets that let you browse content, edit business terms and rules, configure access control, manage content change notifications, and other Rule Manager tasks.

CA Aion Rule Manager | [Home](#) | [Help](#)

Logged in as: **pm admin** ([Log Out](#))

Rule Manager

Folders

All Folders

31 - 32 of 32

| Title | Details |
|----------------------|---------|
| All Folders | |
| ZMX Bank Rules | |
| TLPInsurance.zip.zip | |

Search

Recent Searches:

Keywords:

Go [Advanced](#)

Tip: Separate keywords with spaces.

Saved Searches:

susansearch Go

Notification

Show More

| Last Updated | Title | Details |
|--------------|------------------------|---------|
| May 12, 2009 | SampleProj12 | |
| May 5, 2009 | CA Business Vocabulary | |
| Apr 24, 2009 | DemoTestCase | |
| Apr 24, 2009 | demorule | |
| Apr 17, 2009 | AJA Test Case | |

Checked-Out Content

Show More

| Title | Details | Check In Content |
|---------------------------|---------|------------------|
| AJA Rule | | |
| AJA Test Case | | |
| AXA Vocabulary | | |
| CA Business Vocabulary | | |
| CartDefaultRule | | |
| CashAwardTestCase | | |
| CashRewardLookUprule | | |
| CXC Rule | | |
| DefaultSalary | | |
| demorule | | |
| DemoTestCase | | |
| discountLookupRuleUpdated | | |
| EmployeePayableVocab | | |
| FinalPayableRule | | |
| IfThenWithIterator | | |
| inpTerm | | |

Note: The CA Aion Rule Manager installation installs the portlets for the rm admin user.

Portlets are windows in a Portal page that manage content directly in the CA Aion Rule Manager infrastructure. They provide the ability to store, manipulate, publish, and distribute various types of digital assets such as documents, graphics, URLs, and so on. They can also function as containers; they may contain many different items, ranging from one or more files to small applications.

Applications content consists of portlet definitions. Some standard CA Aion Rule Manager portlet definitions such as the Search portlet, Folders portlet, Notifications portlet, and Checked-Out Content portlet are installed with CA Aion Rule Manager. Software developers can create new JSR 168 portlets and deploy them to the CA Aion Rule Manager using Apache Jetspeed 2 environment.

Folder Portlet

Folders portlet is used to manage content. CA Aion Rule Manager content is largely comprised of files and applications. Files exist in folders and can be documents, letters, spreadsheets, graphics, and URLs. This content can be checked out, edited, and use.

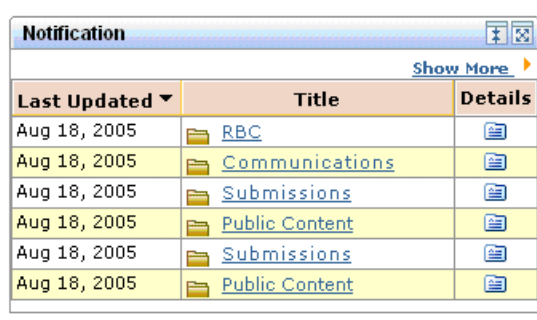
Note: Rules, vocabularies, and tests are implemented as content in CA Aion Rule Manager, and the library services provide the management functions of access control, versioning, and metadata processes. The portlets and functions described in this section apply to all CA Rule Manager content.

Folder portlet allow users to perform a variety of content related activities, including the following:

- Add, delete, and update content using the CA Aion Rule Manager Library portlets or a WebDAV client (such as Microsoft Windows Web Folders).
- Assign permissions to content.
- Create versions of content.
- Check-out content for modification.
- Download content.
- Subscribe to content and receive notifications when content or its properties are changed.
- View and update properties (metadata) for content.

The Notifications Portlet

Often, many people in the organization want to know when changes are made. These users can subscribe to individual content items as well as to entire folders. When a change is made to an item or folder, they receive a notification in the Notification portlet on their Library page. The Notification portlet on the Library page shows the last updated date of the subscribed content.



The screenshot shows a portlet titled "Notification" with a "Show More" link. It contains a table with three columns: "Last Updated", "Title", and "Details". The table lists six items, all updated on "Aug 18, 2005". The titles are "RBC", "Communications", "Submissions", "Public Content", "Submissions", and "Public Content". Each item has a folder icon and a document icon in the "Details" column.

| Last Updated ▼ | Title | Details |
|----------------|--------------------------------|---------|
| Aug 18, 2005 | RBC | |
| Aug 18, 2005 | Communications | |
| Aug 18, 2005 | Submissions | |
| Aug 18, 2005 | Public Content | |
| Aug 18, 2005 | Submissions | |
| Aug 18, 2005 | Public Content | |

The Checked Out Content Portlet

Use the Checked-out Content portlet to see the files you have checked-out. When you are working with a file, you should check it out from the Library so others cannot modify it.

By checking-out a file, you ensure that only one user at a time can update the file or its properties. When you check-in a file, a new version is created. Versioning allows you to record the progress of work on the document and view it as it appeared at different stages throughout the work cycle.

When a file is checked-out, all users who have the appropriate permissions, have access to the following:

- See the file in the Folders portlet
- Open the file
- See the file properties
- Download the file

In addition, the user who checked-out the file can perform the following actions:

- Change the file properties and add contents
- Replace the file

Note: These changes cannot be viewed by other users until the file is checked-in.

Search for Content

The Search engine enables rapid and efficient location of content in CA Aion Rule Manager Library.

CA Aion Rule Manager includes Search and Advanced Search links on every page so you can search for content at any time using various criteria. These links provide quick access to the Search and Advanced Search portlets. The Advanced Search feature lets you specify advanced filters to refine search criteria. The filters offer additional granularity including title, description, creation and modification dates, content location, and file type.

Search Results

Save This Search Done

Search Results

Search Again

Query: dashboard

Go Advanced

Saved Searches:







Go

Recent Searches:

Search Results

[Hide Summaries](#)
[Show Summaries](#)

1 - 10 of 12 > >>

| Score | Title | Add | Details | Description | Collection | Created |
|----------|---|-----|---|-------------|----------------|------------------------|
| 30% |  Dashboard VII.xppg | |  | | Portal Library | Dec 9, 2005 5:22:02 PM |
| Summary: | | | | | | |
| 30% |  Dashboard X.xppg | |  | | Portal Library | Dec 9, 2005 5:22:02 PM |
| Summary: | | | | | | |
| 30% |  Dashboard XI.xppg | |  | | Portal Library | Dec 9, 2005 5:22:02 PM |
| Summary: | | | | | | |

Chapter 4: Managing Business Rules

Business rules are the practical knowledge of running your business, based on your business expertise, concepts, policies, and objectives. Using CA Aion Rule Manager your business rules are exposed as a web service that an application can use. With this practical knowledge implemented in a rules-based application, you can manage business processes and procedures to achieve your business objectives automatically.

Business rules combine complex business logic, knowledge, and policies in a straightforward manner. CA Rule Manager provides the ability to create, change, and manage the business rules that are under your domain to achieve the business objectives.

This section introduces you to CA Rule Manager, a web-based rules management facility that enables business and IT alike to create, manage, and maintain rules without programming skills.

Note: An example for various options in CA Aion Rule Manager is available in the appendix [Configuring CA Aion Rule Manager for the TLP Insurance Tour](#) (see page 113).

This section contains the following topics:

[Rule Manager Organization](#) (see page 36)

[Log In to the Rule Manager](#) (see page 43)

[Browse Projects and Rule Sets](#) (see page 44)

[Icons list](#) (see page 45)

[Create a Project](#) (see page 46)

[Business Vocabulary Manager](#) (see page 47)

[Create a Rule Set](#) (see page 53)

[How to Create a Rule](#) (see page 54)

[Work with Rule Sets](#) (see page 94)

[Work with Rules](#) (see page 97)

[Specify Access and Permission](#) (see page 105)

Rule Manager Organization

Before using CA Rule Manager, it is important to understand its hierarchical organization.

CA Rule Manager lets you define business terms, organize rules into logical groups, and edit and test rules using the rule editor, all without programming. CA Rule Manager offers rule editing templates for creating, renaming, editing, and deleting rules.

Each rule is defined in a *rule set*, a collection of rules, stored in a *project* in CA Rule Manager. A project is a collection of rules in rule sets. Projects also contain the *vocabulary* used in the rules. *Test cases* are used to verify the rules.

Projects

CA Rule Manager organizes rules into a hierarchy of folders. This allows users to organize and manage access to rules efficiently. The project folder, at the top of the hierarchy, groups all the rules based on the purpose of the rules.

Projects are deployable units of business logic. A project is a folder which contains all the entities related to one business rule service. They contain:

- Business vocabulary (or business terms)
- Rule Sets
- Test cases (for testing rules)

Note: The portal administrator or any user with rights to create content in All Folders must initially create the project in Rule Manager.

The rule project also corresponds to a deployed rules web service.

The All Folders folder lists the Rule Manager projects as you develop them. You can also create a Rule Manager project in any folder that is not contained in a Rule Manager project hierarchy.

Business Terms

A Rule Manager Project requires a vocabulary of business terms, which are created using the Business Vocabulary Manager. Rules in CA Aion Rule Manager are defined using business terms. A business term is a word or phrase that has some meaning to the business (like cost of goods sold or application date) rather than a programming term like a variable name or a database field. Rules test and set the values of business terms. To a programmer, business terms are the variables of rules.

A business term is defined by its name (includes spaces, capitalization, and so on), data type (text, numeric, Boolean or user type), usage (input, output or internal), and description. You can identify how the business terms are used after the project is deployed (that is, accessed by client applications). You can mark business terms as an input to the project, output, or internal. An input term represents a value that the rules project needs before execution. An output term represents a value that is returned by the rules project after execution. An internal term is one whose value can be set during execution, but whose value is not returned after execution. Designating input or output terms defines the interface of the project with the rest of the business environment.

Business Vocabulary Manager

The Business Vocabulary Manager helps you create, edit and delete vocabularies, business terms and types. A Business Vocabulary is a collection of terms and types. You can use the default types or create types based on the default types. The vocabulary is provided with check-out and check-in features. The vocabulary must be checked out to add or edit terms and types. The Vocabulary is stored in XSD (XML Schema Definition) format and integrates seamlessly with the Rule Manager.

Rule Sets and Rules

Rule set, a collection of rules, is defined in a Project.

A *rule* is a single piece of knowledge, defining the value of a business term, perhaps by referring to other business terms. Rules can be simple expressions ("monthly salary = annual salary / 12") or conditional ("if amount > 1000 then refer for approval = true"). Rules can constrain some aspect of the business ("interest rate > 0"), and some rules can define conditions that apply in the absence of other rules ("refer by default"). CA Aion Rule Manager provides the ability to create rules that are in the common business language - not code. Business analysts can specify and process rules, and apply them to a particular situation.

When a rule project is called by an application, the values of the input business terms are provided by the client application. The rules run, making as many decisions as possible from the given data. If a Business Term used as a condition in a rule does not yet have a value, CA Aion Rule Manager automatically tries other rules and then returns to this business term later. The values of the output business terms are then returned to the calling application.

Business Rule Types

There are five basic rule types that fall into two categories used in the Rule Manager.

- [IF/THEN Rule](#) (see page 38)
- [Lookup Table Rule](#) (see page 38)
- [Definition Rule](#) (see page 40)
- [Mandatory Constraint Rule](#) (see page 41)
- [Default Rule](#) (see page 42)
- [Lookup Table from Data](#) (see page 43)

IF/THEN Rule

The IF/THEN rule is a simple rule that has a single condition and one or more actions associated with the condition. The actions typically assign a value to a business term, they can also create an instance of a type and perform set remove, combine operations. When creating a new rule, this is the default selection on the Create Rule page.

This rule contains the classic If/then statement in the following format:

IF (condition) THEN (actions)

Condition

Defines a logical expression based on business terms, specific values, or more complex expression containing business terms and/or specific values.

Actions

Defines the actions to be performed when the condition is satisfied.

Example

IF (age > 18 and age < 35 and gender = "male") THEN
driver class = "young adult male"

Lookup Table Rule

A Lookup Table Rule is a collection of individual rules that use the same business terms.

Each row in the Rule Table comprises of one or more conditions, each of which tests the value of a different business term (these conditions form columns in the table) and one or more actions which set the value of a business term.

The rows in a Rule Table are evaluated from top to bottom. The first row whose conditions are true is *executed*; its actions are performed.

In a Lookup Table Rule you are free to use different test values for each business term in each row. You can even leave a test value blank or test for unknown, in which case that condition is always considered true. If the final row in a rule table has no test values, it behaves as a default and the actions are performed if no earlier row in the table was executed. This means you should put specific tests before *don't care* tests in a Table Rule.

Example

This example contains a different pricing policy. In this example, the sizes and weights are different for different colored boxes. Because of this variation it is necessary to use Lookup Table to represent the rules.

Notice that some of the possible combinations of conditions are not expressed in the table. This may be because boxes are not sold in that category, or someone could have forgotten to write the rule. Or, another Rule Table may deal with that combination of conditions. It may even be that we have a separate Default rule for price so that if no rule sets that value we use a default value. It is the task of the Rule Author to determine how best to organize the rules.

Notice the use of the blank test value for size for green boxes - this is interpreted to mean that size does not matter in determining the price of green boxes (only weight matters) - and there are two categories (over 300lbs or not).

The final row in the table shows how a (partial) default is specified. Since all the test values except weight are blank, this rule executes if none of the previous rules executed. This rule assigns a price of \$45 only if the weight exceeds 300lbs. For boxes whose weight is no more than 300lbs that are not covered by the earlier rules, the price is not defined here.

| Rule Table | | | |
|------------|---------|--------------|-------|
| color | IF | | THEN |
| | size | weight | price |
| red | = large | > 500 | 60 |
| | | 100 >..< 500 | 50 |
| | | <= 100 | 40 |
| | = small | > 150 | 30 |
| | | < 150 | 25 |
| green | = large | > 700 | 60 |
| | | <= 200 | 50 |
| blue | = small | < 200 | 30 |
| | = large | > 100 | 99 |
| | | <= 100 | 99 |
| green | | > 300 | 65 |
| | | <= 300 | 75 |
| | | > 300 | 45 |

Here is another example of how such a rule table might appear in the policy manuals of an Auto Insurance company that has different policies for its driver ages in different states across the US:

| State | Age | Driver Category |
|-------|---------|-----------------|
| AK | <14 | Not allowed |
| AK | 14-15 | Provisional |
| AK | 15 - 65 | Adult |
| AK | > 65 | Senior |
| AZ | <15 | Not allowed |
| AZ | 15 - 67 | Adult |
| AZ | > 67 | Senior |
| CA | <15 | Not allowed |
| CA | 15-16 | Provisional |
| CA | 17 - 66 | Adult |
| CA | > 67 | Senior |

Definition Rule

A Definition rule specifies the value for a business term unconditionally, that is, without testing a condition that must be true in order to execute. The value of the business term can be specified either using a specific value (like "100"), or that of another business term, or by using an expression that computes its value using specific values, functions and/or other business terms.

This type of rule specifies how to compute or derive a value, based on some mathematical and other functions that provide a precise definition for a derived term.

An expression can be of multiple types and varying complexity, for example:

- Mathematical expression where all the terms must be numeric.
- A Boolean expression where all the terms must be true or false.
- An expression involving strings where all the terms must be String.

Definition Using a Specific Value

A Definition rule can provide a specific value for a business term.

Definition Using Expressions

A Definition rule can compute the value of a business term based on a mathematical formula.

Example: Definition Using a Specific Value

Rule: Department 10 = "Commercial Loan Underwriting"

Rule: interest rate = 6.25

Rule: hot = true

Example: Definition Using Expressions

Rule: Coverage Premium = Auto Replacement Cost * Policy Risk Factor / 12

In an automobile insurance application, the Rule Author defines the premium in terms of other available values.

Definitions can also be written using terms that are of type String.

Rule: error message = "Warning: " + *customer name* + " is overdrawn"

Where the customer name is inserted into the text of a message.

You can also define a term as a Boolean expression such as:

Rule: unpleasant = (hot and wet) or (cold and windy)

Each of the terms hot, wet, cold and windy must be of type Boolean.

Mandatory Constraint Rule

A Mandatory Constraint rule is a statement that protects assumptions about the value of an input business term, or about a relationship among input business terms. The statement must be true before the rules can execute. When a mandatory constraint rule fails, rule processing stops and an error message can be issued to the calling application, which can take action to satisfy the rule.

Note: Mandatory constraint rules apply only to input business terms.

Example

A mandatory constraint rule expresses assumptions that are required before the rules can execute. It also checks that the terms used in the expression are not unknown and throws a constraint violation if they are unknown.

Rule: Driver Age > 15

In an application to determine the automobile insurance premium for a driver, the driver always must be of legal age to have a driver's license. If this assumption is not met, all subsequent rules to calculate the auto insurance premium are irrelevant. Therefore, if a driver's age is less than sixteen, the mandatory constraint rule causes rule processing to stop.

Rule: theCart <> null

This mandatory constraint rule checks two preconditions:

- theCart should not be null
- theCart should not be unknown

If either of these assumptions are not met, rule processing stops.

Default Rule

Note: This rule is applicable only to the output terms.

A default rule is applied when a business term is still without a value after rule processing. It is used to assign a default value to a business term. The value of the business term can be specified either using a specific value (like "100"), or that of another business term, or by using an expression that computes its value using specific values and/or other business terms.

Only output business terms may have defaults. Input terms or internal terms cannot have defaults.

Since defaults are applied after rule processing has finished, the value that is assigned as the result of a default cannot be used by other kinds of rules.

Example

A default rule, sets a value for the business term Risk if rule processing does not provide one.

Rule: Risk = "Refer to risk consultant for analysis."

In a risk evaluation application, there is no guarantee that a Risk value is always set. Even when rules are provided to set this value, their If conditions might not be satisfied. If it is critical that Risk always be assigned a value, the default above sets Risk to "Refer to risk consultant for analysis" as a last resort.

Lookup Table from Data Rule

A Lookup Table from Data rule is a form of [Lookup Table rule](#) (see page 38) created automatically by mining the user data from the database tables using the data-mining engine. Based on the data in the data columns selected by the user, one or more rules will be created automatically. The functionality and the execution process of the rule is similar to that of the Lookup Table rule, with the only difference being the rule getting generated automatically.

To create the Lookup Table from Data rules, map the data columns to the business terms manually. The business terms for mapping can be created either manually or automatically based on the data type of the data column.

The inputs for Lookup Table from Data rule can be of either string or numeric data type and the output data type has to be string in any case.

Example

The example assumes a database table containing the details of the employees of a company. On mining the database table using the Lookup Table from Data rule with the data columns "Employee Rank", "Employee Rating" as inputs and "Percentage increase in salary" as output, the Lookup Table rule is generated. Based on the data present in the database table's columns, the number of rules and conditions in the Lookup Table rule are generated automatically. The rule created will specify the range of percentage increase in salary for a specific range of rating and rank.

Log In to the Rule Manager

The Rule Manager runs under CA Aion Rule Manager.

To log in to Rule Manager

1. Click Run from the Windows Start menu and enter the following in the command line

`http://your_machine_name:8080/rmportal`

Note: You must enter your machine name rather than *localhost*. The port may differ, depending on the HTTP port on which you installed CA Aion Rule Manager.

The CA Aion Rule Manager Log In page displays.

2. Enter the User Name in the User Name field.

3. Enter the Password in the Password field.

For more information, see User Names/Passwords.

4. Click Log In.

Once you have logged in, you continue to the Rule Manager portlet.

User Names/Passwords

The following default user name, password combination is provided:

rm admin

Defines the user as a Portal Administrator during the installation. Portal Administrators have special permissions and can access a broad range of Rule Manager functionality, such as creating projects, writing business terms and rules, testing rules, and deploying rule projects as Web services. Default password is not available, you can however, change the password during install.

Note: Contact your administrator to know your user name and password.

Browse Projects and Rule Sets

As you develop Rule Manager Projects, they are listed in the All Projects folder as shown in the following illustration:

The screenshot displays the Aion Rule Manager web application. At the top, there is a navigation bar with 'Home' and 'Help' links. Below this, a status bar indicates the user is logged in as 'rm admin1' with a 'Log Out' link. The main content area is titled 'Rule Manager' and shows a 'Folders' section. Under 'All Folders', there is a list of projects. The projects are displayed in a table with columns for 'Select', 'Title', 'Details', 'Description', 'Last Updated', and 'Actions'. The projects listed are 'PayrollProject', 'Pricing Tier Project', 'SimpleProject', 'Test Project', and 'TLP Insurance'. Each project has a 'Details' icon and a 'Download Later' button. The interface also includes a search bar, a 'Go' button, and a 'View Downloads' link.

| Select | Title | Details | Description | Last Updated | Actions |
|--------------------------|----------------------|---------|-------------|--------------------------|-------------------|
| <input type="checkbox"/> | PayrollProject | | | Sep 17, 2009 11:58:58 AM | Download Later Go |
| <input type="checkbox"/> | Pricing Tier Project | | | Sep 17, 2009 11:58:31 AM | Download Later Go |
| <input type="checkbox"/> | SimpleProject | | | Sep 17, 2009 11:58:04 AM | Download Later Go |
| <input type="checkbox"/> | Test Project | | | Sep 15, 2009 11:38:35 AM | Download Later Go |
| <input type="checkbox"/> | TLP Insurance | | | Sep 17, 2009 12:28:57 PM | Download Later Go |

Note: Click a project's Details icon to explore the project.

