



PARTICIPANT GUIDE

WEBMETHODS BUSINESS RULES

629-6CE

Software AG
Internal Use Only!

This publication is protected by international copyright law. All rights reserved. No part of this publication may be reproduced, translated, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of Software AG.

Software AG and all Software AG products are either trademarks or registered trademarks of Software AG. Other product or company names mentioned herein may be the trademarks of their respective owners.



webMethods Business Rules

629-6CE

Notes:



Copyright

- This publication is protected by international copyright law. All rights reserved. No part of this publication may be reproduced, translated, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of Software AG.
- Software AG and all Software AG products are either trademarks or registered trademarks of Software AG. Other product or company names mentioned herein may be the trademarks of their respective owners.

Software AG Training | Page 2

Notes:

Welcome to Software AG Training!

Housekeeping items

- Class hours / Refreshments
- Restrooms / Smoking
- Emergency exits
- Sign-in sheets

For everyone's benefit, please:

- Turn off/mute phones
- Check e-mail only at breaks
- Refrain from side discussions during the presentation. It can be very distracting.
- Feel free to ask questions during the lecture.

Notes:

Table of Contents

- ▶ 1. Introducing webMethods Business Rules
- ▶ 2. Creating Simple Decision Tables
- ▶ 3. Rule Sets, Rule Execution, and Processing Modes
- ▶ 4. Verifications
- ▶ 5. Internal Event Rules
- ▶ 6. Decision Trees
- ▶ 7. Service Actions and New Data Actions
- ▶ 8. Built-in Functions
- ▶ 9. Deployment to IS and MWS
- ▶ 10. Working with the Rules Management Console
- ▶ 11. BPM invokes Decision Entities

Notes:

Table of Contents

- ▶ 12. Business Rule interacts with BPM (Process Actions)
- ▶ 13. Business Rules invoked from BPM User Tasks
- ▶ 14. Business Rules invoked from IS Services
- ▶ 15. External Event Rules and Event Models
- ▶ 16. Business Rule Metadata
- ▶ 17. Logging, Auditing and Administration
- ▶ 18. Wrap Up

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



1

Introducing webMethods Business Rules

Notes:

Objectives

At the end of this chapter you ...

- Can explain the benefits of use cases of webMethods Business Rules
- Can list the main components of by webMethods Business Rules
- Know about the naming conventions and terminology used by Business Rules

Notes:

 software AG

Business Rules Defined

What are Business Rules?

A Business Rule is a statement or a set of statements that defines or constrains some aspect of a business process

What do you use them for?

- Enhance structured processes with dynamic elements
- Empower business users to quickly change process behavior
- Enable organizations to be structured and flexible at the same time

What business challenges relate to Business Rules?

- Market dynamics require organizations to rapidly change business decisions ...
- If you can't change them in real-time you are too slow

Previous Purchase	Price of Item	Discount
>10	>50	20%
1-10	>50	15%
	=>50	2%

Software AG Training | 1 - 3

Notes:

Business Rules – Key Benefits

Reduce Costs

- Fewer resources, less time to change decisions
- Lower fines, legal costs from bad decisions
- Reduced IT costs to implement decisions

Improve Decision Making

- Clear policies and procedures
- Consistently applied across channels, systems, processes
- Increased accuracy through business users control

Business Agility

- More rapid response to business threats
- Fewer missed opportunities
- Faster time to market

Rules capabilities allow for more flexible process automation

Externalizing and automating decisions is the key to process agility

Software AG Training | 1 - 4

Notes:

webMethods Business Rules and webMethods BPMS

▪ **webMethods Business Rules**

- webMethods suite component
- Powerful and advanced Business Rules optimized for webMethods BPMS

▪ **Integration** between webMethods Business Rules & webMethods BPMS

- Bi-directional, synchronous & asynchronous
- Integration of webMethods Business Rules in Designer, Process Engine and Task Engine



Notes:



webMethods Business Rules - Key Component & Functionality

Rule Design



- Fully integrated into Designer
- Graphical building - Decision tables, Decision Trees, Event Rules, Rule Sets,...
- Execute (test) rules in Designer using an embedded Rule Engine – no server connection required
- Data Model and service integration with Integration Server

Rule Engine



- Bi-directional integration with BPMS
- Multiple processing modes for sequential or powerful inferential rule execution
- Event-enabled – Event Rules for internal/external events
- Available at design time and runtime

Rules Management Console



- Fully integrated into My webMethods
- Embedded business-centric editing (browser-based)
- Shared run-times with webMethods BPMS
- Hot Deploy Business Rules when needed

Software AG Training | 1 - 8

webM Business Rules:

- Powerful & advanced business rules optimized for our BPMS, integrated with wM:
- Bi-directional, Synchronous & Asynchronous
- Integration of webMethods Business rules in Designer, PE & TE and MWS

DETAILS:

Business Rules Development

The design-time component of webMethods Business Rules is fully integrated into Software AG Designer. This includes a special perspective for rules development – complete with a business rules editor that allows graphical design of decision entities and a Rules Explorer view that allows easy organization of business rule projects. Furthermore, rules can be executed (for testing purposes) directly from Designer.

Actions

Business rules can take actions (apart from simply returning decision results). The actions supported are service actions (where business rules can trigger service invocations) and process actions (where business rules can trigger process actions such as start process, suspend process, and resume process).

BPM Integration

Bi-directional integration with BPM exists – wherein business processes in BPM can invoke business rules – and business rules can invoke business processes (as described under ‘Actions’).

Decision Tables , Decision Trees and Event Rules

webMethods Business Rules supports definition of decisions using both a Decision Tables metaphor (wherein rules are defined using an Excel-like interface) and a Decision Tree metaphor. Event Rules are also supported where the rules respond to events happening during rule execution.

Rule Sets

Grouping of decision entities (decision tables, event rules) is supported. A rule set groups decision entities – and this is useful both from an organization perspective and also from an execution standpoint – wherein different decision entities within a rule set might be triggered based on inference. BPM can invoke both decision tables and rule sets.

Business Rules Runtime

The rules engine runs inside of webMethods Integration Server and has full inference capabilities (for Decision Tables only), wherein rule executions can lead to results/actions which in turn can trigger other rules.

Rules Management Console

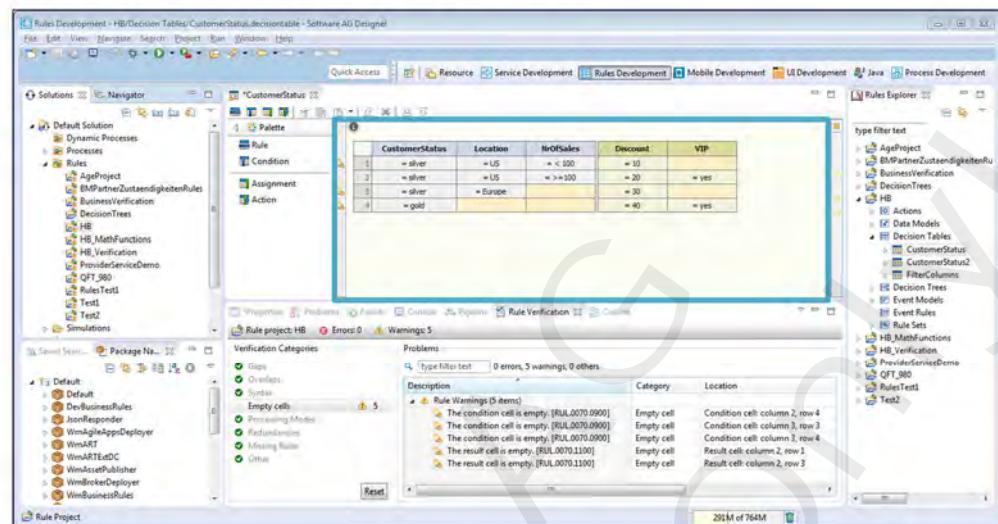
The Rules Management Console is a Web-based interface hosted in My webMethods for editing of business rules. Business users, based on privileges defined in My webMethods, can access and edit business rules from this Web-based interface. They can also hot deploy these business rules, if needed, so that rule changes can take effect immediately.