Icons list

The following are the icons used in Rule Manager:

| Icon | Graphic |
|------------------------|--|
| Builder |  |
| Default Type |  |
| Duration |  |
| Enum |  |
| Free Editor |  |
| New row |  |
| Rule |  |
| Rule type - Constraint |  |
| Rule Type - Default |  |
| Rule Type - Definition |  |
| Rule Type - Table |  |
| Project - Opened |  |
| Project - Closed |  |
| Rule set - opened |  |
| Rule set - closed |  |
| Term |  |
| Termset |  |
| Type |  |
| Test Case |  |
| Test Icon |  |
| Tree Node |  |
| Vocabulary |  |

Create a Project

Rule Manager organizes rules into a hierarchy of folders. This allows users to organize and manage access to rules efficiently. The project folder, at the top of the hierarchy, groups all the rules based on the purpose of the rules. The rule project contains all business vocabularies necessary for defining the rules and test cases for testing the rules. The rule project also corresponds to a deployed rules web service. The Rule Manager administrator must initially create the project in Rule Manager.

The difference between Project and Collection is as follows:

Project

A project is a deployable set of unit that contains business terms (vocabulary), Rule sets, rules, test cases and can be exposed as web service.

Collection

A collection is a set of any items or content e.g. document, text, images, video files etc.

To define a Rule Manager project

1. Click details icon in All Folders.
All Folders portlet appears.
2. Click Create Project to create a folder.
3. Select a template from the Property Template drop-down. The templates listed are, COLLECTION and PROJECTS.

Select PROJECTS when creating a Rule Manager project.

Note: Collection folder is used to store Projects or other content such as word documents and other external files.

4. Complete the following fields:

Title

Defines the rule title. TLP Insurance for this example.

Description

(Optional) Defines the business purpose of the project. For example, Evaluates a TLP Insurance application and assigns a pricing tier.

Keywords

(Optional) Defines the words that describe the project. For example, customer stability, coverage liability, property risk, and pricing tier.

Notify Me When

(Optional) Notifies the user when the resource (for which the user registers the notification) is updated or deleted. You will receive an email notification if an email address was entered for the user in the EEM Profile.

Note: Description and keywords field content can be used for future searches.

5. Click OK.

A confirmation message appears above the All Folders portlet. The Project folder is created and the folder view appears.

You can create vocabularies, test cases, rulesets, content folders, and content in a project.

Note:

- The tabs Deployment, Administrator and the options Validate, Convert to BRE, Create Vocabulary, Create Test Case are available only for the PROJECTS property template.
- You can also create a rule project by uploading a compressed rule project. For more information, see [Upload a Compressed File](#) (see page 47).

Upload a Compressed File

To upload a compressed file

1. Login to CA Aion Rule Manager
2. Click *All Folders* icon from Folder portlet
3. Click *Add Content*.
4. From *Select Content* page section, click *Browse* adjacent to *Add File:* text box.
5. From File Upload dialog pop-up, select zip file to upload, and click *Open*.
6. Select *Upload and Extract* option, and click Ok.

Content of the zip file will be uploaded in their respective projects and respective message (error or success) will be shown to user.

Business Vocabulary Manager

The Business Vocabulary Manager helps you create, edit and delete vocabularies, types and business terms. Terms are used when creating rules and test cases. You can add types based on the default types.

Create a Vocabulary

A Vocabulary is a collection of terms and types. You can use terms while defining rules and test cases. A term is partly defined by its type.

To create a Vocabulary

1. Click Create Vocabulary in the Project view.
The New Vocabulary page appears.
2. Enter the Vocabulary name in the Name field.
3. (Optional). Enter a Description for the Vocabulary.
4. Click OK.

The vocabulary file is created.

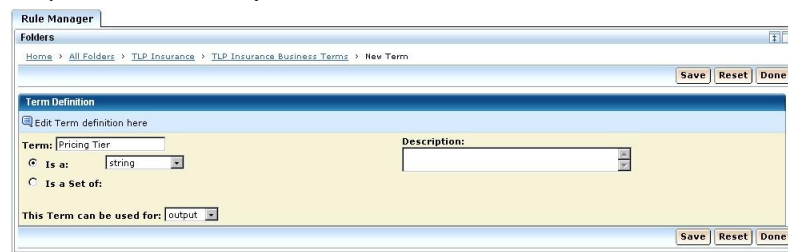
Create Business Terms

You need to create business terms to reflect the business environment. Rules are written to test and change the values of business terms.

To create a term

1. Click New Term in the vocabulary details page.
The Term details window appears.
2. Enter the business term name.
3. (Optional). Enter the description for the term.
4. Select if the term Is a or Is a set of and then select Text from the drop-down list.

The default types are boolean, string, duration, datetime, and numeric. If numeric is selected, the precision value has to be set (precision box allows only whole numbers).



The screenshot shows the 'Rule Manager' application window with the 'Term Definition' dialog box open. The dialog has a 'Term' field containing 'Pricing Tier' and a 'Description' field. Below these, there are two radio buttons: 'Is a:' (selected) and 'Is a Set of:'. To the right of the 'Is a:' radio button is a dropdown menu currently showing 'string'. Below the radio buttons is a label 'This Term can be used for:' followed by a dropdown menu showing 'output'. At the top right of the dialog are 'Save', 'Reset', and 'Done' buttons. At the bottom right are also 'Save', 'Reset', and 'Done' buttons. The background shows a breadcrumb trail: 'Home > All Folders > TLP Insurance > TLP Insurance Business Terms > New Term'.

5. Select Usage of this term.

Usage tells how the deployed rule service uses the term.

As an Internal Term

Specifies that the business term is used only within the context of the project; it is not exposed to an external process or application. The value can be tested and set by rules in the project.

As an Input Term

Specifies that the term represents a value that must be provided to the service from an earlier step in the business process that the service needs to apply the rules. Rules can optionally redefine this term, so if necessary, an Input Term can also provide output value, similar to the way Output Term works.

As an Output Term

Specifies that the term represents something that the rules define and subsequent business process steps can use.

6. Click Save

A confirmation message appears above the definition portlet.

7. Click Done to return to the vocabulary list.
8. (Optional). Repeat steps 1 through 7 to add more business terms.
9. Select Done from the vocabulary list to return to the project folder.

Note: Term names can only contain alpha-numeric and underscore characters. They must begin with an alpha or numeric character. If a Term begins with an underscore, an error message displays and the Term is not saved.

Example

The following table contains list of sample business terms:

| Term | Type | Usage |
|------------------------------|--------|--------|
| Customer Financial Stability | String | Input |
| Property Risk | String | Input |
| Pricing Tier | String | Output |

Edit a Term

You can edit Term details to change the term definitions.

To edit a term

1. Click the Details icon to edit the term.
The term details appear.
2. Modify the required fields. All fields except the Term Name can be edited.
3. Click Save.
A confirmation message appears above the definition pane.

Delete a Term

You can delete a term by clicking on the term in the Vocabulary tab.

Note: If you attempt to delete a term when it is currently used in any rule definitions, the following message is displayed:

Error: Some rules are dependent on the term.

To delete a term

1. Click the Trashcan icon in the row of the item you want to delete.
A dialog appears to confirm the deletion.
2. Click Yes to delete and Cancel to abort the operation.
The term is removed from the term list. A confirmation message appears above the definition pane. An error message appears if the delete operation fails.

Create User Types

You can create a user type and define it as either a Complex Type or an Enumeration. A Complex Type is a collection of named Fields while an Enumeration contains a list of possible values. Types created here are listed with the default types when creating Terms.

To create a user type

1. Click New Type in the vocabulary details page.
Type details are displayed.
2. Enter a name for the type.

3. (Optional). Enter the description for the type.
4. Select one of the following options to define the type attributes

Is composed of

Contains a collection of named fields. You can create multiple named fields.

Is an enumeration with the following values

Contains a list of possible values. You can create multiple values.

More information:

[Define Values For Option Is Composed Of](#) (see page 51)

[Delete a Term](#) (see page 50)

[Define Values For Option Is Composed Of](#) (see page 51)

[Define Values for Option Is an enumeration with the following values](#) (see page 52)

[Using CA Aion Rule Manager](#) (see page 29)

Define Values For Option Is Composed Of

To define values for complex type

1. Enter a Name for the value, decide whether the type is Is a or Is a set of, and select the type.
2. Select the data type. The default types are boolean, string, duration, datetime, and numeric. If numeric type is selected, the precision value has to be set, only whole numbers are allowed.

Note: Type and field names can only contain alpha-numeric and underscore characters. They must begin with an alpha or numeric character. If a Type begins with an underscore or unsupported character, an error message is displayed and the Type is not saved.

3. Select the usage of this row:

As an Input Term

Specifies that the term represents a value that must be provided to the service from an earlier step in the business process that the service needs to apply the rules. Rules can optionally redefine this term, so if necessary, an Input Term can also provide output value, similar to the way Output Term works.

As an Output Term

Specifies that the term represents something that the rules define and subsequent business process steps can use.

As an Internal Term

Specifies that the business term is used only within the context of the project; it is not exposed to an external process or application. The value can be tested and set by rules in the project.

4. Click Save.

System carries out data validation. If validation is successful the type is added to the vocabulary. An error message appears if validation fails.

Note: Click the Insert Row icon in the Action column to add a row, click the Trash icon to delete a row.

Define Values for Option Is an enumeration with the following values

To define values for option is an enumeration with the following values

1. Select the Is an enumeration option button with the following values.

Enter a value in the text box.

2. Click Save.

A confirmation message appears above the definition portlet.

Note: Click the Insert Row icon in the Action column to add a row and the Trash icon to delete a row.

Edit a Type

You can edit the Type details to modify type attributes.

To edit a type

1. Click the Details icon to edit the type.

The type details appear.

2. Modify the required fields. All fields except the Type Name can be edited.

3. Click Save.

A confirmation message appears above the definition portlet.

Delete a Type

You can delete a type by clicking the trash icon.

To delete a type

1. Click the Trashcan icon in the row of the item you want to delete.
A confirmation dialog appears.
2. Click Yes
The type is removed from the type list. An error message appears if the delete operation fails.

Create a Rule Set

You can organize rules by placing them into rule sets. A rule set is a collection of rules that is stored in a project.

To create a Ruleset folder

1. Click Create Rule Set in the Project folder.
The Rule Set screen appears.
2. Select RULESET from the Property Template drop-down.

The screenshot shows a web interface for creating a Rule Set. It is divided into two main panels: 'Properties' and 'Notifications'. The 'Properties' panel has a header with 'Property Templates: RULESET' and 'Template Description: RULESET'. Below this, there are fields for 'Title' (with a hint '* Required'), 'Description', and 'Keywords'. A tip below the keywords field says 'List words that can be used to search for this content.' There are also date pickers for 'ENDDATE' and 'STARTDATE', each with 'Month', 'Day', and 'Year' dropdowns. A 'LOCALTIMEZONE' section has radio buttons for 'True' (selected) and 'False'. A 'PRIORITY' field is also present. The 'Notifications' panel on the right has a section 'Notify Me When' with checkboxes for 'Folder is Updated' and 'Folder is Deleted'.

Note: RULESET is the default Property Template used to create Rule set folder.

3. Complete the following fields:

Title

Defines the rule title. Enter Pricing Tier Rules for this example.

Description

(Optional) Defines the business purpose of the rule set.

Keywords

(Optional) Defines the words that describe the rule set. These words are used for search.

ENDDATE

(Optional) Defines the date when the rule set's validity ends.

LOCALTIMEZONE

(Optional) Specifies whether the effective dates should be based on client-local time or GMT.

Default: True - indicates that Local Time Zone has to be followed.

PRIORITY

(Optional) Defines the preference to be given to the rule. Priority may be stated as a positive or negative integer.

STARTDATE

(Optional) Defines the date when the rule set is launched.

Notify Me When

(Optional) Notifies the user when the resource (for which the user registers the notification) is updated or deleted.

Note: You will receive an email notification if an email address was entered for the user in the EEM Profile.

Note: Rulesets and rules can have effective dates associated with them. Rule Manager will not execute rules if the current date is outside the range of the rule's or containing rule set's effective dates. Enter the title for the rule set.

4. Click OK.

A confirmation message appears.

How to Create a Rule

Once you have created a rule set, you can populate it with rules. There are two ways to create a rule:

- Manually, using the Create Rule wizard and expression builder (the default).
- Automatically, using another data source to extract a rule from data.

Depending on which option you select, the screen layout changes.

How to Create a Rule Manually

The manual rule creation process is presented in a series of steps:

1. [Rule creation](#) (see page 55)
2. [Specify the Rule Type](#) (see page 57)
3. [Select Business Terms \(for Lookup Table Rules only\)](#) (see page 73)
4. [Configure Selected Business Terms \(for Lookup Table Rules only\)](#) (see page 78)
5. [Edit the Rule](#) (see page 79)

Create Rule

After the ruleset folder is created, you create Rules.

To create rules

1. Click Create Rules in the Ruleset folder.

The Rule creation wizard opens.

Create Rule: Edit Properties 1 of 3 [Back] [Next] [Finish] [Cancel]

Home > All Folders > Test 1 Folder for Rules > Rules Folder > Aion

1 Enter Rule Properties 2 Choose Rule Type 3 Define Iteration Variables 4 Add Rule Contents

Properties

Title: [Text Box]

Description: [Text Box]

Keywords: [Text Box]

Version Comments: [Text Box] Tip: List words that can be used to search for this rule.

Author: mukla01

Local Time Zone: ☒

Start Date: [Text Box] dd/MM/yyyy

End Date: [Text Box] dd/MM/yyyy

Priority: 0

Notifications

Notify me when: ☐ Rule is updated.
☐ Rule is deleted.

[Back] [Next] [Finish] [Cancel]

2. Complete the following fields:

Title

Defines the rule title. Enter Pricing Tier Rules for this example.

Description

(Optional) Defines the business purpose of the rule.

Keywords

(Optional) Defines the words that describe the rule set. These words are used for search.

Version Comments

(Optional) Defines the version comments when editing rules.

Default: Initial version

Author

Defines the Author's name. It takes the login user's ID and you cannot edit it.

LOCALTIMEZONE

(Optional) Specifies whether the effective dates should be based on client-local time or GMT.

Default: True - indicates that Local Time Zone has to be followed.

PRIORITY

(Optional) Defines the preference to be given to the rule. Priority may be stated as a positive or negative integer.

Default: 0

STARTDATE

(Optional) Defines the date when the rule is launched.

ENDDATE

(Optional) Defines the date when the rule's validity ends.

Notify Me When

(Optional) Notifies the user when the resource (for which the user registers the notification) is updated or deleted.

Note: You will receive an email notification if an email address was entered for the user in the EEM Profile.

3. Click Choose Rule Type or click Next to select a Rule Type.

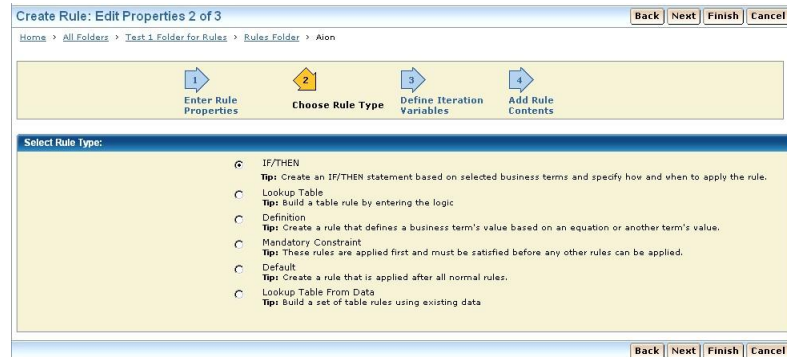
Select a Rule Type

There are five basic rule types which determine the structure of a rule and how that rule is executed.

To select a rule type

1. Select a Rule Type.

Default: IF/THEN



Following are the Rule Types:

IF/THEN

Specifies a simple rule with a single condition and one or more actions associated with the condition. The action(s) typically assign a value to a business term, they can also create an instance of a type and perform set remove, combine operations.

Lookup Table

Build a Table Rule by entering the logic. The table contains one or more rules each with both a condition and an action. The condition tests one or more business terms and the action assigns values to one or more business terms. Although all rules in a Lookup Table must use the same set of business terms, each rule is free to test for different values.

Definition

Specifies how to compute or derive a value, based on some mathematical functions that provide a precise definition for a derived term. A definition rule defines rather than deduces a term's value. A definition rule does not have a condition to be evaluated. Its action part (which contains the actual definition of the term) is always executed.

Mandatory Constraint

Specifies rules with conditions that must be TRUE before the other rules can execute. These rules also check that terms used in the Boolean expression are not unknown. These are rules with conditions that must be TRUE before the other rules can execute. Constraint rules are used to ensure the validity of assumption of the values or relationship among terms. These rules are applied first and must be satisfied before other rules can be applied.

Note: Mandatory constraint rule can be applied ONLY ON 'input' business terms.

Default

Defines a default rule to be used as a last resort to set the term's value when other rules do not resolve the value of a term. This assures that the term always has a value regardless of the results of the other rules. If the term's value is resolved by other rules, then the default rule for the term does not execute.

Note: Default rule can be applied ONLY ON output terms.

Lookup Table from Data

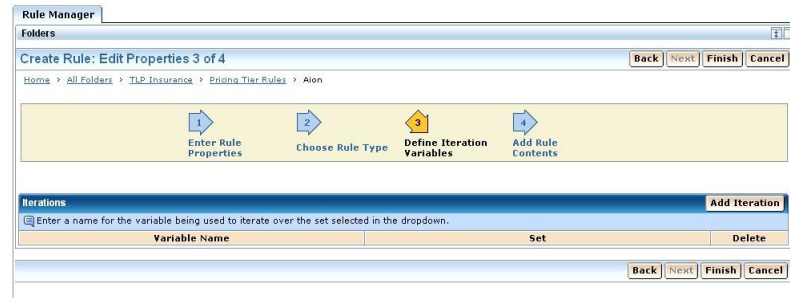
Build a table of rules using existing data.

2. Click Next to define Iteration Variables.
3. Click Finish to save the Rule Type. The Rule Type is saved without defining the Iteration Variables and Rule Contents. The rule however is invalid as it has no contents and Validation on such rules displays an error. You can define them later by editing the rule.

Note: The wizard portlet options change based on the rule selected. For Lookup Table from Data Rule Type, you have to select a Template from the Saved Templates list.

Create Iteration Variables

After selecting the Rule Type, the next step in the wizard is creation of iteration variables. You can use iteration variables to apply a rule to all items in one or more collection business term. For instance, if there is a set of numeric values called ages in the list of terms, an iteration variable called, anAge can be created to iterate over the set. Once created, these variables appear in the Terms branch of the Expression Tree. Iterations are stored in table form.



Note: If you are not defining rules that apply to individual values in collections, you can skip this step and proceed to Adding Rule Contents which is the next step in the rule creation wizard.

Note: The loops in a rule correspond to the number of iteration variables defined here and *not* to the usage of the iteration variable in the rule.

Example:

If a user defines the following two iteration variables:

- iNum on Set of Numbers
- iString on Set of Strings

But, the user uses only one of these in the rule, the rule will still have two loops corresponding to each of the two iteration variables.

To have a cleaner rule execution, it is recommended that the user should delete any unused iteration variables. This can be done by editing iterations using the Edit Iterations option in the rule editor.

Select the Business Terms

Note: This step of the procedure is determined by the rule type you selected and is valid only for Lookup Table Rules.

Expression Builder

You can use the Expression Builder to build an expression from a set of expression fragments. The expression builder consists of the following components:

- Expression Tree
- Terms
- Templates
- Statements
- Constants
- Free Form Editor
- Cursor
- Iteration Variables
- Expression Blocks
- Item Manipulators
- Validation Output
- Iterations

Expression Tree

Contains the expression fragments or items needed to build an expression. Click an item to enter values for the selected item. The items are split in the following categories.

Terms

Contains the business terms defined in the business vocabulary. It also contains Iteration variables if defined in the previous step.