Business Rules Metaphors – Decision Tables



Decision Table:

- Tabular decision entity
 - Very structured
 - Ensuring completion and that you have analyzed every possible combination of conditions
 - Compact method to view all of the available decisions at once
 - Can deal with large amount of data and become very large

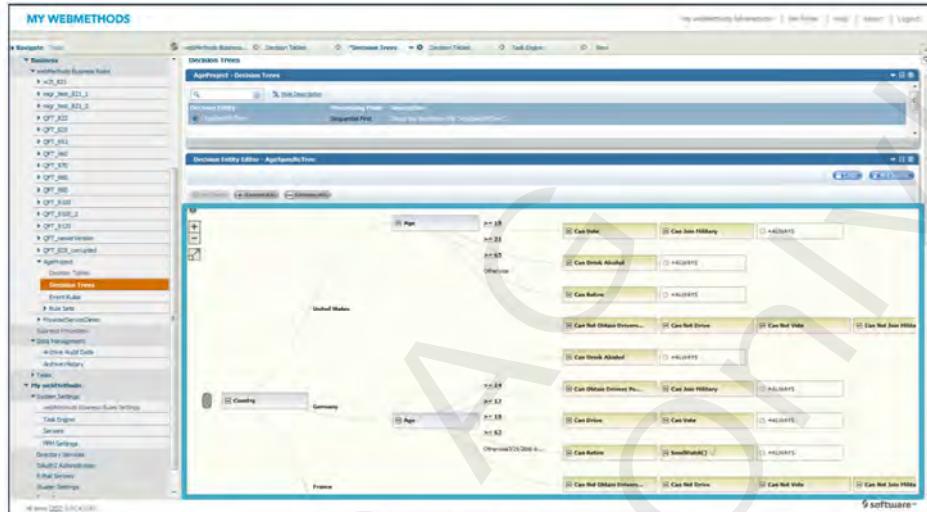


Decision Table (in Designer)

Software AG Training | 1 - 7

Notes:

Business Rules Metaphors – Decision Trees



Decision Tree (in Rules Management Console)

- Graphical decision entity
 - Easy to create
 - Easy to read and explain
 - Can easily deal with missing data - conditions that do not matter are absent
 - Quickly become very large and hard to read

The screenshot shows the Oracle Rules Management Console interface. On the left, there's a navigation sidebar with options like 'Servers', 'RDB Settings', 'Directory Services', 'Identity Administration', 'Audit Services', and 'Audit Settings'. The main area displays a 'Decision Tree' with nodes and conditions. One node is expanded, showing 'Can Active' and 'Can Not Active' as possible outcomes. The bottom right corner of the interface has a status bar with the text '9 software'.

Software AG Training | 1 - 8

Notes:

webMethods Business Rules - Key Advantages

1. Decision Making Flexibility
2. Rule Actions
3. Manual Decisions
4. Rule-based Task Assignments
5. Event-driven Results and Actions
6. Rule Verifications



Software AG Training | 1 - 9

Notes:

1. Decision Making Flexibility...

The diagram illustrates the flexibility of decision making in webMethods Business Rules. It shows two separate business processes interacting with a central decision table.

Left Business Process: A user icon says, "I want to call the calculateDiscount() service that wraps a Decision Table called 'Calculate Discount' to get the discount rate". This process leads to a gear icon, which then points to a decision table.

Decision Table:

Previous Purchase	Price of Item	Discount
>10	>50	20%
1-10	>50	15%
	>=50	2%

Right Business Process: A user icon says, "I need the discount rate". This process leads to a database icon, which then points to the same decision table.

Summary: A green box at the bottom right states: "Just ask for **WHAT** you want – **no need** to specify **HOW** to get what you want".

Software AG Training | 1 - 10

Notes:

...1. Decision Making Flexibility

Business Rules

Previous Purchase	Price of Item	Discount
>10	>50	20%
1-10	>50	15%
	>=50	2%

Rules Reorganized and Changed

Previous Purchase	Price of Item	Discount Tier
>10	>50	Tier 1
1-10	>50	Tier 2
	>=50	Tier 3

D1

D2

D3

Customer Status

Customer Status	Discount Tier	Discount
Gold	Tier 1	20%
Gold	Tier 2	20%
Silver	Tier 1	15%
Silver	Tier 2	10%
Regular		1%

- Rule Engine figures out, based on the data you pass to it, which rules to invoke

Software AG Training | 1 - 11

Notes:

2. Rule Actions ...

Many Business Rules solutions allow you to only return values

webMethods Business Rules allows you to configure Actions in the result columns

Salary	Score	Level
>50k		Gold
20k – 50k	>=50	Silver
	>=50	Regular

Salary	Score	Level	Send Welcome Kit	Start Upgrade Process
>50k		Gold	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20k – 50k	>=50	Silver		<input checked="" type="checkbox"/>
	>=50	Regular		

Assignments Service Actions Process Actions

Software AG Training | 1 - 12

Rule Actions: Most decisions include actions in them. Very rarely would you find rules that are purely returning things like interest rate or discount percentage. Most of the time, decisions are like “for salary > 50k, irrespective of the score, the /level/ should be upgraded to GOLD and when the level is upgraded to GOLD, a welcome kit must be sent to the customer (a service action!) and the upgrade process should be started (a process action!). webMethods Business Rules allows you to specify holistic rule definitions, which include actions.

Apart from allowing you to specify holistic rule definitions, this allows for injection of dynamic behavior in process context. For example, the service invocation above (Send welcome kit) and process action above (Start Upgrade process) were never “coded” into a process model – but that behavior is injected when the rule gets invoked.

... 2. Rule Actions

Action types:

- **Service Actions**

Invoke any IS service

- **Process Actions**

Start, Join, Suspend, Resume, Cancel, Fail processes

- **New data actions**

Create data facts based on decisions

- **Manual Actions**

See next case

“Manual Decisions”

Possibilities:

- Various Process Actions allow to interface with running processes or to start new ones
- Manual Actions allow dynamic injection of User Tasks into a workflow
- New Data Actions allow to create data facts based on decisions

Rules like the following can be easily realized: When a rule determines that a customer has defaulted on loan of one type, suspend all other process instances (loan apps, other apps) for the customer

3. Manual Decisions

- Not all business decisions are fully automated
- Manual decisions are a part of everyday life

Salary	Score	Rate
>200	>800	3%
<=200	700-800	4%
<=200	>700	Manual

When a Business Process invokes this Decision Table, for most cases, the control returns immediately with a value (3%, 4%). For manual decisions, a User Task gets queued!

- webMethods Business Rules allows you to define these types of rules
- New User Task steps are dynamically injected into the process
- Ability to go from manual decisions to automated decisions and vice-versa

Software AG Training | 1 - 14

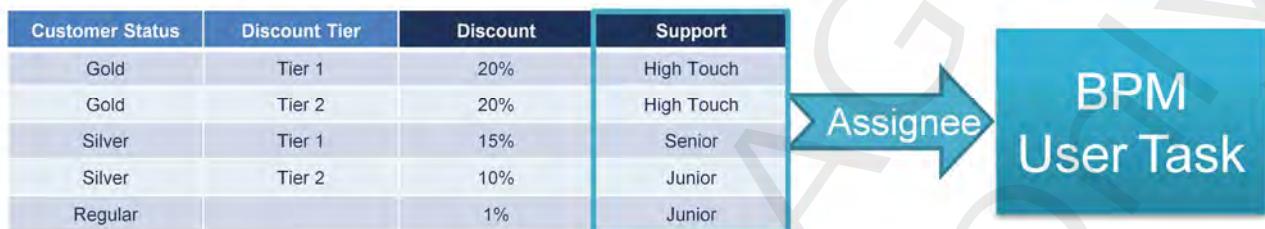
Manual decisions: Most rule vendors (and certainly BPM vendors) would have you believe that all decisions taken in the rule engine are fully automated!! But business decisions are RARELY (if at all) fully automated. Many times, the answer is “it depends” – and a human determines what the interest rate should be, for example. Specifying these kinds of partially automated decisions is important NOT ONLY from holistic definition standpoint – but also important because as mentioned above, reality requires manual decisions! webMethods Business Rules supports manual decisions.