Templates

Contains the following options, Text, Math, Date, Duration, Collection and Statement.

For example, the Text "Begins With" function must act upon a text term and a user entered value. The correct syntax is, "{term} Begins With("User Entered Value")". The template for "Begins With" enters all of the items into the expression together for the user whereas if the user simply enters the "Begins With" statement, the user will also have to enter the term, the parentheses and the value input box to complete the expression. The statement branch gives the user the flexibility of entering a complex expression for the "With" clause. For example: you can enter a term or a text concatenation expression. The template on the other hand inserts an expression with the very basic functionality.

Note: On selecting a template from Templates, an expression is inserted in the expression builder with complete structure of template.

Statements

Contains Text, Math and Collection statement fragments that can be entered in an expression. These fragments are all incomplete on their own and rely on other fragments to complete an expression. These are very powerful as they allow complex expressions to be created. An alternative to using Statements is free form where the user can actually type in the expressions.

Note: On selecting any Statement from this section, partial expression is inserted in expression builder, not complete expression structure.

Constants

Contains constant values that can be entered into an expression such as the math constant "Pi."

Iteration Variables

Contains variables defined in Iterations at the bottom of the Expression Builder. Iterations are stored in table form just below the expression editor. These are variables that serve as an index on a set of items and are used to construct collection expressions such as Select All, Select one. Once created, these variables appear in the Expression Tree under the Iteration Variables branch.

Point-And-Click Editor

The Point-And-Click editor is a viewer for the ordered expression fragments that make up the expression. When the cursor is on the fragment it can be edited or removed. When the cursor is off the fragment, a link containing descriptive text of the item appears. Clicking the link makes the fragment editable. Cursor and item manipulators are available under the expression blocks. You can manipulate the contents in the builder using the Undo, Redo, Delete and Clear options.

Free Form Editor

The Free Form editor is another viewer for the ordered expression fragments that make up the expression wherein the user can type the expression using the expression builder features except disabled features. You use the rule editor language to create the free form expressions. You can switch between free form and expression builder except when the syntax in the free form is incorrect.

You can toggle between the Point-And-Click Editor and the Free-Form Editor. The expression appears in both with some limitations when switching from free form to Point and Click mode.

Expression Blocks

Expression blocks are a set of buttons provided below the Point-And-Click or Free Form builder. They contain comparison operators, math operators, logical operators, parentheses, dot (for complex terms), space and return. There is also a button to enter a term which in the Point-And-Click builder enters a drop-down starting at the first level of terms. Additional term levels can be inserted using the dot button. Complex terms are inserted by right-clicking the inserted term and selecting the required option from the context menu. String or numeric value can be entered using the value button. The editor determines if a Boolean value is appropriate else it inserts a true/false drop-down. The template type can be chosen using the template option.

Validation Output

Validation output is a message box available to the right of the Cursor and Item manipulators. Validation check is carried out whenever the user enters or changes an item in the Point-And-Click editor or when the user toggles between the Point-And-Click and the Free Form editor. If the validation check is successful, the validation output box displays Valid: No Errors else it displays Invalid: Details.... To view the error message, click on the Details link.

Rule Editor Language

The rule editor language is used in the free form and the same language is used in the condition and action expression in the various rule types.

The rule editor language has the following structure:

- Terms, iteration variables, enumerations and enumeration values are all enclosed in curly braces i.e. "{" and "}"

Example

```
{the Cart.theCustomer.iiCategory} = {IICustomerCategory.GOLD}
```

- DateTime constants are displayed as DateTime()
- Duration constants are displayed using Duration()

Example

```
{a duration } = Duration(2Y45DT5m)
```

- String constants are displayed using quotes
- Number and Boolean constants are displayed as is

Example

```
: { a Bool } = True
```

The rule editor language is defined by the following BNF:

IFTHEN_RULE

The ifthen rule has the following syntax:

```
:: IF 'GeneralExpr' THEN 'ThenExpr'
```

Note: 'GeneralExpr' is valid for if condition and terms in if/then, LookupTable Rule, and Constraint rules. 'ThenExpr' is valid for then action in if/then, Lookup Table, Definition, and Default rules.

GeneralExpr

```
::= BooleanTerm ( 'OR' BooleanTerm )*
```

BooleanTerm

```
::= RelTerm ( 'AND' RelTerm )*
```

RelTerm

```
::= BaseExpr [CompRHS]
```

```
::= BaseExpr 'MODULUS' BaseExpr [CompRHS]
```

Note: Returns numeric

```
::= BaseExpr 'TO THE POWER OF' BaseExpr [CompRHS]
```

Note: Returns numeric

```
::= BaseExpr 'TO DEGREES' [CompRHS]
```

Note: Returns numeric

```
::= BaseExpr 'TO RADIANS' [CompRHS]
```

Note: Returns boolean

```
::= BaseExpr 'BEGINS WITH' BaseExpr [CompRHS]
```

Note: Returns boolean

```
::= BaseExpr 'CONTAINS' BaseExpr [CompRHS]
```

Note: Returns boolean

```
::= BaseExpr 'ENDS WITH' BaseExpr [CompRHS]
```

Note: Returns boolean

CompRHS

```
::= CompOp BaseExpr
```

```
::= Includes BaseExpr
```

Note: CompOP is comparison operators such as: >, <, >=, <=, =,

Includes

BaseExpr

`::= BaseTerm (BaseTermOp BaseTerm)*`

Note: BaseTermOp is '+' or '-'

BaseTerm

`::=BaseFactor (BaseFactorOp BaseFactor)*`

Note: <BaseFactorOp> is '*' or '/'

BaseFactor

`::= LiteralConstant`
`::= SetLiteral`
`::= UnaryExpr`
`::= ParenthesizedExpr`
`::= StringExpr`
`::= MathExpr`
`::= TermId`
`::= TermId 'IS' 'UNKNOWN'`

Example

`{theCart.theCustomer.vouchers } is UnKnown`

`::= SetExpr`
`::= DateDurExpr`

SetLiteral

`::= 'SET' '(' ')'`

Note: SET() returns an empty set so can be compared, assigned to another set.

Example

`{theCart.theCustomer.vouchers } = Set ()`
`::= 'Make Set' '(' SetValue (',' SetValue)* ')'`

Note: Make Set makes a set of all basic types: boolean, numeric, text, datetime, and duration. It can be assigned to or compared with another set.

Example

`{theCart.theCustomer.numVouchers } = Make Set (12, 34, 56)`

SetValue

`::= [+/-]LiteralConstant`

ParenthesizedExpr

`::= '(' GeneralExpr ')'`

UnaryExpr

```
::= UnaryOp BaseFactor
```

UnaryOp

```
::= '+'  
::= '-'  
::= 'NOT'
```

StringExpr

```
:: BaseExpr 'Begins With' BaseExpr
```

Note: Returns Boolean.

Example

```
{ the IIWork.itemGroup.category} Begins With ("a") = true  
:: BaseExpr 'Ends With' BaseExpr
```

Note: Returns Boolean.

Example

```
{ theCart.boolTerm} = { theIIWork.itemgroup.category} Ends With {theCart.textTerm}  
::= 'Replace' BaseExpr 'With' BaseExpr in BaseExpr
```

Note: Returns Text.

Example

```
{theCustomer.name} = Replace "Mr" with "Ms" in {theCustomer.name}  
::= 'Length of' BaseExpr
```

Note: Returns numeric.

Example

```
{theCustomer.size} = Length Of ( {theCustomer.firstName} + {theCustomer.lastName}  
)  
::= 'Substring' 'OF' BaseExpr 'FROM' BaseExpr** ['TO' BaseExpr**]
```

Note: Returns Text.

Example

```
{theIIWork.itemGroup.category} = Substring Of {theIIWork.itemGroup.category} from 2  
To 5  
::= 'Index' 'OF' BaseExpr 'IN' BaseExpr ['FROM' BaseExpr**]
```

Note: Returns Numeric.

Example

```
{theCustomer.size} = Index Of "ab" in {theIIWork.itemGroup.category}
```

MathExpr

```
::= 'ABSOLUTE VALUE OF' BaseExpr
```

Example

```
{ theCart.netPurchases} = Absolute Value of ( {theCart.netPurchases + {  
theCart.totalPurchases} )  
::= 'ARC SINE' 'OF' BaseExpr  
::= 'ARC COSINE' 'OF' BaseExpr
```

Example

```
Arc Cosine Of ( {theIIWork.newVoucher.value} + 67) > 45 AND (Arc Sine of {  
theIIWork.itemcount } < 23)  
::= 'ARC TANGENT' 'OF' BaseExpr  
::= 'TANGENT' 'OF' BaseExpr
```

Example

```
Arc Tangent Of ( {theIIWork.newVoucher.value} ) > 45 OR (Tangent of {  
theIIWork.itemcount } < 50 )  
::= 'CEILING' 'OF' BaseExpr  
::= 'FLOOR' 'OF' BaseExpr
```

Example

```
Ceiling Of ( {theCart.Customer.id} ) > 23 And Floor Of (34.34) < 35  
::= 'COSINE' 'OF' BaseExpr  
::= 'SINE' 'OF' BaseExpr
```

Example

```
Cosine Of ( {theCart.angle} ) = {theCart.anotherAngle} OR Sine of ( {theCart.angle}  
) < { theCart.secondAngle}  
::= 'EXPONENTIAL' 'OF' BaseExpr  
::= 'LOGARITHM' 'OF' BaseExpr
```

Example

```
( {theCart.Customer.id} ) = Exponential Of ( {theCart.Customer.id} ) And Logarithm  
Of ( 2.3 ) = { theCart.numeric}  
::= BaseExpr 'To The Power Of' BaseExpr
```

Example

```
{theCart.Customer.id} To the Power Of ( 2 ) >= 44  
::= 'SQUARE ROOT' 'OF' BaseExpr
```

Example

```
Square Root Of ( 64 ) <= 5  
::= BaseExpr 'To Degrees'
```

Example

```
( {theCart.Customer.id} To Degrees ) >= {theCart.Customer.id}  
::= BaseExpr 'to Radians'  
( {theCart.Customer.id} To Radians ) >= {theCart.Customer.id}
```

Note: All of the above Math Expressions return numeric.

DateExpr

```
::= Specific Date  
::= 'Year From' '('BaseExpr')'  
::= 'Month From' '('BaseExpr')'  
::= 'Day From' '('BaseExpr')'  
::= 'Hour From' '('BaseExpr')'  
::= 'Minute From' '('BaseExpr')'  
::= 'Second From' '('BaseExpr')
```

Note: All of the above except first one return numeric.

Example

```
Date (5/6/07 6:49:00 AM) >= {shoppingDate} And Year  
From ( {shoppingDate} ) = 2007 And Month From  
( {shoppingDate} ) = 5 And DayOfMonth From  
( {shoppingDate} ) = 6 And DayOfWeek From  
( {shoppingDate} ) = 5 And Hour From  
( {shoppingDate} ) = 6 And Minute From  
( {shoppingDate} ) = 49 And Second From  
( {shoppingDate} ) = 0
```

DurationExpr

```
::= Specific Duration  
::= 'Year Count From' '('BaseExpr')'  
::= 'Month Count From' '('BaseExpr')'  
::= 'Day Count From' '('BaseExpr')'  
::= 'Hour Count From' '('BaseExpr')'  
::= 'Minute Count From' '('BaseExpr')'  
::= 'Second Count From' '('BaseExpr')
```

NOTE: All of the above except first one return numeric.

Example

Duration (34Y2M23DT20h45m28s) >= Duration (45YT678m) And
Year Count From (Duration (34Y2M23DT20h45m28s)) = 34 And
Month Count From (Duration (34Y2M23DT20h45m28s)) = 2 And
Day Count From (Duration (34Y2M23DT20h45m28s)) = 23 And
Hour Count From (Duration (34Y2M23DT20h45m28s)) = 20 And
Minute Count From (Duration (34Y2M23DT20h45m28s)) = 45 And
Second Count From (Duration (34Y2M23DT20h45m28s)) = 28

SetExpr

::= 'Number Of' BaseExpr

Note: Returns numeric.

::= 'Sum Of' BaseExpr [Using GeneralExpr]

Note: Returns numeric.

::= 'Avg Of' BaseExpr [Using GeneralExpr]

Note: Returns numeric.

::= 'stddev Of' BaseExpr [Using GeneralExpr]

Note: Returns numeric.

Example

Number Of ({theCart.theCustomer.vouchers}) And
Sum Of ({theCart.theCustomer.vouchers}) Using ({theCart.theCustomer.vouchers})
>= 34 And
Avg of ({theCart.items}) Using {theCart.items.price} And
{theCart.stdDevPrice} = StdDev of ({theIIWork.filteredVouchers}) Using
{theIIWork.filteredVouchers.value}
::= 'Max in' BaseExpr [Using GeneralExpr] ['default Value is' GeneralExpr**]

Note: Returns an element of the same type as the set.

::= 'Min in' BaseExpr [Using GeneralExpr] ['default Value is' GeneralExpr**]

Note: Returns an element of the same type as the set.

Example

Max in ({theIIWork.sortedVouchers})
Using the {theIIWork.sortedVouchers.value} = {theCart.voucher} And
{theCart.theCustomer.anItem} = Min
in ({theCart.items})
Using {theCart.items.price}
::= 'SORT' Identifier 'IN' BaseExpr SortLevel+ 'END'

Note: Returns a sorted set.

SortLevel

```
::= 'USING' GeneralExpr Order [Ascending|Descending]
```

Example

```
{theCart.items } = Sort {p } In {theCart.items}
Using {p.price} Order ASCENDING Using ({p.desc})
Order DESCENDING End
::= SET_SELECT_ALL
```

Note: Returns a set

```
::= SET_SELECT_ONE
```

Note: Returns a single element from the set

```
::= 'SubSet' 'OF' BaseExpr 'FROM' BaseExpr*** ['TO' BaseExpr***]
```

Note: Returns a sub set

Example

```
{theIIWork.filteredVouchers } = Subset Of
{theCart.theCustomer.vouchers} From 2 to 4
***-Must evaluate to a numeric expression
**-Default value required only if set is of simple type
```

SET_SELECT_ALL

```
:: 'Select All' BINDVARIABLE 'IN' BaseExpr 'Such That ( ' GeneralExpr )' ['Return'
GeneralExpr]
```

The screenshot shows a rule editor interface. At the top, a text area contains the rule definition for SET_SELECT_ALL:

```
{theCart.items } = Select All {p} In {theCart.items}
Such That
( {p.price} < 23)
```

Below the text area is a toolbar with various operators and functions. The 'Insert Expression Blocks' section includes buttons for '<', '<=', '<>', '=', '>', '>=', 'Includes', '+', '-', '*', '/', 'And', 'Or', 'Not', '(', ')', and '.'. There are also buttons for 'Space', 'Return', 'Value', and 'Template'. A dropdown menu is open, showing 'shoppingDate', 'theCart', and 'theIIWork', with 'theCart' selected.

Below the toolbar is a row of buttons: 'Reset', 'Redo', 'Delete', 'Clear', and a green checkmark icon with the text 'Valid: No Errors'.

At the bottom, there is a table titled 'Iterations' with a column 'Add Iteration'. The table has three columns: 'Variable Name', 'Set', and 'Delete'. The first row shows 'p' in the 'Variable Name' column, 'theCart.items' in the 'Set' column, and a trash icon in the 'Delete' column.

SET_SELECT_ONE

```
:: 'Select One' BINDVARIABLE 'IN' BaseExpr 'Such That ( ' GeneralExpr ' ) [Return'
GeneralExpr] 'default value is' GeneralExpr
```

Note: If return clause is specified, the data type of return clause and default clause should be same. The data type of default GeneralExpr should be same as GeneralExpr.

Code Editor:

```
{theIIWork.filteredVouchers } = Select One {a Voucher} In
{theCart.theCustomer.vouchers}
Such That
( {a Voucher.value} <> 23)
default Value Is 0
```

Insert Expression Blocks:

Operators: < <= <> = > >= Includes + - * / And Or Not { } .

Buttons: Space Return Value shoppingDate theCart theIIWork Template

Buttons: Reset Redo Delete Clear Valid: No Errors

Iterations:

| Variable Name | Set | Delete |
|---------------|------------------------------|--------|
| p | theCart.items | |
| a Voucher | theCart.theCustomer.vouchers | |

Note: SET_SELECT_ONE, SET_SELECT_ALL, and SORT require iteration variable to be created at the bottom of the builder.

ThenExpr

```
:: TermId '=' GeneralExpr
:: TermId '=' InstMakeStmt | SetLiteral
:: Statement
```

Statement

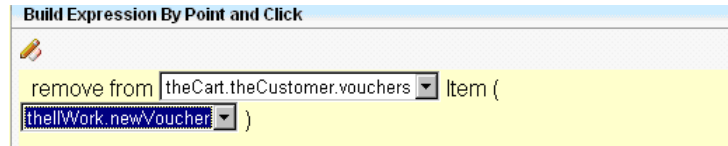
```

:= CollCombineStmt
:= CollRemoveStmt
CollRemoveStmt::
'Remove From SET 'Item' GeneralExpr

```

Note: Doesn't return anything, modifies the set inline.

Can only be used in then action for if/then rule.

**CollCombineStmt::**

```
'Combine Into SET 'Item' GeneralExpr
```

Note: Doesn't return anything, modifies the set inline.

Can only be used in then action for if/then rule.

Example

```

combine into {theIIWork.sortedVouchers}
Item ( { theIIWork.newVoucher } )

```

InstMakeStmt::

```
'Make Instance of' <termType>
```

Note: Creates an instance of type "termType" and returns it. "TermType" has to be a user defined type.

Example

```
{ theIIWork.newVoucher } = Make Instance Of {Voucher}
```

Select Business Terms for If/Then Rule Type

The If Then Rule is structured such that there is only one If condition and one or more Then Actions. The entire rule contents can be saved or reset (revert to last save) by clicking on the Save or Reset buttons that appear above the table on the right-hand side of the screen.

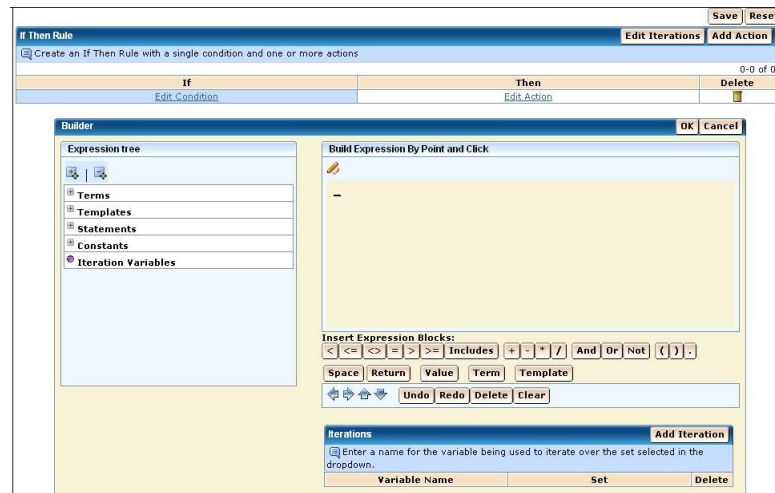
Note: Before you create this rule, we recommend that you go through the Expression Builder topic to get familiar with its components.

To select business terms

1. Click the If condition or Then action you want to edit.

The Expression Builder interface appears.

Note: Click Add Action to add a Then Action row. Click on the trash can icon to delete a row. User cannot delete the action row if only one action row is present.



2. Create the expression in the Expression Builder.
3. Select the term(s) by clicking Terms (plus sign) in the expression tree area.
Terms are displayed. Click on a term to place it in the Point and Click editor.
Follow this procedure to insert other options listed in the Expression tree.
4. Click the required option from the Insert Expression Blocks area to place it in the Point and Click editor.
5. Click Add Iteration to add Iteration variables.
Enter the Variable Name and select the Set from the drop-down list.
6. Click OK to place the expression in the rule table.

The expression builder carries out a validation of the rule. If the validation is successful, a No Errors message appears, else an Invalid message appears.

Note: To view the error message, click on Details... in the Invalid message box.

7. Click Save to save the rule.

A confirmation message appears.

Select Business Terms for Lookup Table Rule Type

To create a Lookup Table Rule Type, follow these steps.

Note: Before you create this rule, we recommend that you go through the Expression Builder topic to get familiar with its components.

To select business terms for Lookup Table Rule Type

1. Select the business term(s) from the Available Business Terms window to be used in the If part of the rule.

The If conditions let you choose which business terms are to be tested in the rule.

Click Edit Vocabulary to create Terms and Types, if needed.