Example: So in the above example, when a business process invokes this decision entity, many times, it gets 3% or 4% back immediately – but sometimes (third row) – a human task is queued and gets assigned to a human. When that human makes the decision, the process continues.

Of course the business rules side is free to change decisions from automated to manual and vice-versa, WITHOUT requiring any changes on the business process side!

4. Rule-based Task Assignments

- Manage Task assignments as part of Business Rules
- Like processes, User Tasks also don't need to know WHICH decision entity to call - they can simple invoke a Decision Entity and the Rule Engine can determine which rules to trigger

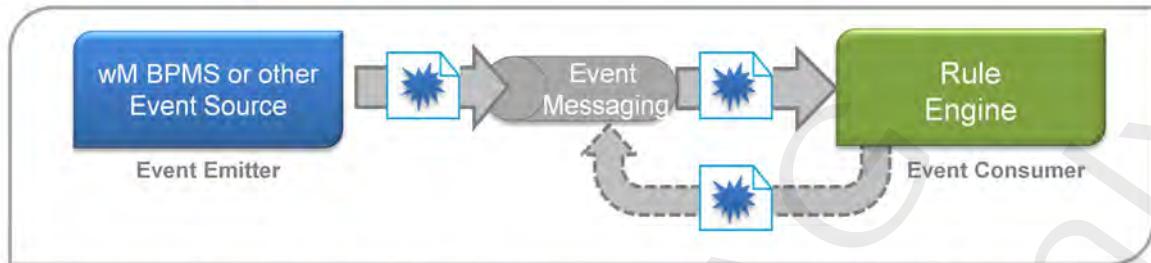


- No need to split real business rules into some rules in the Rule Engine and Task assignments in the Task Engine
- So, new behavior can be affected by simply changing business Rules – no need to change the User Task definitions
- You can update rules in the Rules Management Console to change task assignments

- **Holistic rule definitions:** This is something you will observe in a lot of BPM vendors' products – "business rules" are specified in the rules engine – and "assignment rules" are specified in the task engine (or BPM engine). Assignment rules – which determine who (participants) who work on a specific user task type are just as important business decisions as decisions like interest determination and discount determination. So why separate them into two places? webMethods Business Rules allows holistic rule definitions – and the individual components of a business process can all get the information they need from rules engine directly. No need to split decisions. In the following example, typically, "Discount" is specified in the rule engine and "Support" is determined by task assignment rules. webMethods Business Rules allows you to define them together. And webMethods BPMS allows you to leverage this holistic definition in processes and tasks.

5. Event Enablement

- Event Rule support in webMethods Business Rules



- Consumes internal and external Events (coming from BPM and others)
 - Example: Step started, process completed, fraud detected etc.
- Associated event could pass received Event data to related Decision Entities which can perform event-based assignments, actions, ...
- Business Rules may emit Events too using Service Actions
- Example: Rule processing can detect a complex event ("Fraud Detected"). webMethods Business Rules can be used to specify what to do when such an event happens and can make decisions and perform different actions based on the Event data.

Notes:

6. Rule Verifications

- Verification Categories:
 - Gaps and Overlaps
 - Syntax
 - Empty cells
 - Redundancies and Missing Rules
 - ...
- Optimistic and Pessimistic Rule Verification
- Verifications can be performed
 - by Rule Developer in Designer
 - by Business User in
Rules Management Console (RMC)

Verification Categories

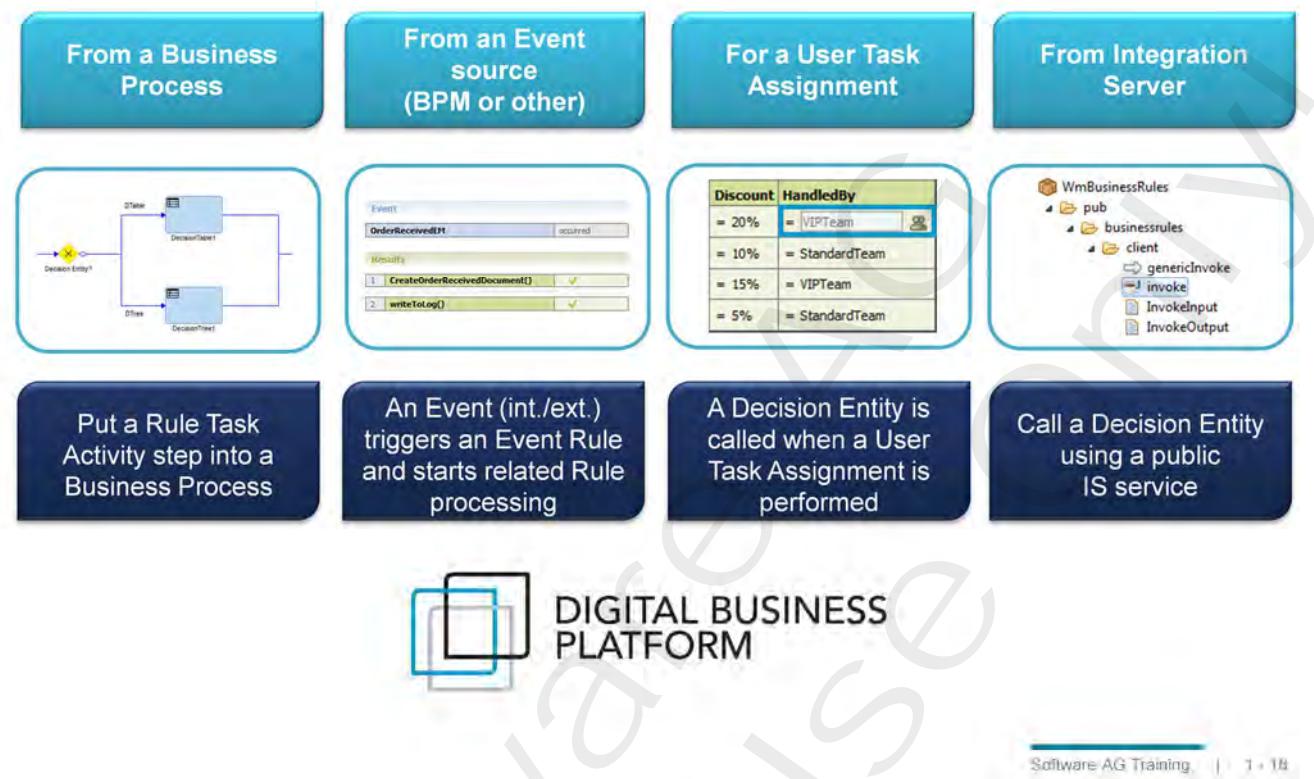
Category	Status	Count
Gaps	Warning	5
Overlaps	Warning	3
Syntax	OK	0
Empty cells	Warning	0
Processing ...	Warning	0
Redundancies	Warning	1
Missing Rules	Warning	96
Other	OK	0

Reset

Software AG Training | 1 - 17

Notes:

Using Business Rules in the Digital Business Platform



Notes:

Software AG

Some More Rule Terminology



The book is bound in orange leather with gold-colored edges. The title 'RULE TERMINOLOGY' is printed in gold letters on the front cover. The background is a light blue surface.

Software AG Training | 1 - 19

Notes:

software

webMethods Business Rules - Based on Jess

- Jess® is a Rule Engine for the Java platform 
- Provided twice by webMethods Business Rules in
 - Designer for development in testing
 - IS as runtime environment
- Developed by Ernest Friedman-Hill of Sandia National Labs
- It provides rule-based programming suitable for automating an expert system, and is often referred to as an expert system shell
- Based on the declarative paradigm that continuously applies a collection of rules to a collection of parameters (facts) by a process called pattern matching. Rules can modify the collection of facts, or they can execute any Java code.
- The Jess Rules Engine uses the Rete algorithm


the Rule Engine for the Java™ Platform

Software AG Training | 1 - 20

Notes:

Rete Algorithm

- The Rete algorithm is an efficient pattern matching algorithm for implementing production rule systems
- The Rete algorithm was designed by Dr Charles L. Forgy
- The word 'Rete' is Latin for 'net' or 'comb'
- http://en.wikipedia.org/wiki/Rete_algorithm



Software AG Training | 1 - 21

Notes:

Rule, Parameters, Parameter Elements & Data Models

- Rules are based on **Parameters (Facts)** consisting of **Parameter Elements**. Each parameter belongs to a **Data Model**.

Example:

Each Rule consists of:

- a number of **Conditions (IF)**
- a number of **Results (THEN)**

Example: IF (A.one = "abc")
THEN B.three := "def"

Software AG Training | 1 - 22

Notes:

Decision Entities and Rule Sets

- Rules of the same structure can be combined to a Decision Entity*
(e.g. Decision Table, Decision Tree,..)
- Decision Entities can be combined to a Rule Set

The diagram illustrates the relationship between a Rule Set and its components. At the top, a box labeled "Rule Set" contains a sub-section titled "ASimpleExample" which includes two items: "A1" and "A2". Arrows point from "A1" and "A2" down to a central area labeled "Decision Entities (Decision Tables)". From this central area, arrows point to two separate boxes labeled "A1" and "A2". Each box displays a decision table with two columns: "A.one" and "B.three" (in A1) or "B.three" and "B.four" (in A2). The tables contain rows with values like "1 = abc" and "1 = def".

* webMethods Business Rules currently supports Decision Entities of type Decision Table, Decision Tree, and Event Rule

Software AG Training | 1 - 23

Notes:

Processing Modes

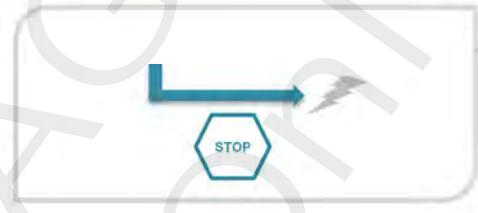
Inferential*

- Rule execution is based on making inferences, NOT on order
- Draw a conclusion from a given information with the help of a rule



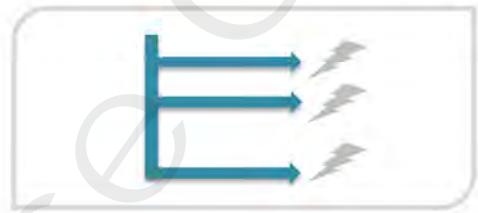
Sequential First

- Execution is order based
- If a rule fires, the evaluation and execution is stopped
- Only one rule can fire



Sequential All

- Execution is order based
- All rules are evaluated and executed from top to bottom
- More than one rule can fire



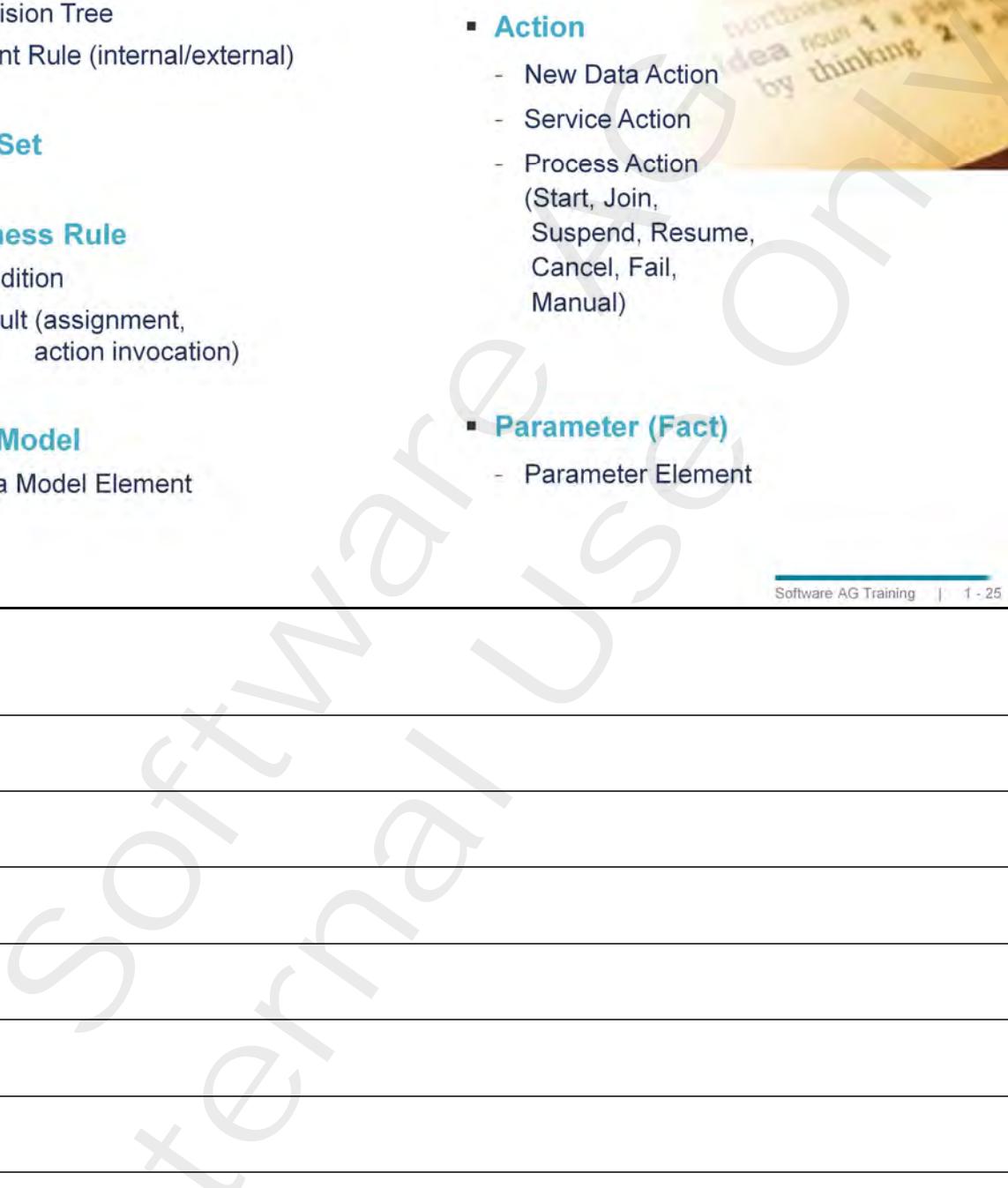
* currently Decision Tables only

Software AG Training | 1 - 24

Notes:

Software AG

webMethods Business Rules - Rules Terminology



- **Decision Entity**
 - Decision Table
 - Decision Tree
 - Event Rule (internal/external)
- **Rule Set**
- **Business Rule**
 - Condition
 - Result (assignment, action invocation)
- **Data Model**
 - Data Model Element
- **Event Model**
- **Action**
 - New Data Action
 - Service Action
 - Process Action (Start, Join, Suspend, Resume, Cancel, Fail, Manual)
- **Parameter (Fact)**
 - Parameter Element