The screenshot shows a software interface for selecting business terms. On the left, the 'Available Business Terms' window lists 'Property Risk', 'Customer Financial Stability', and 'Pricing Tier'. Below this list is a partially visible 'TLP Insurance Business' window and an 'Edit Vocabulary' button. In the center, there are 'Add To' and 'If' buttons. On the right, the 'Selected Business Terms' window contains two sections: 'If Condition Business Terms' which lists 'Customer Financial Stability' and 'Property Risk', and 'Then Result Business Terms' which lists 'Pricing Tier'. At the bottom, there is an 'Iterations' section with a text input field and an 'Add Iteration' button.

2. Click If.

The business term(s) populates in the If Condition Business Terms window.



3. Select the business term(s) from the left to be used in the Then part of the rule.

The THEN condition lets you choose the business terms you will be assigning values to depending on the conditions tested.

If additional business terms and types are needed, click Edit Vocabulary to navigate to the Define the Business Terms page.

4. Click Then.

The business term(s) populates in the Then Result Business Terms window.

5. Use  and  arrows to change the order of the If Condition or Then Result business terms. To delete a business term, click the Trash icon.
6. Click Finish and then edit a rule, if you are satisfied with the order in which the conditions are listed.
7. Click Next to configure selected business terms, if you want to specify how the business terms in the If part of the rule are compared to other business terms, or whether the terms in the If and Then terms are defined by an expression.

If additional business terms and types are needed, click Edit Vocabulary to navigate to the Define the Business Terms page.

Note: Make sure that you put the terms in the correct order before proceeding. You cannot change the order later, or add or remove terms.

8. Click Add Iteration to add Iteration variables.

Enter the Variable Name and select the Set from the drop-down list.

If additional business terms and types are needed, click Edit Vocabulary to navigate to the Define the Business Terms page.

9. Click Finish to enter the If Then rule conditions.
10. Click Save to store the rule.

A confirmation message appears.

More information:

[Create Business Terms](#) (see page 48)

[Configure Selected Business Terms](#) (see page 78)

Select Business Terms for Definition Rule Type

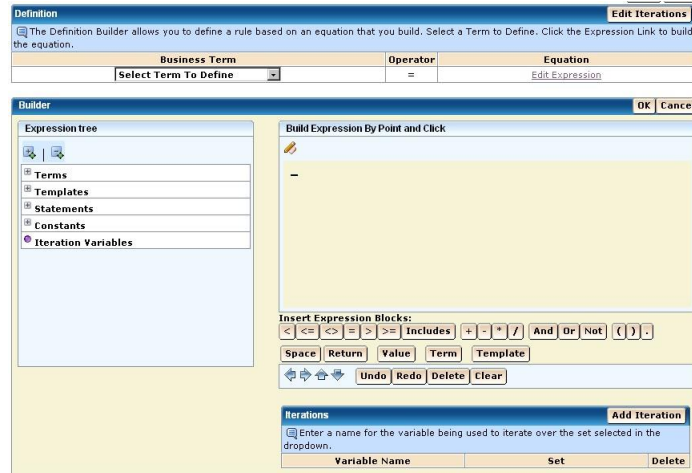
To create a Definition Rule Type, follow these steps.

Note: Before you create this rule, we recommend that you go through the Expression Builder topic to get familiar with its components.

To select business terms for the Definition rule type

1. To select Business Term click on Select Term to Define.

Business Terms drop-down appears.



2. Select the term to define.
3. Click on Edit Expression in the Equation column.
The Expression Builder interface appears.
4. Create the expression in the Expression Builder.
5. Select the term(s) by clicking Terms (plus sign) in the expression tree area.
Terms are displayed. Click on a term to place it in the Point and Click editor.
Follow this procedure to insert other options listed in the Expression tree.
6. Click the required option from the Insert Expression Blocks area to place it in the Point and Click editor.
7. Click Add Iteration to add Iteration variables.
Enter the Variable Name and select the Set from the drop-down list.
8. Click OK to place the expression in the rule table.
The expression builder carries out a validation of the rule. If the validation is successful, a No Errors message appears, else an Invalid message appears.
9. Click Save to save the rule.

A confirmation message appears.

Note: To view the error message, click on Details... in the Invalid message box.

Select Business Terms for Mandatory Constraint Rule Type

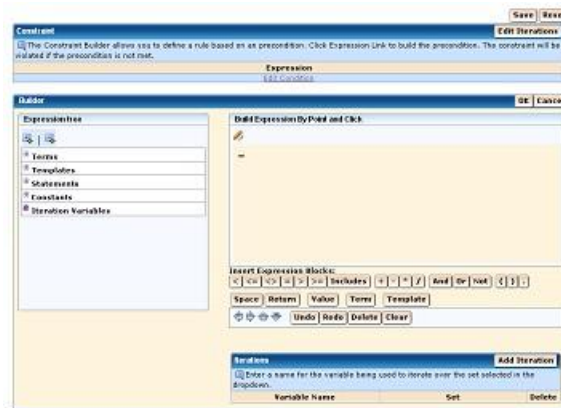
To create a Mandatory Constraint Rule Type, follow these steps.

Note: Before you create this rule, we recommend that you go through the Expression Builder topic to get familiar with its components.

To select business terms for the Mandatory Constraint rule type

1. Click on Edit Expression.

The expression builder interface appears.



2. Create the expression in the Expression Builder.
3. Select the input term(s) by clicking Terms (plus sign) in the expression tree area.

Terms are displayed. Click on a term to place it in the Point and Click editor.

Follow this procedure to insert other options listed in the Expression tree.

4. Click the required option from the Insert Expression Blocks area to place it in the Point and Click editor.
5. Click Add Iteration to add Iteration variables.

Enter the Variable Name and select the Set from the drop-down list.

6. Click OK to place the expression in the rule table.

The expression builder carries out a validation of the rule. If the validation is successful, a No Errors message appears, else an Invalid message appears.

7. Click Save to save the rule.

A confirmation message appears.

Note: To view the error message, click on Details... in the Invalid message box.

Select Business Terms for Default Rule Type

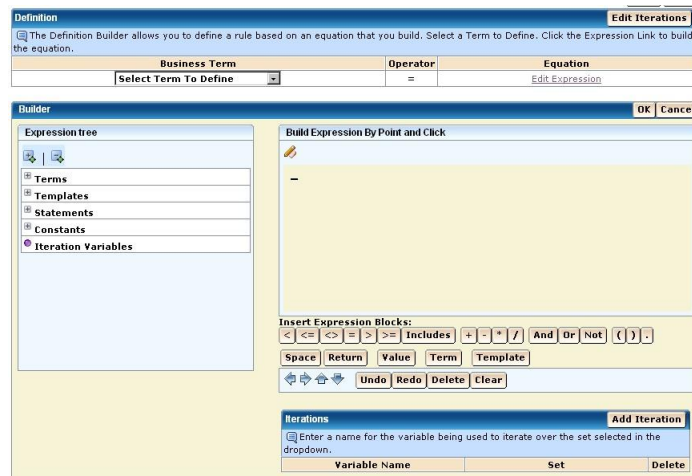
To create a Default Rule type, follow these steps.

Note: Before you create this rule, we recommend that you go through the Expression Builder topic to get familiar with its components.

To select business terms for the Default rule type

1. To select Business Term click on Select Term to Define.

Business Terms drop-down appears.



2. Select the output term to define.
3. Click on Edit Expression in the Equation column.
The Expression Builder interface appears.
4. Create the expression in the Expression Builder.
5. Select the term(s) by clicking Terms (plus sign) in the expression tree area.
Terms are displayed. Click on a term to place it in the Point and Click editor.
Follow this procedure to insert other options listed in the Expression tree.
6. Click the required option from the Insert Expression Blocks area to place it in the Point and Click editor.
7. Click Add Iteration to add Iteration variables.
Enter the Variable Name and select the Set from the drop-down list.

8. Click OK to place the expression in the rule table.

The expression builder carries out a validation of the rule. If the validation is successful, a No Errors message appears, else an Invalid message appears.

9. Click Save to save the rule.

A confirmation message appears.

Note: To view the error message, click on Details... in the Invalid message box.

Configure Selected Business Terms

Note: This step applies only to Lookup Table Rule Type, where the business terms are compared to other business terms, or the results are complex expressions.

This is an optional step. You can specify the conditions necessary in order to perform the desired results, and define how each business term is used for the rule.

If the value of a business term is compared to that of another business term, follow these steps:

1. Select the business term in the Selected Business Terms window (under If Conditions).
2. Click Compare this term to another term in the Condition Input section of the Term Settings.
3. Click Select.
4. Select a business term(s) in the Available Business Terms window.

If additional business terms are needed, click Edit List to continue to the Define the Business Terms page.

5. Click OK.

If a business term is assigned a value by means of an equation, rather than being assigned a specific value, follow these steps:

1. Select the business term in the Selected Business Terms window (under If Conditions Then Actions).
2. Check Use an equation to define the condition or action.
3. Click Finish and then edit the rule.

Note: If multiple conditions are specified, they must all be met in the order in which they are shown.

More information:

[Create Business Terms](#) (see page 48)

Edit a Rule

The previous steps have created an entry for the rule in CA Rule Manager. The next step involves defining the rule contents without which the rule (other than Table rule) is invalid. Now you must edit the rule. Follow the instructions for the specific rule type.

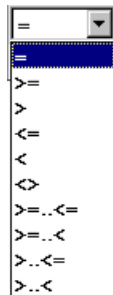
Edit a Lookup Table Rule

The Lookup Table rule editor lets you enter the rule in the form of a table. Each row in the table specifies a list of conditions on the *if* business terms. If these terms are satisfied, it causes the rule to take the action specified in the *then* columns of the row. With an If/Then rule, once a row has been satisfied, no further rows are processed.

After a Lookup Table rule is created, the Contents tab for the new rule displays.

To edit a Lookup Table rule


1. From the drop-down list, select an operator to apply to the Input item.



Note: Equation structures vary, depending upon the type of business term defined. The drop-down lists presented in this step are sensitive to the rule's structure.

2. If you have chosen to enter a value for this term, the text field to the right of the operator is enabled. Enter the value to compare against.



3. Click  to split the row and to reproduce the Input item.



Rows that split to the right of the first column apply all the business term conditions to the left when processing the row. This allows you to reproduce business rules specified as tables.

Note: Once you have multiple rows in a table, the trash can icon appears, allowing you to delete rows of the table.

| If | | Then | |
|---|-----------|--------------|--|
| Customer Financial Stabil... | | Pricing Tier | |
| = | Excellent | | |
|  | | | |
| = | Average | | |
|  | | | |

4. Do *one* of the following:

- a. If you choose to enter a value for this term in the rule configuration, the text field under the THEN column is enabled. Enter the value for the business term in the text box.

| If | | Then | |
|---|-----------|--------------|--|
| Customer Financial Stabil... | | Pricing Tier | |
| = | Excellent | A | |
|  | | | |
| = | Average | C | |
|  | | | |

- b. If you have chosen to use an equation, a link labeled Edit appears under the IF or THEN column. Click this link to define the equation, using the same edit procedure as for Definition Rules (following).

Once you have created an equation for a business term, it appears to the left of the Edit link.

5. Continue adding If/Then rows until the rule is complete.
6. Click Save.

Edit a Definition Rule

After a Definition rule is created, the Contents tab for the new rule displays.

To edit a Definition rule

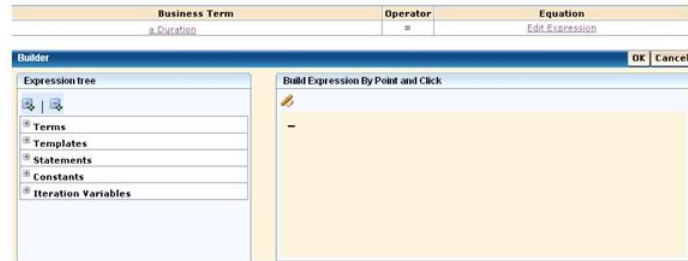
1. On the Contents tab, select an item from the drop-down list.

Note: Equation structures vary, depending upon the type of business term being defined. The drop-down lists presented in this step are sensitive to the type of the rule you are defining.

2. Select the Business Term.

- Click Edit Expression to define the expression.

The expression editor appears.



- Define the expression.
- Click OK to save the expression.
- Click Save to save the rule definition.
- Click Done to return to rule set folder.

Edit a Mandatory Constraint Rule

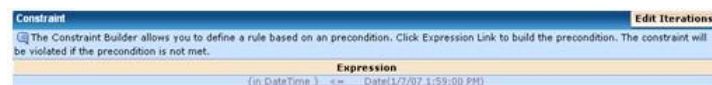
After a Mandatory Constraint rule is created, the Contents tab for the new rule displays.

To edit a Mandatory Constraint rule

- On the Contents tab, you define a rule based on an equation that you build. Select an operator from the drop-down list.
- Click the Edit Condition link.
- Build the Boolean condition.



- Click OK to save the condition.



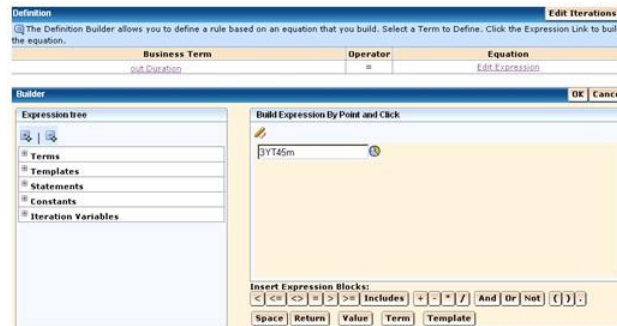
- Click Save to save the rule definition.
- Click Done to return to rule set folder.

Edit a Default Rule

After a Default rule is created, the Contents tab for the new rule displays.

To edit a Default rule

1. Select an output term.
2. Click Edit expression to define the default value.
3. The expression editor displays.



4. Define the expression.
5. Click OK to save the expression.
6. Click Save to save the rule definition.
7. Click Done to return to rule set folder.

How to Create a Rule Automatically

The automatic rule creation process is presented in a series of steps:

- [Rule Creation](#) (see page 55)
- [Select Rule type as Lookup Table from Data](#) (see page 57)
- [Select database, table, and data columns](#) (see page 84)
- (Optional). [Adjust column and rule building settings](#) (see page 85)
- [Map the data columns used to business terms](#) (see page 86)

Note: Before you can use the Rule Manager to create rules from data, make sure that the administrator has set up the connections to the data sources. For more information on setting up data sources, see the *Administration Guide*.

More information:

[Lookup Table from Data Rule](#) (see page 43)

Create Rules from Existing Data

You can create a rule automatically, using a data source and extract rules from existing data.

To create a Lookup Table from Data Rule type

1. Select the Lookup Table from Data rule type.

Select Rule Type:

- ☐ IF/THEN
Tip: Create an IF/THEN statement based on selected business terms and specify how and when to apply the rule.
- ☐ Lookup Table
Tip: Build a table rule by entering the logic.
- ☐ Definition
Tip: Create a rule that defines a business term's value based on an equation or another term's value.
- ☐ Mandatory Constraint
Tip: These rules are applied first and must be satisfied before any other rules can be applied.
- ☐ Default
Tip: Create a rule that is applied after all normal rules.
- ☒ Lookup Table From Data
Tip: Build a set of table rules using existing data.

Saved Templates:

☐ Use Selected Template

Back Next Finish Cancel

2. (Optional). Select a template from the list of previously saved templates. At the end of the wizard you can save your selections to be added to this list.
3. A template lets you save any settings you make during the set up of this rule (when creating future rules).

Note: Click Next as initially there are no templates.

4. Click Use Selected Template if you want to use the template that you have selected from the Saved Templates list.
5. Click Next to select Database, Table, and Data Columns.
6. Select Data Source from the Database drop-down and Table from the Table drop-down.

Select Data Columns

Database: CPAS_ODBCJDBC

Table: SalesData

Available Data Columns:

- ssn
- rrspamt
- list
- Field2
- Field1
- pl1
- sex
- age
- salary
- RawConsultData
- SalesData
- Scextra
- Schule
- Scshift
- Scsquare
- Vendors
- VideoSample
- Weatherdata
- WinConsultResult
- Wine

Selected Data Columns:

In IF conditions, Business Terms specify how to get to results. In THEN Results, Business Terms specify what to do.

If Conditions

Then Result

Back Next Finish Cancel

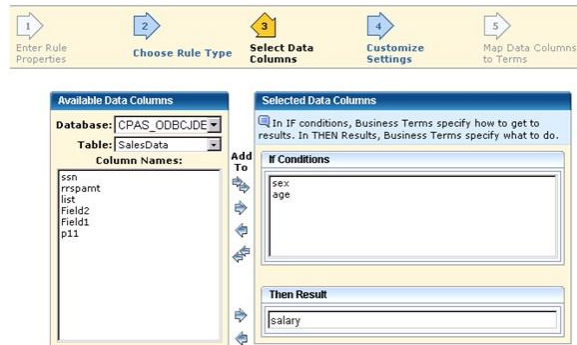
Select Database, Table, and Data Columns

To select Database, Table, and Data Columns

1. Under Available Data Columns, from the Database drop-down list, select one of the available databases.



The available tables in that database are refreshed.

2. Select a table from the Table drop-down list. The column names are populated automatically from the table you select. At this point they are only column names - they are not (yet) business terms.



3. Select the column name(s) from the Column Names window to be used in the If part of the rule.

The If conditions let you specify what is necessary in order to affect the selected results.

4. Click the Select arrow . Click the Select All arrow  to select every column name in the list.
5. Select the column name from the Column Names window to be used in the Then part of the rule.

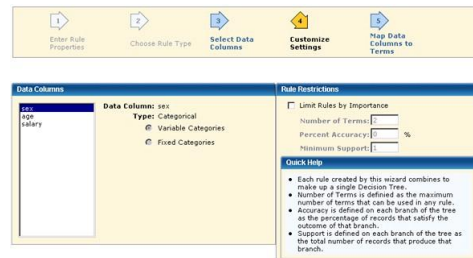
The Then condition lets you choose the business terms you will be assigning values to depending on the conditions tested.

6. Click Next to adjust Column and Rule Building Settings.

Adjust Column and Rule Building Settings (Optional)

To adjust column and rule building settings

1. Click each column under Data Columns to view information about the analysis.



In this example, sex is classified as Categorical data. This is because the database can contain only two values: male and female. Age is treated as a continuous variable (it can have values from 0 to around 100).

Note: Rule Manager examines all the values in a data column. By default, if there are five or fewer distinct values, then the column is considered Categorical. This default can be adjusted; for example, if you frequently process a numeric code for Month (12 distinct values that should be treated as categories, not ranges). The default is set by the value of **rulesFromData.distinctValuesLimit** in the cpaa.properties file. This property file is typically located in the following directory:

```
jboss\server\default\deploy\arm.ear\arm.war\WEB-INF\classes
```

The Rule Manager lets you to set fixed categories or variable categories. This example shows that (by default) three age ranges have been automatically determined by the Rule Manager (by examining the statistics for the column - the mean and standard deviation).

2. Select Variable Categories, to create variable categories (for example, for age).
3. Select Fixed Categories, to create fixed categories.
4. Select Limit Rules by Importance, if you want to create restriction for the rule.

If you select the Limit Rules by Importance checkbox, you can:

- Enter the Number of Terms for the rule, if you selected the Limit Rules by Importance check box
 - Enter the Percent Accuracy for the rule, if you selected the Limit Rules by Importance check box.
 - Enter the Minimum Support needed for the rule, if you selected Limit Rules by Importance.
5. Click Next to map data columns to business terms.

Map the Data Columns to Business Terms

To map the data columns to business terms

1. Under Data Columns, select all of the columns you are using.

The screenshot shows the 'Map Data Columns to Terms' step of a wizard. At the top, there are five numbered steps: 1. Enter Rule Properties, 2. Choose Rule Type, 3. Select Data Columns, 4. Customize Settings, and 5. Map Data Columns to Terms. The main area is divided into three sections. On the left, 'Data Columns' contains a list with 'age', 'salary', and 'sex'. Below this list is an 'Add to Business Terms' button. In the middle, 'Business Terms' is an empty list. Below it is a 'New Business Term' field with an 'Add' button and a 'Vocabulary' dropdown menu. On the right, the 'Mappings' section is empty, with a 'Map Terms' button between the 'Data Columns' and 'Business Terms' sections. At the bottom, there is a 'Save Template' section with a checkbox and a text field for the template name.

2. Click Add to Business Terms if necessary. This creates new entries in the Business Terms.
3. Select the business terms to map (make sure they are also selected in the first column as well).
4. Click Map Terms.

The mappings display in the Mappings window.

This screenshot shows the same wizard step as the previous one, but after the 'Map Terms' button has been clicked. The 'Data Columns' and 'Business Terms' lists remain empty. The 'Mappings' window on the right now contains three entries: 'age->age', 'salary->salary', and 'sex->sex'. The 'Map Terms' button is still present between the two lists. The 'Save Template' section at the bottom is unchanged.