Software AG Training | 1 - 25

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



2

Creating Simple Decision Tables

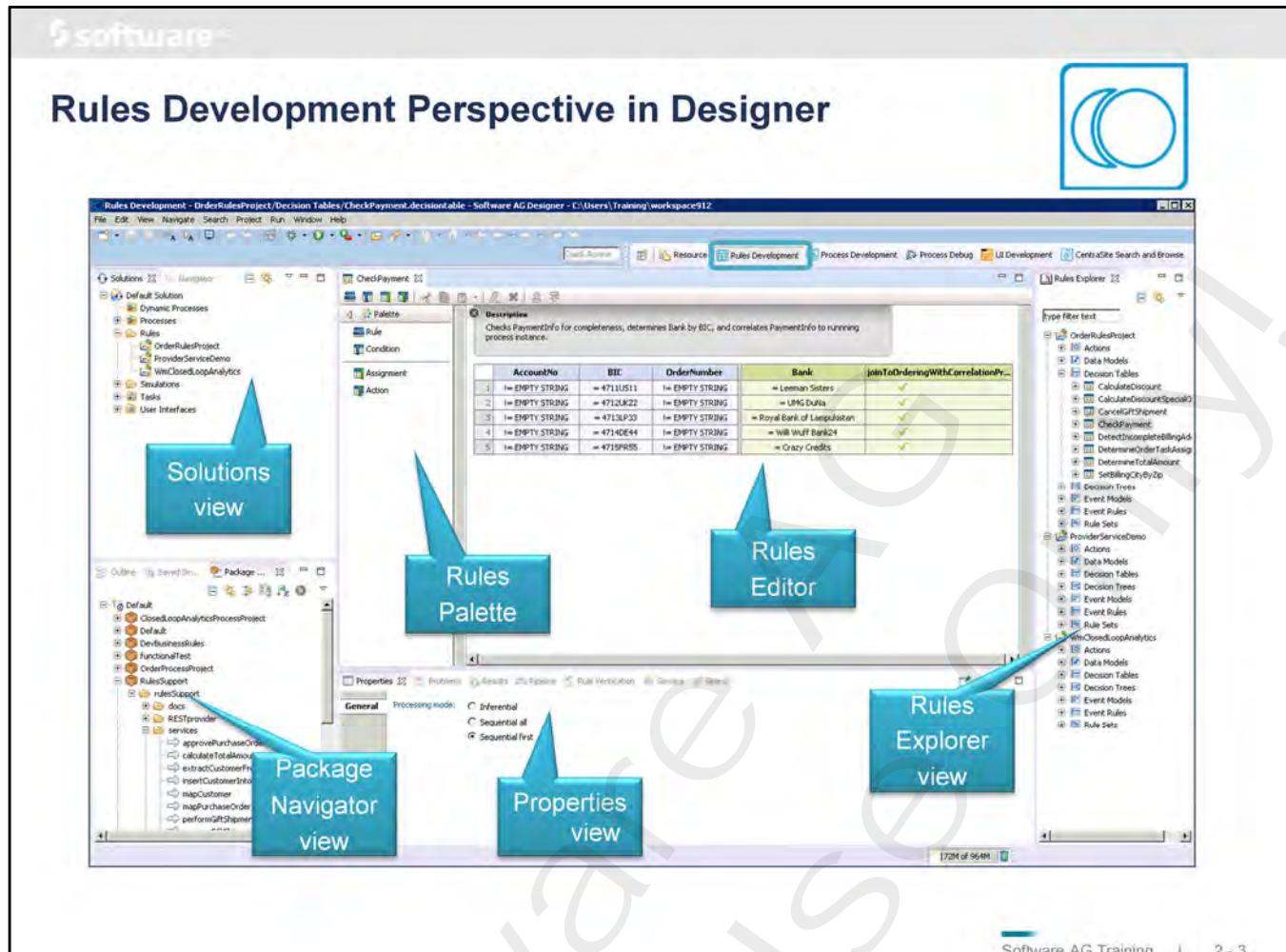
Notes:

Objectives

At the end of this chapter you ...

- Used the Business Rules Perspective in Designer
- Created your first Decision Entity of type Decision Table
- Tested your Decision Table in Designer
- Used Run Configurations

Notes:



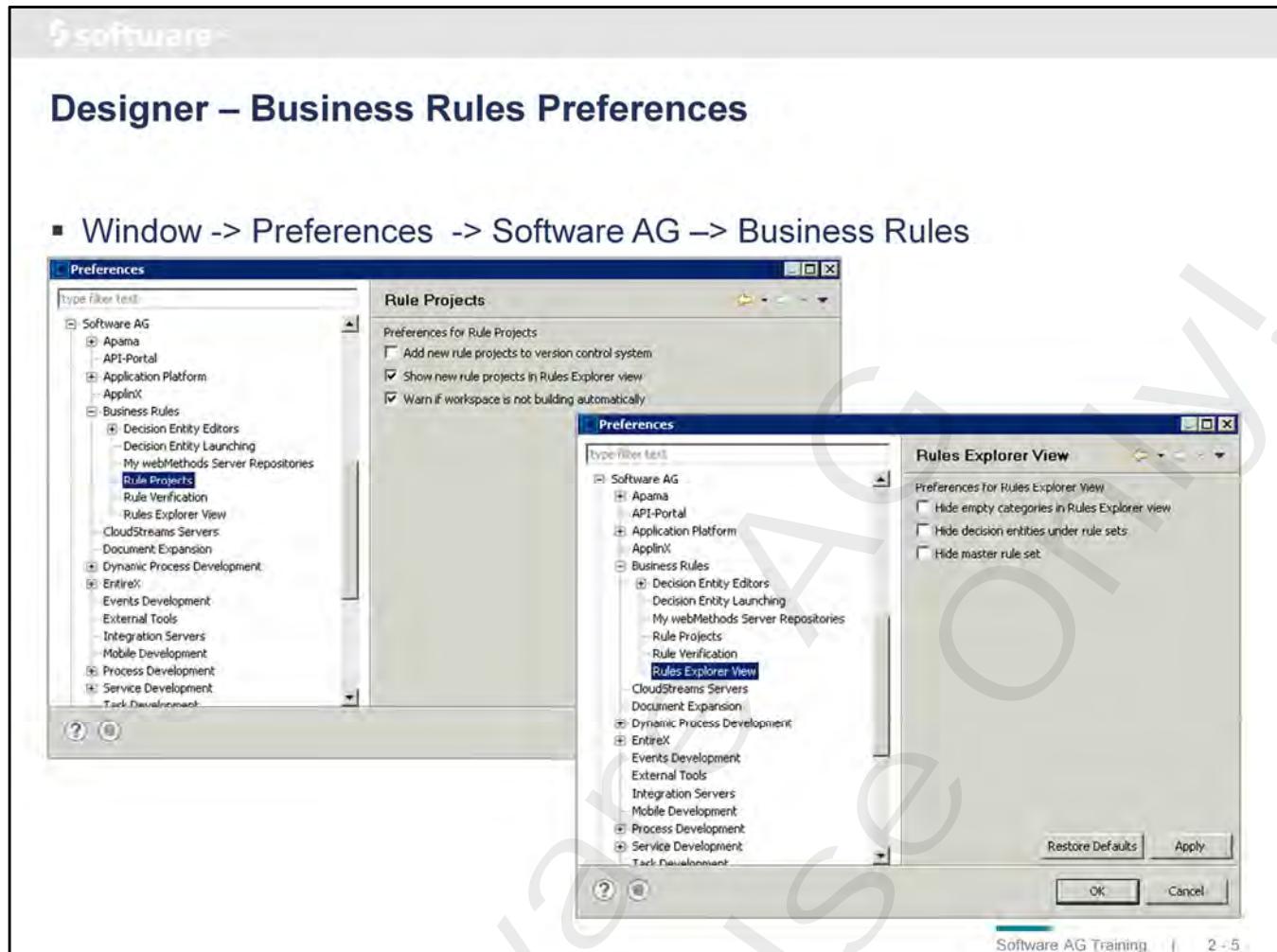
Notes:

webMethods Business Rules in Designer

- Use Rules Development perspective for rule development in Designer.
Rule assets belong to a Rules projects.
- webMethods Business Rules projects:
 - Support for local workspace and VCS
 - Support for local and shared (CentraSite) Metadata
 - Import/Export and "Deployment" to IS/MWS
- Rule projects in Solutions view
 - Assets NOT shown here
- Rule projects in Rules Explorer view
 - Usually you work here
- Navigator view not recommended for Rule Assets



Notes:



Notes:

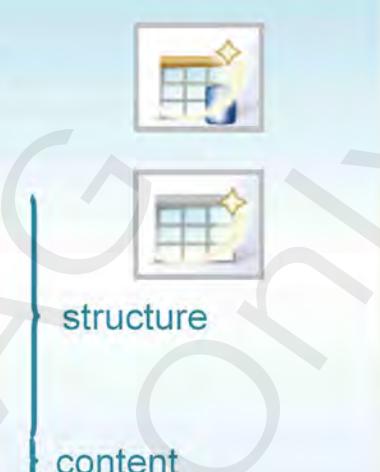
Software AG

Getting Started - Your First Decision Entity of type Decision Table

1. Create new Rule project
2. Create Data Model(s) from IS Document Types
3. Create Decision Entity of type Decision Table
 - a. Name and description
 - b. Parameter(s) with type and direction
 - c. Input parameter elements for conditions
 - d. Output parameter elements for results
 - e. *Optional:* Result actions
 - f. Add Rules
4. Save and test in Designer

See also documentation "*webMethods BPM Rules Development Help*" 

Software AG Training | 2 - 6



The diagram illustrates the components of a Decision Entity. It features a vertical blue line with two windows above it. The left window is labeled 'structure' and the right window is labeled 'content'.

Notes:

1. New Rule Project

- Add Rules Project from Rules Explorer or Solutions view:

The screenshot displays two windows. On the left is the 'New Rule Project' dialog box, which has 'DemoRuleProject' selected in the 'Project name' field. On the right is the 'Rules Explorer' window, where a context menu is open over the 'DemoRuleProject' node. The context menu includes options like 'New', 'Hide Project', 'Delete' (highlighted with a red border), and 'Verify'.

- Project added to Rules Explorer and Solutions view

Notes:

2. Create Data Models from IS Document Types

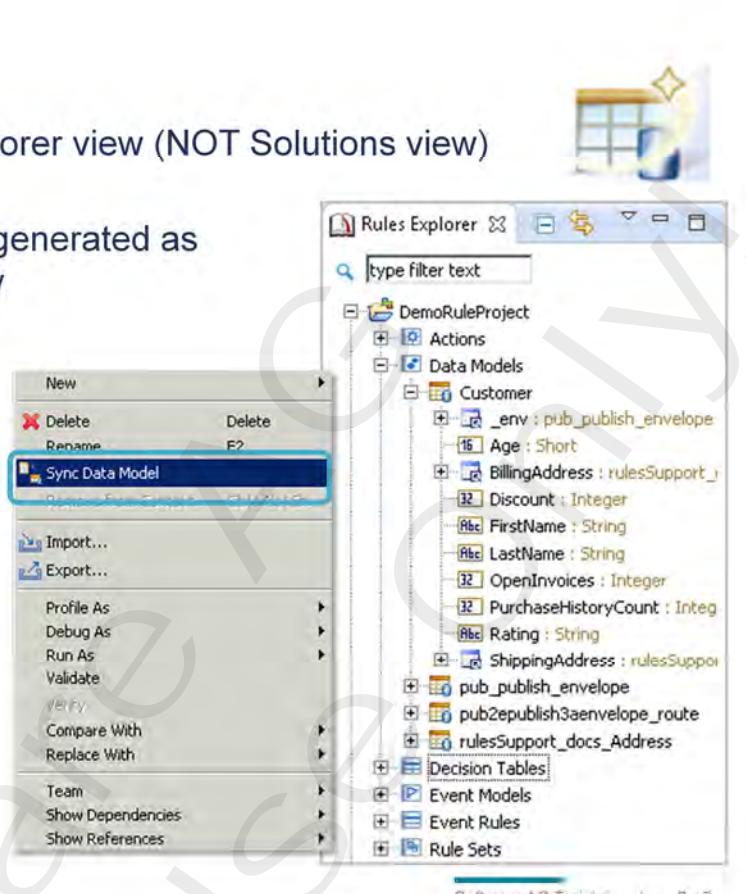
- Add Data Model to Rules Project:

Software AG Training | 2 - 8

Notes:

Resulting Data Model

- Data Model added to Rules Explorer view (NOT Solutions view)
- Embedded IS Document Types generated as additional Data Models on the fly
- Origin is kept in Event Model
- Option for later Sync



The screenshot shows the Software AG Rules Explorer interface. On the left, a context menu is open over a Data Model named 'Customer'. The menu options include New, Delete, Rename, E2, Sync Data Model (which is highlighted), Import..., Export..., Profile As, Debug As, Run As, Validate, Verify, Compare With, Replace With, Team, Show Dependencies, Show References, and Help. On the right, the Rules Explorer tree view shows a project named 'DemoRuleProject' containing 'Actions', 'Data Models', 'Decision Tables', 'Event Models', 'Event Rules', and 'Rule Sets'. Under 'Data Models', there is a 'Customer' node which is expanded to show properties such as '_env : pub_publish_envelope', 'Age : Short', 'BillingAddress : rulesSupport_...', 'Discount : Integer', 'FirstName : String', 'LastName : String', 'OpenInvoices : Integer', 'PurchaseHistoryCount : Integ', 'Rating : String', 'ShippingAddress : rulesSupport_...', 'pub_publish_envelope', 'pub2epublish3aenvelope_route', and 'rulesSupport_docs_Address'.

Notes:

3.a-b: Create Decision Table Structure

▪ Provide name, Rule Set, Processing mode and (optional) description
 ▪ Master Rule Set is mandatory

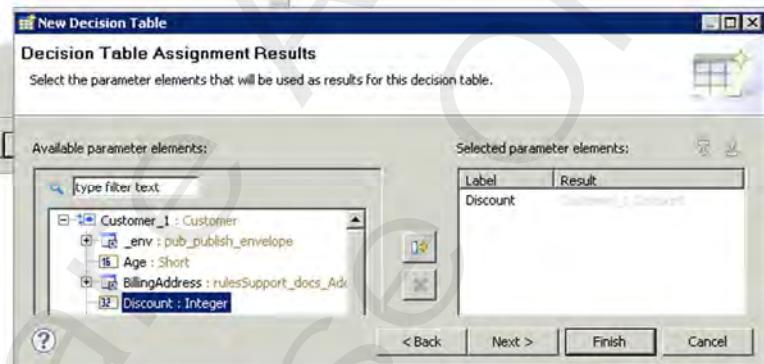
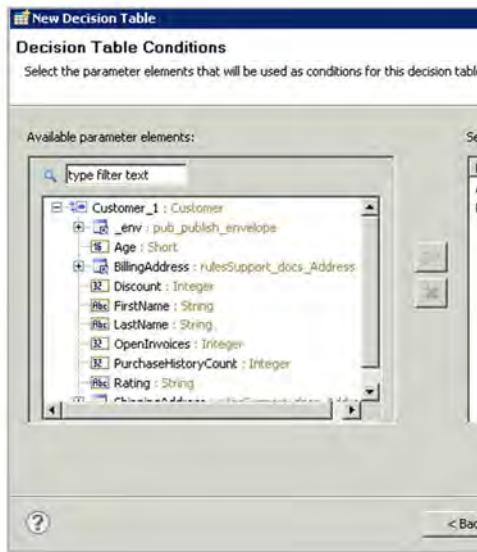
Named Instance of a Data Model with I/O direction (JESS: Fact)