5. Click Save Template and enter the new template name if you want to save your selections for future use.
6. Click Finish.

The rules are created.

ca Aion Rule Manager [Home](#) [Help](#)

Logged in as: **pm.admin** (Log Out)

Rule Manager

Folders

WSUpdateTestRuleSet [Download Later](#) [Duplicate](#) [Delete](#) [Done](#)

[Home](#) > [All Folders](#) > [WSUpdateTestProject](#) > WSUpdateTestRuleSet [View Downloads](#)

Contents **Properties** **Sharing** **Location**

Search Folder: [Go](#) [Advanced](#)

Folder Contents [Create Rule](#) [Create Test Case](#) [Add Content](#)

Select and: [Download Later](#) | [Delete](#) 1 - 1 of 1

| Select | Title | Details | Description | Last Updated | Actions |
|--------------------------|--|---------|-------------|-------------------------|--|
| <input type="checkbox"/> | WSUpdateTestLookupRule | | | Apr 30, 2009 3:55:26 PM | Duplicate Go |

View lookup table rule created from data

Condition Builder **Table View** [Save](#) [Reset](#)

Conditions [Edit Iterations](#) [Configure](#) [Hide Edit Buttons](#)

The Condition Builder allows you to create an IF/THEN statement. This statement defines your Rule which can then be reused in your business applications. To change or reorder the Business Terms, click the Configure button.

| If | | Then |
|--------------|-----|----------------|
| age | sex | salary |
| > 46.808702 | = F | > 23910.427487 |
| > 46.808702 | = M | > 43476.353669 |
| > 39.880249 | = F | > 23910.427487 |
| > 39.880249 | = M | > 43476.353669 |
| <= 39.880249 | = | > 23910.427487 |

Create Test Cases

You can construct test cases, after rules are created, to assess your project.

A test case may consist of any number of scenarios. A scenario is a set of input values and expected results for the business terms used by a project or rule set. Each column in the test case table is one scenario. After the test case is executed, the scenarios are marked successful or failed by comparing the expected results and the actual results. A formatted trace statement is available for each scenario after each execution. It shows how rules are applied to derive at the actual results.

Test a Project

You can test the accuracy of a project by creating test cases under a rule project folder. All rules within this rule project folder are tested when you run your test case.

Test a Rule set

You can test the accuracy of a single rule set by creating the test case within a particular rule set folder. When a test case residing under a rule set folder is executed, only the rules within the rule set are tested.

Create a Project Test Case

To create a Project Test Case

1. Click the Details button of the project.
2. Click Create Test Case.

The Configure Test Case screen displays.

3. Enter a Title for the Test Case.
4. (Optional). Edit the Input or Output Business Terms to fit the characteristics of specific rule sets. To test the project as a whole, you may wish to add any internal terms to the outputs.

By default, the Input Business Terms include any business terms specified as Input terms; Output Business Terms include all terms specified as Output terms.

Note: To add a term, check the term in the available terms list. Click the input button to add it to the input terms list. Click the output button to add it to the output terms list. To delete a term, select the term and click the trash icon.

Note: If you choose a term which is a field of another term of complex type, the top level term(s) will be automatically added for you. For example, if you choose John.ssn as you input, you will see both John, and John.ssn in your input list.


5. (Optional). Enter the Description for the Test Case.
6. Click Next to define the Test Case.

The Test Values and Expected Results screen appears.

7. (Optional). Set up scenarios that you want to include in the test case.

Note: You can change the Scenario name. To add scenarios click Add Column. To delete scenarios click the trash can icon.

8. (Optional). Set up sets.


Add set element by clicking  next to a term of type set. Remove set element by clicking the trash icon next to it. Select Remove Element for any set elements to be excluded for a single scenario.



9. Enter Input values and expected Output results.

Note: Any cell for terms of simple data type left blank is interpreted as an Unknown value.

Note: For a term that is a user defined type, you can explicitly set it to Unknown or NULL by selecting the appropriate check box. If neither unknown nor null box is checked, an instance will be created.

Note: NULL value is not allowed for set terms. Set terms can only be marked Unknown.



Note: For a term of type string, you can set it to an empty value equivalent to "" by clicking the down arrow icon .

Note: You can collapse or expand the branches by clicking on  or  icon. When a branch is collapsed, a summary of the sub-branch contents is displayed.

Note: For a more concise view of the Test Case, click the Compress Table button. The compressed table shows only terms of simple data types. This view is good for data entry. To return to normal view, click the Uncompress button.

10. Click Save to save the test case (if you want to run the test case later).
11. Click Save and Run to save the test case and run it immediately.

Viewing Test Results

Results block display expected values as well as the actual value as calculated by the rules. If both sets of value are equal, the scenario is marked successful by a green check mark , otherwise, the scenario is marked failed by a red cross . The comparison is case insensitive. When comparing sets, order does not matter. Set (2, 1) is considered the same as set (1, 2).

For a more concise view you can collapse and expand branches. When a branch is collapsed you will see a summary of the sub-branch contents. You can also compress or uncompress the table. A compressed table shows only terms of simple data types.

Each scenario column will also have a show trace link. Clicking on this link will take you to the trace page which shows you how the rules are applied to derive the actual results.

The screenshot shows the Aion Rule Manager web application. At the top, it says "Aion Rule Manager" with a logo. There are links for "Home" and "Help". Below that, it says "Logged in as: pm admin (Log Out)". The main section is titled "Rule Manager" and shows a breadcrumb path: "Home > All Folders > WebServiceProject > WSRuleSet > TestWSRule". There are buttons for "Check In", "Download Later", "Delete", "Duplicate", and "Done". A confirmation message says "Confirmation: Save Test case successful". Below that, there are tabs for "Testcase", "Properties", "Version History", "Sharing", and "Location". The "Test Results" tab is selected, showing a table with input and output values for two scenarios. The table has columns for "Input values", "Scenario 1", and "Scenario 2". Under "Input values", there is a row for "age" with values 26 and 25. Under "Output values", there is a row for "num1" with expected values 50 and 70, and actual values 55 and 70. There are "Show Trace" links with red X icons for each scenario.

Test Results

| Input values | Scenario 1 | Scenario 2 |
|--------------|------------|------------|
| age | 26 | 25 |

| Output values | Expected | Actual | Expected | Actual |
|---------------|----------|--------|----------|--------|
| num1 | 50 | 55 | 70 | 70 |

✖ Show Trace ✖ Show Trace

Understanding the Trace

The left side of the trace page shows the Test Values for the scenario and the right side of the trace page shows the detail trace messages. Each trace message has an order number, Rule Name, Iteration, Status, and Summary.

Order

The order of the rules being applied.

Rule

The name of the rule evaluated. You can click on the rule name to see the rule definition.

Iteration

An iteration number shows within which iteration in the loop this rule is evaluated. Since loops can be nested, the iteration number can also be nested. For example 2.3 mean it's the second iteration in the outer loop and third iteration in the inner loop.

Status



Condition not satisfied, no action taken.



Condition satisfied, actions taken.



Insufficient data to apply, no action taken.

Summary

The summary column lists values of all terms read and set by the engine.

| Test Values | | Rules Tracing | | | | |
|-------------------------------|---------|---|-----------------------|-----------|--------|---|
| Scenario 2 | | Show the rules used to generate the results and the order the rules were applied. | | | | |
| shoppingDate 3/07/ 3:02:00 PM | | Order | Rule | Iteration | Status | Summary |
| theCart | | 0 | Cart | 1 | ✗ | theCart = ShoppingCart.theCart. No Action taken. |
| theCart.theCustomer | | 1 | ShoppingDate | 1 | ✗ | shoppingDate = DTM:2007-03-07T15:02:00. No Action taken. |
| theCustomer.id | 600000 | 2 | Id | 1 | ✗ | theCart = ShoppingCart.theCart and ShoppingCart.theCart.theCustomer = Customer.theCustomer and Customer.theCustomer.Id = 600000. No Action taken. |
| theCustomer.region | REGION1 | 3 | Nonnull | 1 | ✗ | theCart = ShoppingCart.theCart and ShoppingCart.theCart.theCustomer = Customer.theCustomer. No Action taken. |
| vouchers.Voucher1 | | 4 | Region | 1 | ✗ | theCart = ShoppingCart.theCart and ShoppingCart.theCart.theCustomer = Customer.theCustomer and Customer.theCustomer.Region = CustomerRegion.REGION1. No Action taken. |
| Voucher1.expirationDate | | 5 | Description | | ✗ | theCart = ShoppingCart.theCart and ShoppingCart.theCart.Items = SET(). No Action taken. |
| Voucher1.itemCategory | | 6 | Price | | ✗ | theCart = ShoppingCart.theCart and ShoppingCart.theCart.Items = SET(). No Action taken. |
| Voucher1.isUsed | | 7 | ExpirationDate | | ✗ | theCart = ShoppingCart.theCart and ShoppingCart.theCart.Items = SET(). No Action taken. |
| vouchers.Voucher2 | | 8 | IsUsed | | ✗ | theCart = ShoppingCart.theCart and ShoppingCart.theCart.theCustomer = Customer.theCustomer and Customer.theCustomer.Vouchers = SET(). No Action taken. |
| Voucher2.value | | 9 | ItemCategory | | ✗ | theCart = ShoppingCart.theCart and ShoppingCart.theCart.theCustomer = Customer.theCustomer and Customer.theCustomer.Vouchers = SET(). No Action taken. |
| Voucher2.expirationDate | | 10 | Value | | ✗ | theCart = ShoppingCart.theCart and ShoppingCart.theCart.theCustomer = Customer.theCustomer and Customer.theCustomer.Vouchers = SET(). No Action taken. |
| Voucher2.isUsed | | 11 | IsIWork | 1 | ✓ | theIWork = IWork.IWork(1). |
| vouchers.Voucher3 | | 12 | CalcItemCategoryNames | 1 | ✓ | theCart = ShoppingCart.theCart and ShoppingCart.theCart.Items = SET() and theIWork = IWork.IWork(1) then IWork.IWork(1).CategoryNames = SET(). |
| Voucher3.value | | | | | | |
| Voucher3.expirationDate | | | | | | |
| Voucher3.itemCategory | | | | | | |
| Voucher3.isUsed | | | | | | |
| vouchers.Voucher4 | | | | | | |
| Voucher4.value | | | | | | |
| Voucher4.expirationDate | | | | | | |
| Voucher4.itemCategory | | | | | | |
| Voucher4.isUsed | | | | | | |
| theCart.items | | | | | | |
| items.Item1 | | | | | | |
| Item1.price | | | | | | |
| Item1.desc | | | | | | |
| items.Item2 | | | | | | |
| Item2.price | | | | | | |
| Item2.desc | | | | | | |
| items.Item3 | | | | | | |
| Item3.price | | | | | | |
| Item3.desc | | | | | | |

How to Modify a Test Case

Test Cases can be modified by editing them. You have to check-out the Test Case before modifying it.

Modify a Project Test Case

To modify a Project Test Case

1. Click the Test Case Details button.

Test Case page appears.

2. Modify scenario titles. Add scenarios by clicking on the Add Column button. Delete scenarios by clicking the trash can icon.
3. Modify the test values for the scenarios.

Note: To add set element click next to a term of type set. To remove set element by click the trash icon next to it. Select Remove Element for any set elements to be excluded for a single scenario.

Note: Any cell for terms of simple data type left blank is interpreted as an Unknown value.

Note: For a term that is a user defined type, you can explicitly set it to Unknown or NULL by selecting the appropriate check box. If neither Unknown nor Null box is checked, an instance will be created.

Note: NULL value is not allowed for set terms. Set terms can only be marked Unknown.

Note: For a term of type string, you can set it to an empty value equivalent to "" by clicking the down arrow icon.

Note: You can collapse or expand the branches by clicking on or icon. When a branch is collapsed, a summary of the sub-branch contents is displayed.

Note: For a more concise view of the Test Case, click the Compress Table button. The compressed table shows only terms of simple data types. This view is good for data entry. To return to normal view, click the Uncompress button.

4. Click Configure to or remove input or output business terms.

Note: This will take you to the term selection page. You can add and remove input and output terms. If the term is removed from input and output list, all data assigned to that term for all scenarios will be removed.

5. Click Save to save the test case (if you want to run the test case later).
6. Click Save and Run to save the test case and run it immediately.

Validate a Project

You can validate a project to ensure that it is created correctly.

To validate a project

1. Click the Details button, in the Rule Manager portlet, of the project that contains the new rule set.
2. Click Validate.

A confirmation message appears if the project is valid. If the project is valid but with errors and warnings, a page with the warning and error message appears.

Work with Rule Sets

Once a rule set has been created using the Rule Manager, you can update a rule set's location, properties, versions, and so on.

Convert Rule Manager Project to BRE

The Rule Manager to BRE conversion utility can be used by customers to integrate projects authored in Rule Manager with existing BRE applications or by customers who want to take advantage of the faster speed of inference of BRE.

A project created using the Rule Manager can be converted to a BRE rule library in the form of a BRE application file using the Convert to BRE option in Rule Manager.

To convert Rule Manager Project to BRE

1. Log in to CA Aion Rule Manager.
2. Click Convert to BRE option in the Project folder.

A confirmation message displays above the definition portlet. An error message displays if the conversion fails.

The converted BRE code is a BRE library containing the rules of the rulebase. This library uses a specific structure to implement the converted rules. For more information about description of this structure and procedure to invoke inference using this library, see [Configuring CA Aion Rule Manager for the TLP Insurance Tour](#) (see page 113).

The converted file is created under the same project (Rule Manager Project folder) as a separate file in the following format:

```
<<Projectname>>_BRE.app
```

Note: Conversion to BRE is possible only at project level. When the conversion is successful, the BRE code is checked into Library and appears as a separate entry in the Project folder. If an older version exists already, that version needs to be checked out for update first. The update process is automatic if the previously generated application has not been checked out by another user. Rules created with a prior version of the product may need to be checked out and saved again into the current format to help in proper conversion.

Duplicate a Rule Set

Duplicating a rule set puts a copy of that rule set with a new name in the same parent project. For example, if you duplicate Rule SetB, which is a subfolder of ProjectA, then Copy of Rule Set B is created in ProjectA.

There are two ways to duplicate a rule set.

Use the Details Button for the Project

You can duplicate a rule set in order to make changes without losing the original rule set.

To duplicate a rule set using the Details button for the project

1. Click the Details button, in the Rule Manager portlet, for the project that contains the rule set to duplicate.
2. Find the rule set you want to duplicate.
3. Select Duplicate, from the Actions drop-down menu, and click Go.



A confirmation message indicates that the rule set was duplicated successfully. The new copy of the rule set is located in the Rule Set list.

Use the Details Button for the Rule Set

To duplicate a rule set using the Details button for the rule set

1. Click the project, in the Rule Manager portlet, that contains the rule set to duplicate.
2. Click the Details button for the rule set.
3. Click Duplicate.

A confirmation message indicates that the rule was duplicated successfully. The new copy of the rule set is located in the Rule Set list.

Move a Rule Set's Location

The Location tab of a rule set's Detail page identifies where a rule set is currently stored, and lets you move a rule set from one location to another.

To move a rule set's location

1. Click the Details button, in the Rule Manager portlet, for the project that contains the rule set you want to move.
2. Click the rule set that you want to move, and click the Location tab.
3. Click a project name to view its rule sets. The project directory displays all the available projects where you can move the rule set to.

Note: If you do not have Write permissions for a project, you cannot move any rule set to that project.

4. Click Save to save the rule set to the new location.

A confirmation message confirms that the rule set was moved successfully.

5. Click Reset before clicking Save to return the folder tree to its original structure.

Edit a Rule Set's Properties

The Properties tab of the rule set's Details page lists all of the features and elements that make up a rule set.

To edit a rule set's properties

1. Click the project, in the Rule Manager portlet, that contains the rule set to edit.
2. Click the Details button for the rule set.

3. Update the required fields.
If you wish to return the fields to their original settings, click Reset.
4. Click Save.
A confirmation message indicates that changes were successfully applied.

Work with Rules

Once a rule has been created using the Rule Manager, you can update a rule's location, properties, versions, and so on.

Check In and Check Out a Rule

You can make any changes to a rule or its properties by checking-out the rule. If someone else has the rule checked-out, you cannot make any change to it. By checking-out a rule, you ensure that only one user at a time can update the rule or its properties. If a checked-out rule is opened by another user, a read-only view of the last checked-in version is obtained.

A new version of the rule is created once it is checked-in. A detailed record of the rule's version history can be found on the Version History tab (for more information, see Version History later in this chapter).

Check Out a Rule

The Rule Manager checks-out a rule by default when it is created. You can use the the Details Button to check out a rule.

To Check Out a rule using the Details button for the rule

1. Click the project, in the Rule Manager portlet, that contains the rule to check out.
2. Click the rule set that contains the rule to check out.
3. Click the Details button for the rule.
4. Click Check Out.

A confirmation message indicates that the rule was checked out successfully.

Check In a Rule

You can use the following ways to check in a rule that has been subsequently checked out:

- Use the Details Button for the Rule
- Use the Details Button for the Rule Set

Note: You can check in only the test cases and the other documents from rule set page. To check in the rules, you must use the rules page.

To Check In a rule using the Details button for the rule

1. In the Rule Manager portlet, click the project that contains the rule to check in.
2. Click the rule set that contains the rule to check in.
3. Click the Details button for the rule.
4. Click Check In

A confirmation message indicates that the rule was checked in successfully.

To Check In a rule using the Details button for the rule set

1. In the Rule Manager portlet, click the project that contains the rule to check in.
2. Click the Details button for the rule set that contains the rule to check in.
3. From the Actions drop-down menu, select Check In and click Go.

A confirmation message indicates that the rule was checked in successfully.

Note: The procedure to check-in a test case is similar to that given for a rule.

Duplicate a Rule

Duplicating a rule puts that rule in the same rule set, but with a new name. For example, if you duplicate RuleB in Rule Set A, then Copy of RuleB is created in Rule Set A.

Note: The user that performs the duplication has full permission (Read, Write, Delete, and Change Permission) on the new rule. All of the other permissions from the rule that was duplicated apply to the newly-copied rule as well.

There are two ways to duplicate a rule.

Use the Details Button for the Rule

Duplicate a rule in order to make changes without losing the original rule.

To duplicate a rule using the Details button for the rule

1. Click the project, in the Rule Manager portlet, that contains the rule to duplicate.
2. Click the rule set that contains the rule to duplicate.
3. Click the Details button for the rule.
4. Click Duplicate.

A confirmation message indicates that the rule was duplicated successfully. The duplicated rule can be found on the rule set's Contents tab, where it is named "Copy Of x" where x=rule name.

Note: The procedure to duplicate a test case is similar to that given for a rule.

Use the Details Button for the Rule Set

To duplicate a rule using the Details button for the rule set

1. Click the project, in the Rule Manager portlet, that contains the rule to duplicate.
2. Click the Details button for the rule set that contains the rule to duplicate.
3. Select Duplicate, from the Actions drop-down menu, and click Go.



A confirmation message indicates that the rule was duplicated successfully. The duplicated rule can be found on the rule set's Contents tab, where it is named "Copy Of x" where x=rule name.

Note: The procedure to duplicate a test case is similar to that given for a rule.

Delete a Rule

There are two ways to delete a rule.

Use the Details Button for the Rule

You can delete a rule when you no longer need it.

To delete a rule using the Details button for the rule

1. Click the project, in the Rule Manager portlet, that contains the rule to delete.
2. Click the rule set that contains the rule to delete.
3. Click the Details button for the rule.
4. Click Delete.

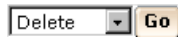
A confirmation message indicates that the rule was deleted successfully.

Note: The procedure to delete a test case is similar to that given for a rule.

Use the Details Button for the Rule Set

To delete a rule using the Details button for the rule set

1. Click the project, in the Rule Manager portlet, that contains the rule to delete.
2. Click the Details button for the rule set that contains the rule to delete.
3. Select Delete, from the Actions drop-down menu, and click Go.



A confirmation message indicates that the rule was deleted successfully.

Note: The procedure to delete a test case is similar to that given for a rule.

Move a Rule's Location

The Location tab of a rule's Detail page identifies where a rule is currently stored, and lets you move a rule from one location to another.

Rule can be moved from one Rule Set to another Rule Set of same project or into some other project. Moving the Rule from one Rule Set to another Rule Set within the same project will not have any impact on Rule Manager project. But moving the Rule from a Rule Set of a Rule Manager Project to another Rule Set of another Rule Manager Project is similar to deletion of a Rule from source Project.

There are two ways to move a rule.

Use the Details Button for the Rule

If you want to use a specific rule in a different rule set or project, you can move the rule's location.

To move a rule using the Details button for the rule

1. Click the project, in the Rule Manager portlet, that contains the rule that you want to move.
2. Click the rule set that contains the rule that you want to move.
3. Click the Details button for the rule.
4. Click the Location tab.

The rule set's current directory displays. Use the breadcrumb links to navigate to the rule set that you want to move the rule to.

Note: If you do not have Write permissions for a rule set, you cannot move any rule to that rule set.

5. Click Save to save the rule to the new location.

A confirmation message confirms that the rule was moved successfully.

6. Click Reset to return the folder tree to its original structure.

Note: Clicking Reset does not undo a move once it has been saved.

Note: The procedure to move a test case is similar to that given for a rule.

Use the Details Button for the Rule Set

To move a rule using the Details button for the rule set

1. In the Rule Manager portlet, click the project that contains the rule that you want to move.
2. Click the Details button for the rule set that contains the rule that you want to move.
3. From the Actions drop-down menu, select Move and click Go.



Continue from Step 4 in Use the Details Button for the Rule.

Note: The procedure to move a test case is similar to that given for a rule.

Edit a Rule's Properties

The Properties tab of the rule's Details page lists all of the features and elements that make up a rule.

To edit a rule's properties

1. In the Rule Manager portlet, click the project that contains the rule to edit.
2. Click the rule set that contains the rule to edit.
3. Click the Details button for the rule.
4. Click the Properties tab.
5. Click Check Out if the rule is currently not checked out.
6. Update the fields as necessary.
7. Click Reset to return the folder tree to its original structure.
8. Click Save.

A confirmation message indicates that the changes were successfully applied.

Note: The procedure to edit a test case's properties is similar to that given for a rule.

Version History

You can maintain multiple versions of a rule, including a record of the changes to the rule from version to version, through the Version History tab on the rule's Details page.

All rules have a detailed record of their history on the Version History tab. You can view version information, comments, properties, and other details about the rule's history, as well as revert to a previous version of the rule, or delete a previous version.

The current Version Number displays the current version of the rule.

Note: The procedure to Create, Revert and to Delete Versions for a test case is similar to that given for a rule.

Create Rule Versions

You can alter an existing rule without deleting its original version.

To create a new version of a particular rule

1. Click the project, in the Rule Manager portlet, that contains the rule that you want to create a new version for.
2. Click the rule set that contains the rule.
3. Click the Details button for the rule.
4. Click Check In, if the rule is not checked in already.
5. Update the rule (on the Contents tab or Properties tab), including comments about the new version in the Properties tab's Version Comments text box.
6. Click Check Out at the top of the Details page.

A message indicating that the rule was checked-out successfully displays under the breadcrumb. The editing buttons are enabled.

A new version of the rule is created when it is checked out. The previous version can then be found on the Version History tab.

The screenshot shows the Aion Rule Manager interface. At the top, it says 'Logged in as: pm admin (Log Out)'. Below that is the 'Rule Manager' portlet. The breadcrumb path is 'Home > All Folders > ShoppingCartRuleProject > ShoppingCartRuleSet > discountLookupRuleUpdated'. There are buttons for 'Check In', 'Download Later', 'Delete', 'Duplicate', and 'Done'. A confirmation message states: 'Confirmation: Content has been checked-out successfully'. Below this is a tabbed interface with 'Rule', 'Properties', 'Version History', 'Sharing', and 'Location'. The 'Version History' tab is active, showing a table of rule versions.

| Version Number | Title | Details | Created By | Date Created | Version Comments | Delete Version |
|----------------|---------------------------|---------|------------|--------------------------|---------------------------|----------------|
| 145 | discountLookupRuleUpdated | | pm admin | May 11, 2009 6:12:20 PM | check-in for change no 2. | |
| 144 | discountLookupRuleUpdated | | pm admin | Apr 28, 2009 11:22:22 AM | gyy | |
| 143 | discountLookupRuleUpdated | | pm admin | Apr 28, 2009 11:19:50 AM | tgt | |
| 142 | discountLookupRuleUpdated | | pm admin | Apr 24, 2009 6:27:52 PM | hjhj | |
| 141 | discountLookupRuleUpdated | | pm admin | Apr 24, 2009 6:27:11 PM | gyjyj | |
| 140 | discountLookupRuleUpdated | | pm admin | Apr 24, 2009 6:26:54 PM | yujyu | |

Note: When a rule is initially created and saved, the Rule Manager lists it as Version Number 1. Be sure to update the Version Notes on any subsequent rule versions to avoid confusion.

Revert to a Previous Rule Version

You can revert a rule to a previous version using the Details button for the rule.

To revert to a previous rule version

1. Click the project, in the Rule Manager portlet, that contains the rule that you want to revert to a previous version.
2. Click the rule set that contains the rule to revert.

3. Click the Details button for the rule.
4. Click Check Out if the rule is currently not checked out.
5. Click the Version History tab.
6. Click the button next to the version that you want to revert to.
7. Click Select and: Revert to Version.

A confirmation message indicates that the rule was reverted successfully.

A new rule version is created and the selection becomes the current version. Since a new version is created, the current version before the revert becomes the latest version and is put in the Version List on the Version History tab.

Delete a Rule Version

If a particular version of a rule is unhelpful to you, you can delete it without deleting the entire rule.

To delete a rule version

1. Click the project, in the Rule Manager portlet, that contains the rule that you want to delete.
2. Click the rule set that contains the rule to delete.
3. Click the Details button for the rule.
4. Click the Version History tab.
5. Click the trash icon to delete the selected version.

The version is removed from the Version List.

Note: If there is only one version in the Version List and the rule is checked in, then this version cannot be deleted.

Specify Access and Permission

The Sharing tab of a rule or rule set's Details page specifies which users and groups have access to a specific rule or rule set, and set their permissions. This is an administrative function.

Note: You must have Change Other's Permissions access to do this.

To specify the permissions for a rule or rule set

1. Click the rule set or rule's Details button to open the Details page.
2. Click the Sharing tab.
3. Click Save once all settings have been applied.
4. Click Reset to return to the previous settings.

You can edit the level of access of those groups/users listed in the Selected for Sharing pane, or add groups/users to this list to provide access. You can set permissions for a specific rule set or for a rule set and all of its rules. You can view individual members in a group by clicking the Group name.

Note: The procedure to Specify Access and Permission for a test case is similar to that given for a rule.

Available Users

This section lets you locate all available users that can gain access to the rule or rule set.


Search for a User

The Search feature of the Sharing tab lets you find specific users to provide access to the rule or rule set.

To search for a user you want to provide access


1. Select the Type
2. Select the Operator.
3. Type the Value and click Search

The search results displayed below the Search button.

4. Select the Users and click  to move the names to the Sharing Policy portlet.
5. Click the required check-box to select the Permissions
6. Click Save

All Groups and Users

ALL GROUPS AND USERS is a special entry. It provides permissions for anyone not explicitly listed in the list of users and groups selected for sharing for this rule. Users belonging to the Admin group always have full access regardless of the access level assigned for this rule set or rule.

Specific groups and users are listed beneath ALL GROUPS AND USERS. To browse among the groups, click the arrow  to access individual users.



Use the breadcrumb links to return to a previous level.

Selected for Sharing

The Selected for Sharing window displays a list of all groups/users that have some level of access assigned for the current rule or rule set.

You can do the following:

- Add users and groups to the list of users and groups selected for sharing for the rule set or rule:
 - Click the check box next to the users and/or groups you want to select for sharing in the Available Users window.

Or, enter the user or group name that you want to select for sharing in the Search field, and click Search.
 - Once the users and/or groups has been selected for sharing, click Select  to move them to the Selected for Sharing window.
- Remove users and groups from the list of users and groups selected for sharing for the rule set or rule:
 - In the Selected for Sharing window, highlight the users and/or groups that you want to remove sharing privileges for.
 - Click Remove  to remove the selected users/groups from the Selected for Sharing window.
- Edit the access level of those users and groups already listed in the list of users and groups selected for sharing for the rule set or rule:
 - In the Selected for Sharing window, highlight the users and/or groups that you want to edit the sharing privileges for.
 - In the Permissions window, click the check box next to the permission settings you want the users and/or groups to possess.

Permissions

Set the permissions for the selected users by clicking one or more of the following:

Read

Provides only Read permission for a user/group.

Write

Provides both Read and Write permission for a user/group.

Deploy

Provides the permission to deploy the rule as webservice to users other than administrators. This Permission is applicable for Projects type template only.

Delete

Provides Delete permission for a user/group for deleting rules/rule sets.

Change Other's Permissions

Allows you to access the Sharing tab to update permissions for the rule or rule set. Without this permission, you cannot view the Sharing tab, change permissions, give permissions to a new user, remove a user's permissions, and so on.

Note: Users belonging to the Admin group always have full access regardless of the access level assigned for this rule set or rule.

Apply Settings To Rule Set

If you are applying settings for a rule set, select one of the following settings:

- This Rule Set only-Applies the settings you have selected to the current rule set
- This Rule Set and all of its contents-Applies the settings you have selected to the current rule set and everything it contains

Chapter 5: Uninstalling CA Aion Rule Manager

This section contains the following topics:

[Uninstallation Considerations](#) (see page 109)

[How to Uninstall CA Aion Rule Manager](#) (see page 110)

Uninstallation Considerations

Consider the following information before you uninstall CA Aion Rule Manager:

- The complete removal of CA Aion Rule Manager is the only supported option. Uninstalling and subsequent reinstalling the individual product components with continued use of CA Aion Rule Manager is supported.
- Uninstalling CA Aion Rule Manager does not remove Library contents. You can find the Library directory at a location previously specified in the Library Location field on the Content Management Information page during the installation.

Note: To reuse Library content after re-installation, you need to use same CA Aion Rule Manager database with same Instance Id that was used before uninstallation. Also, Library directory path must not be changed. Also, an Instance Id can be reused on only system (system where earlier Rule manager installation with that Instance Id was done) where it was created.

- The uninstall process will remove all files that the install process created including those customized after the installation. The uninstall process will also undeploy CA Aion Rule Manager webservices and delete configuration files that contain deployment information. Before uninstalling, be sure to take backup of all customizations done after installation.
- You must uninstall both CA Aion Rule Manager and the associated external components, like EEM, using the Control Panel (Add or Remove Programs). Uninstalling CA Aion Rule Manager database is an exception. The CA Aion Rule Manager database must be manually [uninstalled](#) (see page 111).
- EEM is *common component* that may be used by other CA products outside of CA Aion Rule Manager. Therefore, use extreme caution when uninstalling EEM. By removing a common component (EEM) installed by CA Aion Rule Manager, a different CA product may not work correctly. Uninstalling CA Aion Rule Manager does not remove EEM.

How to Uninstall CA Aion Rule Manager

If you need to uninstall CA Aion Rule Manager, uninstall the components in the following order.

1. If you have not already done so, stop JBoss.
2. Uninstall the following components in this order using the Control Panel (Add or Remove Programs):

- a. CA Aion Rule Manager

If you are prompted to restart your computer after uninstallation, do so before proceeding to the next step.

Note: If CA Aion Rule Manager is installed in silent mode, uninstallation happens in silent mode, that is, no GUI is shown while uninstalling.

- b. CA Embedded Entitlements Manager (EEM)

Before you uninstall EEM, unregister the CA Aion Rule Manager instance of EEM. For more information, see [Unregister CA Aion Rule Manager Instance of EEM](#) (see page 110).

3. Uninstall CA Aion Rule Manager database.

For more information, see [Uninstall CA Aion Rule Manager Database](#) (see page 111).

Unregister CA Aion Rule Manager Instance of EEM

Before you uninstall EEM, you should unregister the CA Aion Rule Manager instance of EEM.

To unregister the CA Aion Rule Manager instance of EEM

1. Use a browser and navigate to <https://localhost:5250/spin/eiam>.
2. Log in to EEM using the Global application instance, EiamAdmin user, and the password that you provided during the installation.
3. Click the Configure tab.
4. In the Applications section, click the Rule Manager installation instance. It may be RuleManager (default value) or the one specified in Application Instance field on the EEM Information page during installation.
5. On the right side of the page, click Unregister.
6. Click OK and wait until a confirmation message appears.
7. Click Refresh in Applications section to confirm that the Rule Manager application instance is no longer listed.
8. Log out of EEM.

Uninstall CA Aion Rule Manager Database

To complete the uninstallation of CA Aion Rule Manager, you should uninstall CA Aion Rule Manager Database.

To uninstall CA Aion Rule Manager Database

1. Open SQL Server Management Studio

For Microsoft SQL Server 2005, this option is available from Start, Programs, Microsoft SQL Server 2005 menu. For Microsoft SQL Server 2008, this option is available from Start, Programs, Microsoft SQL Server 2008 menu.

2. Log in using SQL Server Authentication with user having administrative privileges.
3. On the left side of the page, expand Databases, right-click CA Aion Rule Manager Database, and click Delete.

Note: Database name may be CA Aion Rule Manager Database (default) or the one specified in Database name field on the Aion Rule Manager Database Information (1 of 2) page during installation.

4. Click OK to confirm the deletion.
5. Expand Security, Logins.
6. Right-click the CA Aion Rule Manager Database login and click Delete.

Note: Login may be rmdba (default) or the one specified in Database username field on the Aion Rule Manager Database Information (1 of 2) page during installation.

7. Click OK to confirm the deletion.
8. Exit SQL Server Management Studio.

Appendix A: Configuring CA Aion Rule Manager for the TLP Insurance Tour

This section uses a sample application known as the TLP Insurance tour as an aid to walk you through the various options of the CA Aion Rule Manager.

Note: TLP Insurance is a fictitious company created specifically for the purpose of a sample application.

For more information about the topics discussed in this chapter, see the *CA Aion Rule Manager Help*.

This section contains the following topics:

[Introduction to the TLP Insurance Tour](#) (see page 113)

[Create Pricing Tier Rules](#) (see page 116)

[User and Group Accounts](#) (see page 128)

Introduction to the TLP Insurance Tour

TLP Insurance is a property insurance business that offers a variety of coverage options. They are a small but growing company, and currently are finding that their existing business processes cannot handle their planned growth. They have decided that they can improve their service levels by identifying and automating key business processes. In this tour you create a business process model that automates the steps that different people and systems take to process new policy applications.

Roles

In this tour you play a number of roles. The first table shows the roles played when you create the insurance business process:

| | Role | User ID | Activity |
|--|---------------|----------|---|
| Create the Insurance Business Process | Administrator | rm admin | Create some user accounts for the staff at TLP Insurance (Alex the Agent, Phyllis the Property Appraiser and Clark the credit manager). |
| | Rule Author | rm admin | Use Rule Manager to create a rule that determines the insurance pricing tier based |

| Role | User ID | Activity |
|--------------------|----------|--|
| Rule Administrator | rm admin | on the customer financial stability (as determined by credit managers) and the property risk (as determined by property appraisers). |
| | | Use the Rule Manager to deploy the rules as a web service. |

Note: The rm admin ID is the one you created during the CA Aion Rule Manager installation process.

The second table shows the roles played when you run the insurance business process:

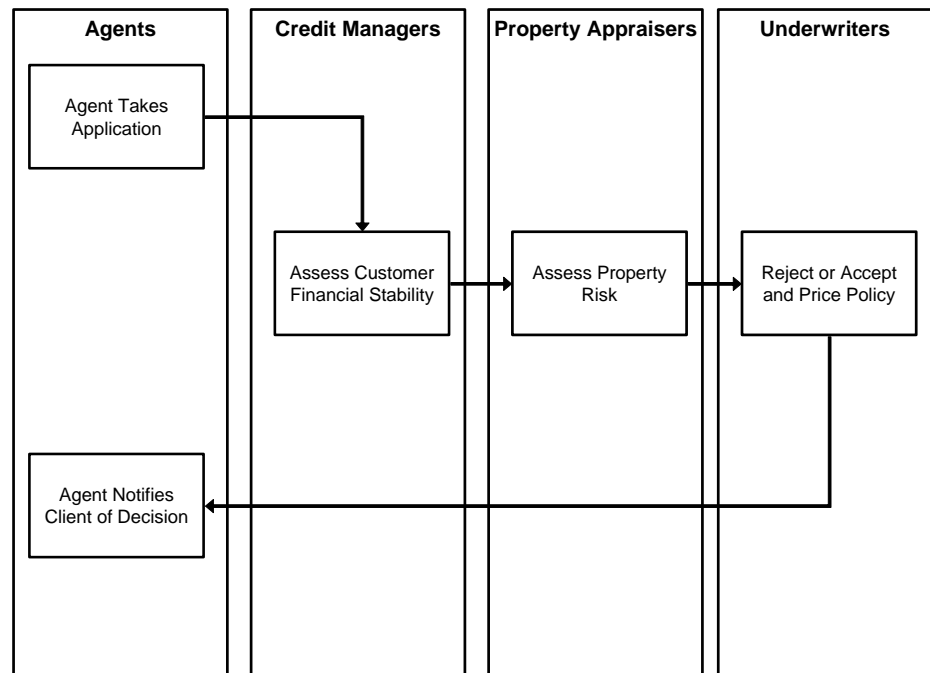
| | Role | User ID | Password | Activity |
|---|--------------------|---------|----------|---|
| Run the Insurance Business Process | Insurance Agent | Alex | Alex | Get the property details from the customer and enter them in the Insurance quote screen; Also received back the final decision to communicate to the customer |
| | Property Appraiser | Phyllis | Phyllis | Review the property and assess its insurance risk (high, medium or low) |
| | Credit Manager | Clark | Clark | Review the customer and assess their financial stability (excellent, average, poor) |

TLP Insurance-Summary of Activities

Now, let us take a closer look at the activities performed by each role in the TLP insurance business process. To do this, you play three different roles previously described by logging into four different parts of the CA Aion Rule Manager tool.

The TLP acceptance and pricing process is well defined by the company, but they want to change it slightly to take advantage of the power of business rules. In the following graphic, when an agent files an application for a new policy, TLP has a credit manager review the customer's financial information. A property appraiser then reviews the building being insured. In the past, every policy was then sent to an underwriter who decided whether or not to accept the application and assign a pricing tier to it. TLP earlier performed some analysis with the underwriters, and found that a number of their decisions could be expressed using simple rules. TLP wants to automate these simple rules so that only the most difficult decisions are routed to underwriters.

The following illustration shows the process:



Later in the tour, you run the insurance business process by playing the three staff roles at TLP Insurance and you interact using a web browser.

The following sections provide detailed, step-by-step instructions to set up the environment and define user access as the first step in creating the TLP insurance business process.

Log In to CA EEM

CA Aion Rule Manager uses EEM to define and control access to the services and interfaces of CA Aion Rule Manager. In this introductory section, you log in to CA Aion Rule Manager as Administrator and set up the environment for the rest of the product tours. You also add users to EEM and configure it for those users.

To log in to CA Aion Rule Manager Administration

1. Enter URL `http://rm_host:rm_port/rmportal`

where,

rm_host

Defines machine name of the server hosting rule manager application server.

rm_port

Defines the port number of rule manager.

Default value: 8080.

2. Log in with the rights of portal library administrator.
3. From All Folders page, click detail icon.
User can see four tabs, with last tab as CPCM Administrator.
4. Click on tab CPCM Administrator.
Now user can do administrative changes.

Create Pricing Tier Rules

In the TLP Insurance example, you create the pricing tier rules of TLP Insurance using the Rule Manager.

To create Pricing Tier rules using Rule Manager

- Create your rule project.
For more information, see [Create a Project](#) (see page 46).
- Create business terms, using the Business Vocabulary Manager, that model the business concepts necessary for rule creation.
For more information, see [Business Vocabulary Manager](#) (see page 47), [Create a Vocabulary](#) (see page 48), and [Create Business Terms](#) (see page 48).
- Create rule set.
For more information, see [Create a Rule Set](#) (see page 53).
- Create a rule.
For more information, see [Create Rule](#) (see page 55), [Select a Rule Type](#) (see page 57), and [Edit Rule for TPL Insurance](#) (see page 117).

- Use the testing module to test the rule.
For more information, see [Test Rules for TPL Insurance](#) (see page 121).
- Deploy the rule so it can be used as a step in a business process.
For more information, see [Deploy Rules for TPL Insurance](#) (see page 124).

In our scenario, the business has defined the following rules for pricing insurance, which needs to be automated in Rule Manager:

| Customer Financial Stability | Property Risk | Set Pricing Tier.... |
|------------------------------|---------------|----------------------|
| Excellent | High | C |
| | Med | B |
| | Low | A |
| Average | High | Refer |
| | Med | C |
| | Low | B |
| Poor | High | Reject |
| | Med | Reject |
| | Low | Reject |

These rules determine the pricing tier based on the Customer Financial Stability and the Property Risk.