Name	Type	I/O	Any
Customer_1	Customer	Both	
		Input	
		Output	
		Both	

Notes:

3.c-e: Create Decision Table Structure

- Select Parameter Elements for Rule conditions
 - Rename label, if desired

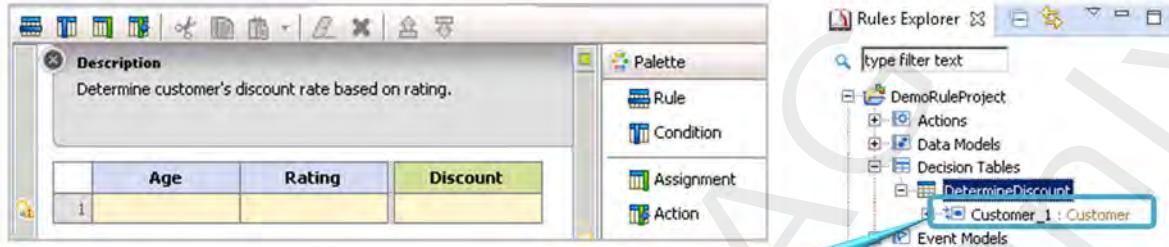


*Additionally, Actions to be executed as a result could be defined here, too (see later topic)

Software AG Training | 2-11

Notes:

Resulting Decision Table Structure

- Empty Decision Table opened in Decision Table Editor and added to Rules Explorer view:
- Add/modify Rules in Decision Table Editor

Software AG Training

| 2 - 12 |

Notes:

3f. Adding Rules - Decision Table Editor and Palette

The screenshot shows the webMethods Business Rules interface. On the left is the **Decision Table Editor**, which displays a table with three columns: **Age**, **Rating**, and **Discount**. A single row is present with values 1, blank, and blank respectively. Above the editor is a **Description** field containing the text: "Determine customer's discount rate based on age and rating". To the right is the **Palette**, which lists four categories: **Rule**, **Condition**, **Assignment**, and **Action**. A toolbar above the editor has several icons with associated callout boxes:

- Add a new Condition to the end**
- Add a new Assignment to the end**
- Add a new Action to the end**
- Clear selected elements**
- Change rule sequence**
- Rule – Add a new Rule**
- Condition – Add a new Condition**
- Assignment – Add a new Assignment**
- Action – Add a new Action**

Other UI elements include a **Collapse/show description** button, a **Delete selected elements** button, a **Cut, copy & paste rules** icon, and a **Warning/error indicator**.

- Use appropriate icons in Decision Table Editor panel or use icons from Palette to customize Decision Table:
 - Add/modify Rules (rows)
 - Cut, copy and paste Rules (rows)
 - Add/remove/rename columns (Conditions, Assignments, Actions)
 - Change Rule sequence

Software AG Training | 2 - 13

Notes:

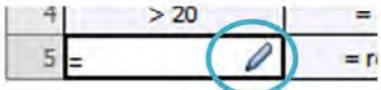
	Age	Rating	Discount
1	<= 20	= gold	= 5
2	> 20	= gold	= 10
3	<= 20	= silver	= 3
4	> 20	= silver	= 5
5			= 0

- To create simple condition/ assignments right-click cell and select appropriate comparison operator
- Offered operators are data type-sensitive

Software AG Training | 2 - 14

Notes:

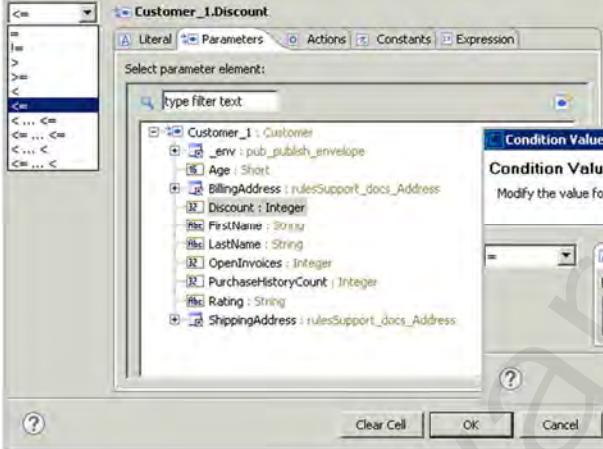
Extended Cell Editing



- Pencil opens Extended Cell Editor
- Allows literals, parameters, constants, actions (see later) and expressions (see later) to be used in conditions & assignments

Condition Value Modification

Condition Value
Modify the value for 'Age' as required.



The dialog shows a dropdown menu with operators like =, >, <, etc. Below it is a tree view of 'Customer_1.Customer' fields, including Age (selected), BillingAddress, Discount, FirstName, LastName, OpenInvoices, PurchaseHistoryCount, Rating, and ShippingAddress. At the bottom are OK and Cancel buttons.

Condition Value Modification

Condition Value
Modify the value for 'Age' as required.



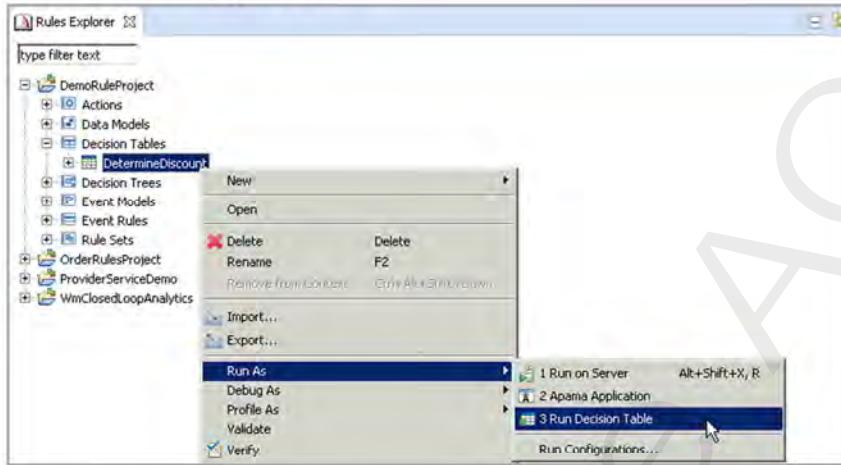
The dialog shows a dropdown menu with operators like =, >, <, etc. Below it is a 'Select constant:' dropdown with options EMPTY STRING and NULL. At the bottom are OK and Cancel buttons.

Software AG Training | 2 - 15

Notes:

4. Save and Test Decision Table in Designer...

- Saved Decision Table can be tested standalone in Designer – no IS connection required



- Prompts for Save if Decision Entity is "dirty"

Notes:

...4. Save and Test Decision Table in Designer

- Provide inputs on appearing panel
- Check "Include empty values for String Types", if necessary

▪ Results are shown in Service view:

Notes:

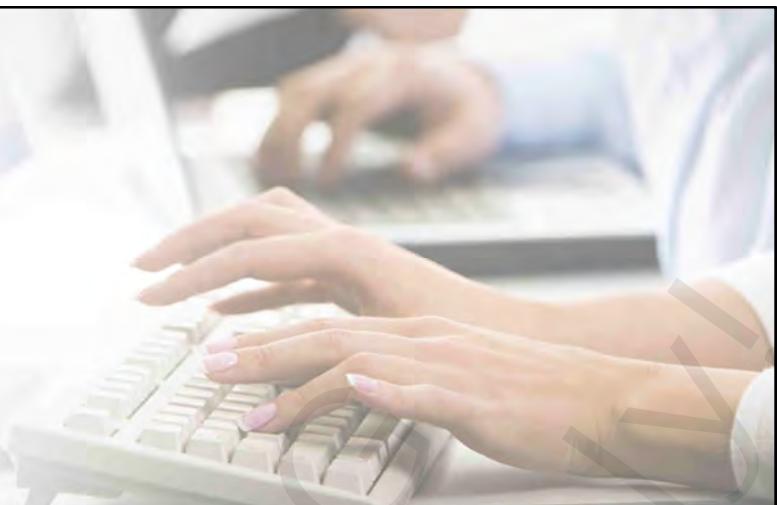
Test Decision Table using Run Configuration

The screenshot shows the webMethods Business Rules interface. On the left, there is a navigation tree with various projects and configurations. A blue box highlights the 'Run Configurations...' option under the 'Run' menu. To the right, a 'Run Configurations' dialog box is open. It has tabs for 'Decision Entity' and 'Logging'. The 'Decision Entity' tab is selected, showing a project named 'DemoRuleProject' and a decision entity named 'DetermineDiscount'. Below this, there is an 'Input' section with a table for setting parameters. The table contains the following data:

Name	Value
Customer_1	
FirstName	Paul
LastName	Newman
Age	66
Rating	gold
Discount	
PurchaseHistoryCount	3

At the bottom of the dialog, there are buttons for 'Save...', 'Load...', 'Clear', 'Apply', 'Revert', 'Run', and 'Close'. The status bar at the bottom right of the dialog box displays 'Software AG Training | 2 - 18'.