Edit Rule for TPL Insurance

The previous steps created a rule in the ruleset with the select business terms. Now you edit the rule to add the conditions and actions for the rule.

The Lookup Table rule editor lets you enter the rule in the form of a table. Each row in the table specifies a list of conditions on the If business terms. If these terms are satisfied, it causes the rule to take the action specified in the then columns of the row. In a Lookup Table rule, once a row has been satisfied, no further rows are processed.

You can now enter the contents of the rule to match the business rule defined at the beginning of this chapter.

To edit the rule

1. Enter the following conditions into the table:
 - Customer Financial Stability as Excellent
 - Property Risk as High
 - Pricing Tier as C

The screen looks like the following example.

The screenshot shows the 'Rule Manager' interface for editing a 'Pricing Tier Evaluation Rule'. The 'Table View' tab is active, displaying a table with the following structure:

| Customer Financial Stabil... | Property Risk | Pricing Tier |
|------------------------------|---------------|--------------|
| Excellent | High | C |

Below the table, there is a '+' button in the 'Property Risk' column, indicating that the row can be split to create multiple rows. The interface also includes buttons for 'Save', 'Reset', 'Check In', 'Download Later', 'Delete', 'Duplicate', and 'Done'.

2. Split the row by clicking the + button in the Property Risk column.

The Property Risk and Pricing Tier cells duplicate. The Customer Financial Stability cell centers between the two rows of the table, signifying that it applies to both rows.

In the rule editor, you can copy or split rows to extend the table. Rows that are split to the right of the first column apply all the business term conditions to the left when processing the row. This allows you to reproduce business rules specified as tables like the one at the beginning of this chapter.

3. Enter Medium for the Property Risk and B for the Pricing Tier of this row.

Note: Once you have multiple rows in a table, the Trashcan icon appears, allowing you to delete rows of the table.

- Split the row again under Property Risk and enter Low for Low Property Risk, and A for Pricing Tier.

Rule Manager

Folders

Pricing Tier Evaluation Rule Check In Download Later Delete Duplicate Done

Home > All Folders > TLP Insurance > Pricing Tier Rules > Pricing Tier Evaluation Rule View Downloads

Confirmation: Changes have been applied.

Rule **Properties** **Version History** **Sharing** **Location**

Condition Builder **Table View** Save Reset

Conditions Edit Iterations Configure Hide Edit Buttons

The Condition Builder allows you to create an IF/THEN statement. This statement defines your Rule which can then be reused in your business applications. To change or reorder the Business Terms, click the Configure button.

| Customer Financial Stabil... | If | Then |
|------------------------------|---------------|--------------|
| | Property Risk | Pricing Tier |
| = | High | C |
| + | Medium | B |
| = Excellent | Low | A |
| + | | |

Check In Download Later Delete Duplicate Done

- Split Customer Financial Stability by clicking the + button under the Customer Financial Stability column. This duplicates the entire set of cells added so far.

- Change the second Customer Financial Stability condition to Average, and change the pricing tiers to Refer, C and B respectively.

Note: You can use the Tab key to move the cursor from cell to cell in the table.

Confirmation: Changes have been applied.

Rule Properties Version History Sharing Location

Condition Builder Table View

Save Reset

Conditions Edit Iterations Configure Hide Edit Buttons

The Condition Builder allows you to create an IF/THEN statement. This statement defines your Rule which can then be reused in your business applications. To change or reorder the Business Terms, click the Configure button.

| If | | Then |
|------------------------------|---------------|--------------|
| Customer Financial Stabil... | Property Risk | Pricing Tier |
| = Excellent | High | C |
| | Medium | B |
| | Low | A |
| | High | Refer |
| | Medium | C |
| | Low | B |

- Split Customer Financial Stability again by clicking the last + button in the Customer Financial Stability column. Enter Poor for the new row of Customer Financial Stability.
- Click the last Trashcan icon under the Property Risk column twice to delete the last two rows of the table, and clear the text in the remaining Property Risk field.

Because the pricing tier is the same (Reject) regardless of the property risk, you do not need to test the Property Risk business term.

Note: You can specify that a rule applies to any value of a business term by leaving its cell blank.

- Set the Pricing Tier to Reject.
- Click Save.
- Select the TLP Insurance link from the trail of links at the top of the page to return to the project folder.

More information:

[Edit a Rule](#) (see page 79)

[Edit a Lookup Table Rule](#) (see page 79)

[Edit a Definition Rule](#) (see page 80)

[Edit a Mandatory Constraint Rule](#) (see page 81)

[Edit a Default Rule](#) (see page 82)

Test Rules for TPL Insurance

After creating rules, you can test them to make sure they have the desired behavior.

To test the rules

1. Click Create Test Case in the TPL Insurance folder.

The Create Test Case wizard opens.

2. Enter a Test Case 1 for Title and click Next to accept the defaults.

Note: By default, the Input Business Terms include any business terms specified as an Input term. Output Business Terms include all terms specified as an output term. A test case must have at least one input term and one output term.

Create Test Case

Home > All Folders > TPL Insurance > Pricing Tier Rules > Create Test Case

1 Configure Test Case 2 Enter Test Values

* = Required

* Title: Description:

Available Business Terms

| Term |
|---|
| <input type="checkbox"/> Property Risk |
| <input type="checkbox"/> Customer Financial Stability |
| <input type="checkbox"/> Pricing Tier |

Selected Business Terms

Select from the list of available business terms. The terms you input will allow you to specify what is necessary in order to affect the values of the terms selected for output.

Input Business Terms

Property Risk
Customer Financial Stability

Output Business Terms

Pricing Tier

Back Next Save Save and Run Cancel

3. Enter test values for the test case as shown in the following table:

| | Scenario 1 | Scenario 2 | Scenario 3 |
|------------------------------|------------|------------|------------|
| Customer Financial Stability | Excellent | Average | Excellent |
| Property Risk | High | Low | Low |

Note: You are not limited to three test scenarios. You can add additional scenarios by selecting the Add Column button.

4. Enter the expected results for Pricing Tier as shown in the following table:

| | Scenario 1 | Scenario 2 | Scenario 3 |
|--------------|------------|------------|------------|
| Pricing Tier | C | B | Reject |

5. Click Save and Run to run the test case immediately.

A confirmation message lets you know that the Test case save was successful. If the Test Case execution fails, an error message appears.

The Test Case result displays.

The screenshot shows the 'Rule Manager' window with the 'Test Results' tab selected. A confirmation message states 'Confirm: Save Test case successful'. Below this, the 'Input values' table shows data for three scenarios. The 'Output values' table shows the results for the 'Pricing Tier' test case, with 'Expected' and 'Actual' columns for each scenario. The results are: Scenario 1 (Expected: C, Actual: C), Scenario 2 (Expected: B, Actual: B), and Scenario 3 (Expected: Reject, Actual: A). Each row in the output table has a 'Show Trace' link with a green checkmark for success or a red cross for failure.

| Input values | | | | | | |
|------------------------------|------------|------------|------------|--|--|--|
| | Scenario 1 | Scenario 2 | Scenario 3 | | | |
| Customer Financial Stability | Excellent | Average | Excellent | | | |
| Property Risk | High | Low | Low | | | |

| Output values | | | | | | |
|---------------|----------------------------|--------|----------------------------|--------|----------------------------|--------|
| | Expected | Actual | Expected | Actual | Expected | Actual |
| Pricing Tier | C | C | B | B | Reject | A |
| | Show Trace | | Show Trace | | Show Trace | |

Note: Each scenario is marked as success with a green check icon if the expected results match with the actual results; otherwise it's marked as failed with a red cross icon.

6. If one of the tests results is not as expected, click Show Trace next to the red cross icon to see how the engine derived at the actual result.



On the left-hand side of the screen you can see the test input and results. On the right-hand side you can see the rules that were executed. The fields provide information about what happened when each rule was visited.

Order

The order of the rules being applied.

Rule

The name of the rule evaluated. You can click on the Rule name to see the rule definition.

Iteration

An iteration number shows within which iteration of the rules this rule is evaluated.

Status

- ✗ — Condition not satisfied, no action taken.
- ✓ — Condition satisfied, action taken.
- ⓘ — Insufficient data to apply, no action taken.

Summary

The summary column lists values of all terms read and set by the engine.

7. Select the TLP Insurance link from the trail of links at the top of the page to return to the project folder.

More information:

[Create Test Cases](#) (see page 87)

[Test a Project](#) (see page 88)

[Test a Rule set](#) (see page 88)

[Create a Project Test Case](#) (see page 88)

[Modify a Project Test Case](#) (see page 93)

Deploy Rules for TPL Insurance

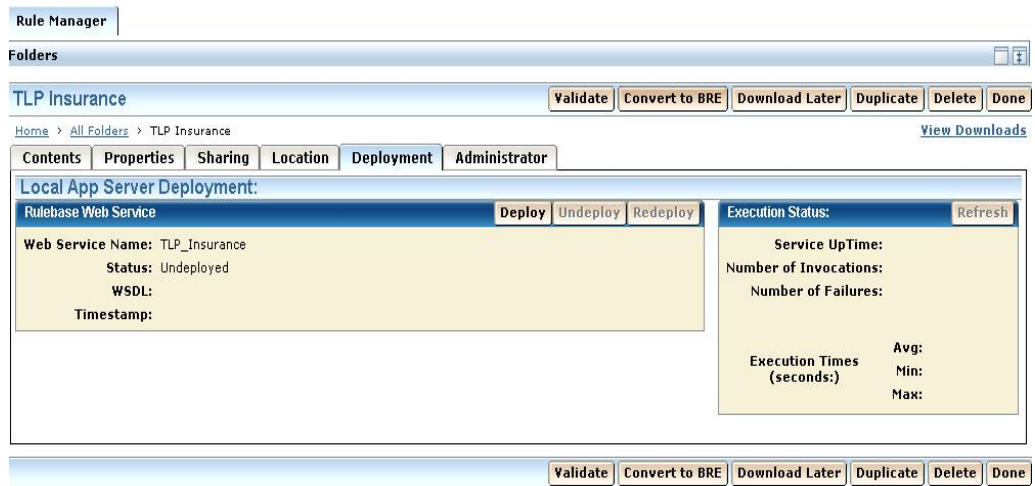
The final step in this exercise is to deploy the rules. When you deploy the rules, the Rule Manager takes the rules and exposes them as a callable web service.

Rule web services are deployed on the local application server for development testing. Rule Manager also generates Web Applications Archives (WAR) that can be deployed on remote application servers when moving rule services into production.

To deploy the rules

1. Click the Deployment tab in the TPL Insurance folder.

The Deployment window appears.



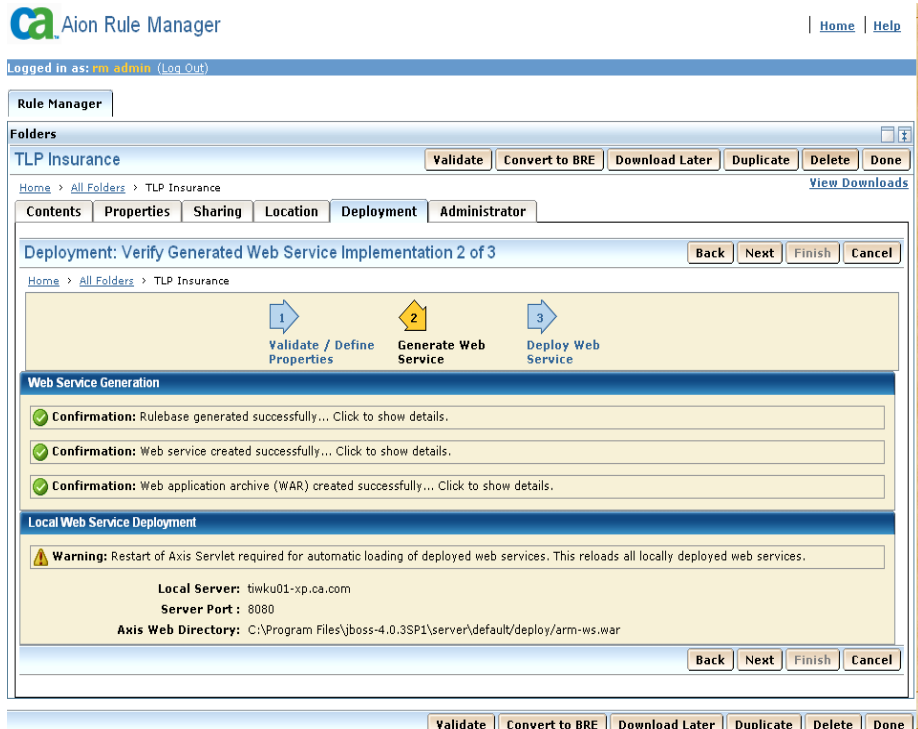
2. Click the Deploy button.

The Rule Manager first validates that the vocabularies and rules are consistent. A confirmation message appears if successful; otherwise, an error message appears.

The screenshot displays the Rule Manager interface for the 'TLP Insurance' folder. The 'Deployment' tab is active, showing a progress bar with three steps: 1. Validate / Define Properties, 2. Generate Web Service, and 3. Deploy Web Service. The first step is currently active. Below the progress bar, the 'Rulebase Validation' section shows a green checkmark and the message 'Confirmation: Validation successful.' The 'Web Service Properties' section shows the 'Web Service Name' as 'TLP_Insurance' and the 'Axis Style/Use' as 'Document/Literal (Wrapped)'. The 'Axis Style/Use' section has two radio buttons: 'Document/Literal (Wrapped)' (selected) and 'RPC/Encoded'. The 'Document/Literal (Wrapped)' option has a tip: 'Tip: Sets Axis Web Service parameters: style="Wrapped" and use="Literal"'. The 'RPC/Encoded' option has a tip: 'Tip: Sets Axis Web Service parameters: style="RPC" and use="Encoded"'. At the bottom of the 'Web Service Properties' section, there are buttons for 'Back', 'Next', 'Finish', and 'Cancel'. At the bottom of the entire interface, there are buttons for 'Validate', 'Convert to BRE', 'Download Later', 'Duplicate', 'Delete', and 'Done'.

3. Click Next to generate the rule web service.

Confirmation messages appear for project generation, web service creation, and web application archival.



4. Click Next to deploy the web service to the local application server.

Note: This will create a callable rule service that you can use for testing with applications.

A confirmation message stating that the project service deployed successfully appears.

Rule Manager

Folders

TLP Insurance Validate Convert to BRE Download Later Duplicate Delete Done

Home > All Folders > TLP Insurance View Downloads

Contents Properties Sharing Location **Deployment** Administrator

Deployment: Verify Deployment Results 3 of 3 Back Next Finish Cancel

Home > All Folders > TLP Insurance

1 Validate / Define Properties 2 Generate Web Service 3 **Deploy Web Service**

Web Service Deployment Results

Alert: Restarted Apache Axis web application

Confirmation: Rulebase service deployed successfully... Click to show details.

Back Next Finish Cancel

Validate Convert to BRE Download Later Duplicate Delete Done

5. Click Finish (or reselect Deployment tab) to view the Deployment status.

Rule Manager

Folders

TLP Insurance Validate Convert to BRE Download Later Duplicate Delete Done

Home > All Folders > TLP Insurance View Downloads

Contents Properties Sharing Location **Deployment** Administrator

Local App Server Deployment:

Rulebase Web Service Deploy Undeploy Redeploy

Web Service Name: TLP_Insurance
Status: Deployed
WSDL: http://dadmo01-2k3-01.ca.com:8080/arm-ws/services/TLP_Insurance?wsdl
Timestamp: 4/17/09 12:36 PM

Execution Status: Refresh

Service UpTime: 0:00:00
Number of Invocations: 0
Number of Failures: 0

Execution Times (seconds): Avg: 0.000
Min: 0.000
Max: 0.000

Export Rulebase for Remote Deployment:

Rule Definition Language: [TLP_Insurance.rdl](#) Tip: Export rulebase RDL for use with AionJRE or JSR-94 wrapper
Web Application Archive: [TLP_Insurance.war](#) Tip: Export rulebase WAR for remote J2EE app server deployment

Validate Convert to BRE Download Later Duplicate Delete Done

6. To view the interface definition of the web service for the deployed rule project, click the link given in the WSDL field. The following window appears:



```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions targetNamespace="urn:impl.TLP_Insurance.ws.ras.rm.aion.ca.com" xmlns:apachesoap="http://xml.apache.org/xml-
soap" xmlns:impl="urn:impl.TLP_Insurance.ws.ras.rm.aion.ca.com" xmlns:intf="urn:impl.TLP_Insurance.ws.ras.rm.aion.ca.com"
xmlns:tns1="urn:beans.TLP_Insurance.ws.ras.rm.aion.ca.com" xmlns:tns2="urn:ras.rm.aion.ca.com"
xmlns:tns3="urn:service.ras.rm.aion.ca.com" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <!--
WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)
-->
- <wsdl:types>
- <schema elementFormDefault="qualified" targetNamespace="urn:impl.TLP_Insurance.ws.ras.rm.aion.ca.com"
xmlns="http://www.w3.org/2001/XMLSchema">
<import namespace="urn:service.ras.rm.aion.ca.com" />
<import namespace="urn:ras.rm.aion.ca.com" />
<import namespace="urn:beans.TLP_Insurance.ws.ras.rm.aion.ca.com" />
- <element name="getAppIFDoc">
<complexType />
</element>
- <element name="getAppIFDocResponse">
- <complexType>
- <sequence>
<element name="getAppIFDocReturn" type="xsd:string" />
</sequence>
</complexType>
</element>
- <element name="loadRulebase">
<complexType />
</element>
- <element name="loadRulebaseResponse">
<complexType />
</element>
- <element name="infer">
- <complexType>
- <sequence>
<element maxOccurs="unbounded" name="inputs" type="tns1:TLP_InsuranceInputs" />
</sequence>
</complexType>
</element>
</schema>
</wsdl:types>
</wsdl:definitions>
```

RDL Link

Lets you view or download the Rulebase Definition Language for external use with CA Rule Engine.

WAR Link

Lets you download the deployed rules as a Web Archive for remote deployment.

Convert to BRE

Integrates projects authored in Rule Manager with existing BRE applications or by customers who want to take advantage of the faster speed of inference of BRE.

User and Group Accounts

CA EEM manages users, roles, and groups. Users with administrator rights can create and maintain users and groups.

Each user must have a *user account* to log in to CA Aion Rule Manager. A user account contains information such as user name, password, name, email address, telephone number, position, and so on.

Policies allow you access to CA Aion Rule Manager services and data. A policy defines access to some aspect of CA Aion Rule Manager activity, like Rule Manager administration. Assignment of a policy to a user or a group grants the ability controlled by the policy to the user or group.

Groups

You can assign users to specific groups to standardize system privileges and to deliver customized content based on their needs and interests. You can create group structures based on organization structures, business processes or products, or any other criteria that apply to your organization.

Create Groups

To create groups

1. Log in to EEM by selecting it from Start, Programs, CA, Embedded Entitlement Manager, EEM UI.
The EEM user interface appears.
2. Select Manage Identities.
The Manage Identities - Users dialog appears.
3. Select the Groups tab.
The Manage Identities - Groups dialog appears.
4. Select the New Group icon located in the User Groups tree window to the left of the Application Groups node.
The New Group window opens.
5. Enter RM Administrators in the Name field.
6. Click Save.
The group is saved.
7. Repeat steps 4-6 to create the RM Authors group.

Associate Groups with Portal Policies

To associate groups with appropriate role-based permissions

1. Log in to EEM by selecting it from Start, Programs, CA, Embedded Entitlement Manager, EEM UI.
The EEM user interface appears.
2. Click the Manage Access Policies tab.
3. Select DUGPOL - Rule Manager Administrators from the Explicit Grants tab on the left.
The Policy user interface appears.
4. In the Identities section, select Application Groups from the Types drop-down list.
5. Type RM Administrators in the Identities field and click the arrow to the right of the field. The group name is added to the list of selected identities.
6. Click Save to complete the operation.

Add Rule Manager Users to CA Aion Rule Manager

You must establish user roles in order to give users access to the applications they need. Roles can be assigned to users, simply by adding users to groups that already have role assignments. Roles have privileges to perform certain types of tasks associated with specified applications. This allows users in a given group to perform specified tasks.

To add users for CA Rule Manager

1. If you have closed the EEM user interface, log in to EEM by selecting it from Start, Programs, CA, Embedded Entitlement Manager, EEM UI.
The EEM user interface appears.
2. Select Manage Identities.
The Manage Identities dialog appears.
3. Select the New User icon located in the Users tree window to the left of the Users node.

The New User window opens.

4. Enter Alex in the Name field.
5. Enter Alex in the First Name field.
6. Enter Smith in the Last Name field.
7. Click Add Application User Details to expose the list of application groups.
8. Select PM Administrators in the Available User Groups list, and click the arrow that points to the right.

The group name appears in the Selected User Groups list.

9. Enter your email address in the Email address field. Messages to Alex during the demo will be addressed to you.
10. Enter Alex in both the New Password and Confirm Password fields.
11. Click Save.

The user is added.

12. Repeat steps 3-11 to define all users described in the following table:

| Property | Alex | Phyllis | Clark |
|----------------|------|---------|-------|
| Name | Alex | Phyllis | Clark |
| Password | Alex | Phyllis | Clark |
| Password Again | Alex | Phyllis | Clark |

| Property | Alex | Phyllis | Clark |
|-----------------|-------------------|---------|-------------------|
| First Name | Alex | Phyllis | Clark |
| Last Name | Smith | Jones | Anderson |
| Member of Group | RM Administrators | | RM Administrators |

Note the following additional information:

- At minimum, include yourself and your email address as a RM Administrators user.
- You can create user accounts manually one at a time or in batch using a CSV (Comma Separated Value) import process.
- When you install CA Aion Rule Manager the installer creates an administrator user (which defaults to rm admin) and associates it with the CA Aion Rule Manager Administrator policy.
- Some of the users you add to these groups participate in a rule management process. In a real life scenario, the CA Rule Manager administrator might set up a Pricing Tier Management group specifically for the purpose of reviewing rules.
- In this example, you assigned the users to groups relating to content and applications, instead of business roles. Remember, you can create workgroups that map to business roles and then assign users to the groups that match their role within the company.

Appendix B: Using Converted BRE Library

While it takes just a click to convert a rulebase created in Rule Manager to a BRE library, a user prepared BRE client program is still necessary to actually carry out inference using the converted library.

This section describes the structure and interface methods provided by the converted BRE library and procedure to invoke inference using it.

After a Rule Manager generated rulebase is converted to a BRE library, the exact behavior of the two may be different depending on the inputs. The difference in supported features between the CA Rule Engine (the JSR-94 component of CA Rule Manager) used by Rule Manager and the BRE rule engine is the source of difference. This section also describes the details of the differences.

This section contains the following topics:

[Library Structure](#) (see page 133)

[Entry Class](#) (see page 134)

[Data Type Differences](#) (see page 138)

[Engine Differences](#) (see page 141)

Library Structure

The conversion strives to preserve the original structure of the Rule Manager rule base. There are additional classes, fields and methods in the BRE library to accommodate the differences between the rule engines used by Rule Manager and BRE.

Entry Class

A class named after the rulebase is generated as the entry class for the converted BRE library. If there are embedded white space characters in the rulebase name, they are converted to underscore characters. If the rulebase name happens to be similar to a keyword defined in BRE, an RM_ prefix is added to the class name.

For each global term user created in Rule Manager, a corresponding field (similarly named) is generated in this entry class. If the name of a global term happens to be similar to a keyword defined in BRE, an rm_ prefix is added to the field name.

For a global term that has been designated as input in Rule Manager, four accessing methods are also generated to get the field, set the field, test if the field is unknown, and reset the field to unknown. The naming of the four accessing methods is based on the name of the global term.

Example: For a field named "shoppingDate", the corresponding accessing methods are called "getShoppingDate", "setShoppingDate", "isShoppingDateUnknown" and "resetShoppingDate" respectively.

For a global term that has been designated as output in Rule Manager, only two accessing methods are generated to get the field and to test if the field is unknown. The naming of the two accessing methods is the same as in the case of input, described above.

The entry class defines a single static instance *this* to accept input values prior to inference and to provide output results after inference. Client program needs to invoke the access methods for global terms described above using this static instance.

The entry class also defines a *doInfer* method that invokes inference and that is the reason why this class is called the entry class. The doInfer method returns null when inference terminates normally. If one or more constraint rules are not satisfied, that method returns an error message string specified by the failed constraint.

In normal usage, client program first invokes the set access methods to set values to the input fields. It then invokes the doInfer method to carry out inference and finally it uses the get and/or isUnknown access methods to obtain results of inference.

Other methods and fields of this class whose names begin with the underscore character are utility fields and methods used by the library to mitigate differences in behavior between the rule engine in BRE and that in Rule Manager. The method that is most likely to be used in a client program is `_formatInstRefList`. This method converts a list of instance references to a string for reporting purposes.

User Defined Classes

For each of the complex type user created in Rule Manager, a corresponding similarly named class is generated in the converted BRE library. If the name of a complex type happens to clash with a keyword defined in BRE, an RM_ prefix is added to the class name.

For each field of a complex type, a corresponding similarly named field is generated in class corresponding to that complex type. If the name of a field happens to be similar to a keyword defined in BRE, an rm_ prefix is added to the field name.

For a field that has been designated as input in Rule Manager, four accessing methods are also generated to get the field, set the field, test if the field is unknown, and reset the field to unknown. The naming of the four accessing methods is based on the name of the global term as described in the Entry Class section. For a field that has been designated as output in Rule Manager, only two accessing methods are generated to get the field and to test if the field is unknown. The naming of the two accessing methods is the same as in the case of input.

In normal usage, the client program needs to instantiate a user-defined class by invoking its Create method, assigning values to the fields of the created instance using its set access methods, and then setting that instance as value to an instance reference field of either the *this* instance in the entry class, or that of another client program instantiated instance of a user-defined class. At the end of inference, client program needs to use the get and/or the unknown test access methods to retrieve results.

Rule Domain Classes

There are four rule domain classes containing all the rules defined in the converted BRE library. Client program normally does not need to interact with the rule domain classes. The AAEntryDomain class contains fixed rules to orchestrate inference. The AAPrepDomain, AAMainDomain and AAWrapUpDomain classes contain user-defined rules converted from the Rule Manager rulebase.

AAEntryDomain Class

This class is used to ensure that constraint rules are posted and invoked first. This class also ensures that default rules are posted and invoked last. It does not contain any user-defined rules and should not be modified.

AAPrepDomain Class

This class is used to host constraint rules of Rule Manager. The `doInfer` method of this class enforces any effectiveness criteria for any rulesets with constraint rules, posts such rulesets and invokes inference over the constraint rules.

For any ruleset with at least one constraint rule, a similarly named method is generated to post the constraint rules in that ruleset and to enforce any effectiveness criteria of any constraint rule. If the name of one such ruleset happens to be similar to a keyword defined in BRE, an `rm_` prefix is added to the method name.

For a constraint rule, a method whose name is constructed from the name of the enclosing ruleset, name is generated with an underscore character in between. This method contains the actual rule implementation.

This class may also contain utility methods generated to handle specific calculations that are provided by the Rule Manager rule engine but not directly available through the BRE rule engine. The name of such a utility method either has a random number suffix or begins with `subInferOf`.

Any fields defined in this class are utility fields to facilitate specific calculations.

AAWrapUpDomain Class

This class is used to host default rules of Rule Manager. The `doInfer` method of this class enforces any effectiveness criteria for any ruleset with default rules, posts such rulesets and invokes inference over the default rules.

For any ruleset with at least one default rule, a similarly named method is generated to post the default rules in that ruleset and to enforce any effectiveness criteria of any default rule. If the name of one such ruleset happens to be similar to a keyword defined in BRE, an `rm_` prefix is added to the method name.

For a default rule, a method whose name is constructed from the name of the enclosing ruleset, name is generated with an underscore character in between. This method contains the actual rule implementation.

This class may also contain utility methods generated to handle specific calculations that are provided by the Rule Manager rule engine but not directly available through the BRE rule engine. The name of such a utility method either has a random number suffix or begins with `subInferOf`.

Any fields defined in this class are utility fields to facilitate specific calculations.

AAMainDomain Class

This class is used to host the other types of rules defined by the Rule Manager. The `doInfer` method of this class enforces any effectiveness criteria for all rulesets, posts the rulesets and invokes inference over the rules inside this class.

For any ruleset defined in Rule Manager, a similarly named method is generated to post the rules other than constraint or default rules in that ruleset and to enforce any effectiveness criteria of any such rule. If the name of one such ruleset happens to be similar to a keyword defined in BRE, an `rm_` prefix is added to the method name.

For a rule, a method whose name is constructed from the name of the enclosing ruleset, name is generated with an underscore character in between. This method contains the actual rule implementation.

This class may also contain utility methods generated to handle specific calculations that are provided by the Rule Manager rule engine but not directly available through the BRE rule engine. The name of such a utility method either has a random number suffix or begins with `subInferOf`.

Any fields defined in this class are utility fields to facilitate specific calculations.

Data Type Classes

The BRE rule engine does not support the `DateTime` and `Duration` types available in Rule Manager. The Rule Manager rule engine uses W3C string format `DateTime` and `Duration` values for I/O purposes. For example, a `DateTime` value of midnight January 1, 2001 is expressed as `2001-01-01T00:00:00` and a `Duration` value of 1 year 2 months 3 days 4 hours 5 minutes and 6 seconds is expressed as `P1Y2M3DT4H5M6S`. The converted library contains the `DateTime` and `Duration` classes to facilitate basic data I/O functionalities. Both classes provide the following three methods:

- `parseW3C`: parses a W3C string format value into a numerical value used internally in BRE engine.
- `formatW3C`: converts an internal numerical value into a W3C format string.
- `formatListW3C`: converts a list of internal numerical values into a list of W3C format strings.

Mapping of Data Types

The following table shows the mapping of data types in Rule Manager to those in BRE:

| Rule Manager Data Type | BRE Data Type |
|--------------------------------------|----------------------------------|
| Boolean | Boolean |
| Number - 0 precision | Integer |
| Number - other precision | Real |
| String | String |
| Datetime | Datetime |
| Duration | Duration |
| Complex type | Class |
| Instance Reference (to complex type) | Pointer (to corresponding class) |

The capability of corresponding data types are not necessarily the same and may cause different inference results.

Data Type Differences

The difference in data types supported mainly relates to the following types offered by CA Rule Engine:

- Number with arbitrary precision and range using fixed-point representation.
- Date and time
- Duration

There are also some notable differences in a few string operations.

Number Type

For BRE, data types for numerical values can be either:

- Integer
- Real

While the BRE integer type can be used to represent an an CA Rule Engine Number with 0 precision, the range of the BRE integer is limited to the 32-bit IEEE integer representation. The BRE real type has to be used to represent an CA Rule Engine Number with a precision other than 0, but the real type is an 8-byte IEEE double precision type which can neither provide more effective digits than about 15 decimal digits nor provide less precision. Therefore, in addition to the range that is limited to the 8-byte IEEE double precision representation, the rounding behavior of the BRE real type is generally different from that of the Number type in CA Rule Engine which uses fixed-point representation that provides virtually unlimited range and precision.

Date and Time

BRE has both a Date and a Time class. The BRE Date class can only represent a date and the BRE Time class can represent both date and time. Therefore the BRE Time class is used as the basis for representation of the DateTime type in CA Rule Engine. However, the range of date and time supported by the BRE is limited since the BRE Time is backed by a 32-bit IEEE integer representing the number of seconds since midnight of January 1, 1970 UTC. This representation has a valid range of 1970-01-01 00:00:00 to 2038-01-18 19:14:07. All date time specifications such as effectiveness criteria and results of date time arithmetic calculations must be inside this range for the converted BRE library to behave correctly.

In BRE, you can specify a date time literal to be local time only. In CA Rule Engine, you can optionally specify such a value as UTC.

In CA Rule Engine, date time arithmetic is calendrically accurate. The BRE Time class provides limited support for creating a date time value from date time components. One cannot enter a component value outside the normal range. This limitation made it virtually impossible to carry out calendrically accurate date time arithmetic.

Example: You cannot compute the exact value of a date that is 10 months from a given date by adding 10 to the month component value. In the converted BRE library, average length of a calendar year (365.2425 days) and average length of a calendar month (1/12th of a year) are used for such calculations.

CA Rule Engine uses W3C string format for specifying date time values. BRE formats date time values in a somewhat different string. To support the W3C format, a customized BRE DateTime class that extends the built-in Time class is introduced to handle parsing and formatting of the W3C format.

Duration

CA Rule Engine uses W3C string format for specifying duration values also. Internally, duration computation is carried out in 64-bit long integer format representing the duration in milliseconds. Since the range, which is around 300 million years, is so large that it is virtually limitless for any practical purposes. BRE does not have a built-in type for duration at all. When one Time value is subtracted from another, the result is an integer representing the difference in seconds. Therefore a customized Duration class, introduced by the converted library, extends the built-in Integer type to provide parsing and formatting of the W3C format and to make duration a distinct type. On parsing a W3C format duration value, the result is the same duration expressed in number of seconds. That means a duration value in BRE cannot exceed approximately 68 years.

Also as mentioned in the preceding section, BRE duration values containing years and months have to be converted to number of seconds using average lengths of year and month and thus duration arithmetic may not yield same results as in CA Rule Engine. When formatting a BRE duration value into W3C string, it is only possible to express it in a form that is normalized to the year.

String Operations

CA Rule Engine supports regular expressions in string replace operation. Since BRE doesn't support regular expressions, the replace operation must be literal in original RM rulebase for the converted BRE library to behave correctly.

CA Rule Engine follows the Java convention of 0-based indexing. BRE uses 1-based indexing. The conversion tool tries to simulate 0-based index by automatically adding 1 to index value on input and subtracting 1 from index value on output. That makes the converted BRE library take the same index input as the original RM rulebase and generate the same index output. However, if an output index value is to be used in client BRE application, client may need to adjust the indexing base.

Engine Differences

The rule languages used by CA Rule Engine and BRE were designed with different set of goals and so there are also subtle differences in inference engine behaviors.

When Unpending a Rule Premise

CA Rule Engine is smarter in pending resolution by using smart AND-ing and OR-ing conditions that caused pending. Therefore, CA Rule Engine is more likely to unpend a rule when the BRE engine will still pend the corresponding converted rule.

Example: Consider the following conceptual rule:

```
IF A OR B
THEN
  ...
END
```

Assume that A is false and the rule is currently pending on its premise reference to B.

CA Rule Engine will unpend the rule whenever:

- B is resolved; or
- A changes value

The BRE Engine will unpend the rule only when B is resolved. The CA Rule Engine will also unpend when any referenced field changes value.

Example: If a rule is pended on an "age" reference

`thePerson.mySpouse.myManager.age`

the rule will unpend not only on age resolution but also on change of spouse or change of manager.

The BRE Engine won't take into account such changes. As a result, CA Rule Engine may fire or fail rules where the BRE Engine only pends them.

When Unpending a Rule Action

CA Rule Engine unpends a rule action in a pin-point fashion, whereas the BRE Engine restarts an unpended rule by fully re-evaluating the rule premise

Example: Consider the following conceptual rule:

```
IF A AND B
THEN
  X = C
  Y = D
END
```

Assume that the rule is currently pended on its action reference to D. CA Rule Engine unpends the rule only when D is resolved - and, when unpending the rule, CA Rule Engine restarts the rule action in a pin-point fashion - *that is* at the second action statement.

The BRE Engine likewise unpends the rule only when D is resolved - but, when unpending the rule, the BRE Engine restarts the rule by re-evaluating the rule premise.

As a result, the BRE inferencing results may differ because:

- Premise conditions (A or B) have changed since the rule was last pended; or
- The BRE Engine re-executes prior action statements - and their behavior changes due to value changes (*for example* for C) since the rule last pended.

Null Instance Handling

CA Rule Engine provides special handling of NULL instance dereference in the context of Boolean expression. Instead of generating an error condition, CA Rule Engine returns false as the value of that expression.

Example: Assuming any or all of thePerson, thePerson.mySpouse and thePerson.mySpouse.myManager evaluates to NULL, CA Rule Engine returns false for the following Boolean expressions:

```
thePerson.mySpouse.myManager.isHealthy  
thePerson.mySpouse.myManager.age > 30
```

The BRE engine may generate an error condition in such cases.

Time Base in Client/Server Scenarios

CA Rule Engine distinguishes client local time from server local time, whereas the BRE Engine does not make this distinction.

CA Rule Engine accepts a client GMT offset as an initialization parameter so that it can correctly distinguish between client time and server time. The BRE Engine makes no such distinction.

If client and server are in different time zones, the BRE Engine enforces effectiveness criteria differently from what CA Rule Engine does.

- The BRE Engine enforces the criteria using the server's local time.
- CA Rule Engine enforces the criteria using the client's local time.

As a result, the two engines may post rules differently.

Name Clashes for Iteration Variables

CA Rule Engine is more tolerant than BRE in allowing the same identifier to be used for different purposes. Since identifiers are matched in a case-insensitive manner in both rule engines, those differing only in case are considered to be the same. Using the same identifier for different purposes may cause name clashes to occur after conversion to BRE.

Example

CA Rule Engine allows the same iteration variable to be used at multiple levels of nested iterative operations, where the iteration variable of an inner iteration overrides the same iteration variable of an outer iteration in the scope of the inner iteration. However, nested loops in BRE must use distinct loop variables. Therefore, if a valid RM rulebase containing nested iterative operations shares the same iteration variable, name clashes may occur after conversion to BRE. Another situation where name clashes may occur is that the same identifier is used both as a name of a type and as a name of a field.

When BRE detects name clashes, it will append an underscore character ("_") to the declaration of the clashed identifier when loading the rulebase and that causes any references to that identifier to become invalid.

To avoid potential name clash problems after conversion to BRE, it is recommended to use distinct identifiers when designing the original RM rulebase.

Appendix C: Using and Creating Iterator

This section contains the following topics:

[Create and Use Iterator in a Rule](#) (see page 145)

[Create Iterator Variable on Expression Builder Page](#) (see page 146)

Create and Use Iterator in a Rule

If your business vocabulary contains terms that are contained in collections (Set/Array/List), and if you want to iterate over for your business logic, then you should create Iterator for those lists. You can create and use an iterator for all rule types.

To create and use iterator in a rule

1. From Add Iteration page, click Add Iteration.
A screen with Variable name and Set is displayed.
2. Do the following:
 - Enter the name of iterator variable, say iterator.
 - From the drop down menu, select the set on which this iterator works, say empListInstance.emps.
3. Click Save Iteration.

The iterator variable is saved and is ready to use in expression builder for expressions involving terms of type collection.

Create Iterator Variable on Expression Builder Page

The Iterator variables created on Expression Builder page are used with Collection templates like Select All, Select One, etc. as it allows user to access individual element of collection.

To create Iterator variable on Expression Builder page

1. Click Add Iteration under Iterations section.
A new row is added under Iteration section with variable name and set.
2. Click Edit link.
The Edit page is displayed.
3. Enter a variable name and select set.
4. Click Save.
The variable is saved.

Index

A

- add
 - rule manager users • 130

B

- Business Rules • 35
- Business Vocabulary Manager
 - about • 47
 - create term • 48
 - create vocabulary • 48
 - creating • 47

C

- CA Portal
 - checked out content portlet • 32
 - Library services • 31
 - notifications portlet • 32
 - search for content • 33
 - workplace • 30
- configure
 - CA Aion Rule Manager for the TLP Insurance Tour • 113
- create
 - new workgroup accounts • 129
 - rule set • 53
 - rules • 55
- creating
 - business terms • 47
 - projects • 46
 - rule sets • 53
 - rule versions • 103
 - rules • 54

D

- Default rule • 42
- duplicating
 - rule sets • 95
 - rules • 98

E

- editing properties
 - rules • 102
- expression builder • 60

I

- introduction
 - CA Aion Rule Manager • 9
 - TLP Insurance Tour • 113

L

- log in
 - CA EEM • 116
- logging in
 - access • 29
 - Rule Manager • 43

M

- Manage
 - business rules • 35
- moving
 - rule sets • 96
 - rules • 100

P

- projects
 - about • 36
 - browsing • 44
 - creating • 46
 - validating • 94

R

- roles • 113
- rule manager
 - deploy • 124
- Rule Manager
 - organization • 36
 - permission settings • 105
- available users • 105
- sharing • 106
- rule sets
 - about • 37
 - browsing • 44
 - creating • 53
 - duplicating • 95
 - editing properties
 - rule sets • 96
 - moving • 96
 - rule versions • 97

- deleting • 104
- reverting • 103
- rules
 - about • 37
 - check-in and check-out rules • 97
 - creating automatically • 82
 - creating manually • 55
 - creating versions • 103
 - deleting • 99
 - duplicating • 98
 - editing properties • 102
 - moving • 100
- rules set
 - create • 55
 - edit • 118
 - test • 121
- rules types • 38

T

- TLP Insurance • 113

U

- User Folder
 - TLP insurance • 116
- users
 - Add Rule Manager user • 130
 - Rule manager users • 10
 - workgroup accounts • 128
- user names and passwords
 - CA Aion Rule Manager • 29
 - Rule Manager • 44

V

- version history • 102

W

- work
 - workgroup accounts • 128
- workplaces
 - Portlet • 30