Notes:



Exercise 1

- In this exercise, you will create a Rule project, import Data Models from existing IS documents and configure a first simple Decision Table based on Customer parameter. The Decision Table determines a discount rate based on various customer parameter elements. You will test your Decision Table in Designer.

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



3

Rule Sets, Rule Execution, and Processing Modes

Notes:

Objectives

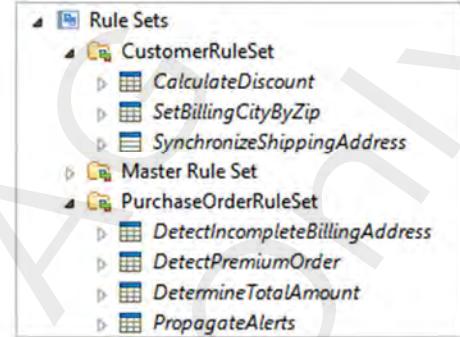
At the end of this chapter you ...

- Can explain the concept of Rule Sets
- Added/removed Decision Entities to/from Rule Sets
- Tested and ran Rules Sets in Designer
- Understand the process of rule execution in the Rule Engine
- Can explain and use the different Processing Modes
- Used the "Any" parameter in Decision Entities

Notes:

Rule Sets

- Rule Sets enable you to combine Decision Entities of type
 - Decision Table
 - Decision Tree
 - Event Rulescontained in one Rule Project for common execution
- Belong to a Rule Project
- Can't be nested
- Decision Entities can belong to multiple Rule Sets
 - Every Decision Entity belongs to default Master Rule Set



The screenshot shows a tree view of rule sets. At the top level is 'Rule Sets'. It contains two main branches: 'CustomerRuleSet' and 'PurchaseOrderRuleSet'. 'CustomerRuleSet' contains three decision entities: 'CalculateDiscount', 'SetBillingCityByZip', and 'SynchronizeShippingAddress'. 'PurchaseOrderRuleSet' contains four decision entities: 'DetectIncompleteBillingAddress', 'DetectPremiumOrder', 'DetermineTotalAmount', and 'PropagateAlerts'. A 'Master Rule Set' node is also visible between the two main branches.

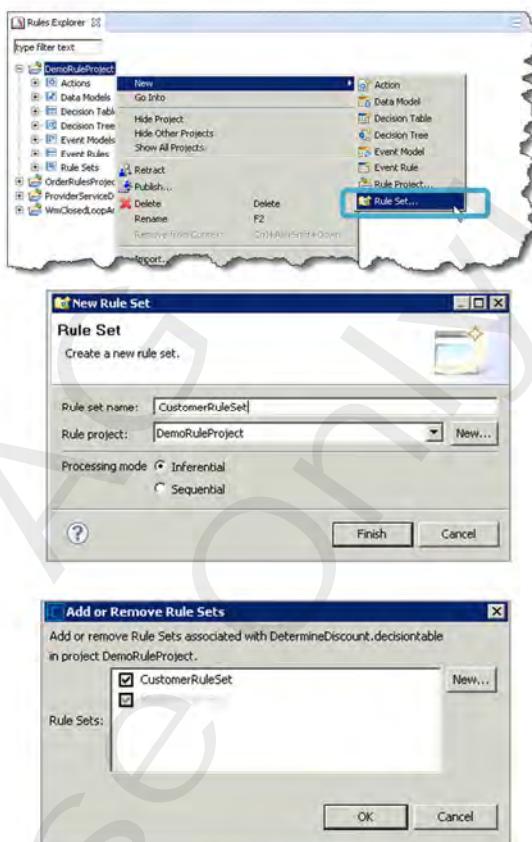
Notes:

Rule Sets

- Create Rule Set in Rules Explorer view
 - Select Processing mode (see chapter 5)

- Add Decision Entities to Rule Set in Rules Explorer view:
 - Choose "Rule Set" from Decision Entities context menu *OR*
 - Drag Decision Entity onto Rule Set

- Remove Decision Entities from Rule Set in Rules Explorer view:
 - Choose "Rule Set" from Decision Entities context menu *OR*
 - Delete Decision Entity from Rule Set



The first screenshot shows the 'Rules Explorer' window with a context menu open over a 'Rule Sets' item. The 'New' option is highlighted, and a callout box points to it.

The second screenshot is a 'New Rule Set' dialog box. It has fields for 'Rule Set name' (CustomerRuleSet), 'Rule project' (DemoRuleProject), and 'Processing mode' (Inferential). Buttons for 'Finish' and 'Cancel' are at the bottom.

The third screenshot is an 'Add or Remove Rule Sets' dialog box. It lists 'CustomerRuleSet' under 'Rule Sets' and has a 'New...' button. Buttons for 'OK' and 'Cancel' are at the bottom.

Notes:

Processing Modes...

- webMethods Business Rules supports three Processing Modes:
 - Inferential (*default*)
 - Sequential all
 - Sequential first
- Processing modes can be set in Designer when creating:
 - Decision Tables
 - Decision Trees*
 - Rule Sets**

Important: Processing Mode of a Rule Set overwrites the Processing Modes of the Decision Entities that are part of it.

Processing mode	<input checked="" type="radio"/> Inferential <input checked="" type="radio"/> Sequential all <input type="radio"/> Sequential first
<input type="checkbox"/> Process aware	
Description:	

* Decision Trees only allow Processing Modes Sequential first and Sequential all, so could not belong to an inferential Rule Set

** Rule Sets only allow Processing Modes Inferential and Sequential

Notes:

...Processing Modes

- Processing Modes can be modified* afterwards using the
 - Properties view
 - Rules Explorer view

The screenshot shows the Software AG Rules Explorer interface. On the left is a tree view of projects and rule sets. On the right is a context menu for a selected item named 'CalculateDiscou'. The menu includes options like New, Open, Delete, Rename, Import..., Export..., Profile As, Debug As, Run As, Validate, Verify, Compare With, Replace With, Team, Rule Sets..., Show Dependencies, Show References, and Processing mode. The 'Processing mode' option has a submenu with three choices: Inferential (selected), Sequential all, and Sequential first.

* Rule Sets only allow Processing Modes Inferential and Sequential.
Decision Trees only allow Processing Modes Sequential first and Sequential all.

Software AG Training | 3 - 6

Notes:

Processing Mode Inferential...

Evaluate
Activate
Fire

- Supported by Decision Tables and Rule Sets
- Inferential Rules can be contained in one Decision Table
- Inferential Rules can be contained in multiple Decision Tables combined in a common Rule Set
- Inferential Rules belonging to one Decision Table or Decision Tables contained in a common Rule Set may invoke each other. Prerequisite:
 - Identical Facts = named Parameter (elements) of same Data Model

The screenshot shows two 'Decision Table Parameters' dialog boxes side-by-side. Both boxes have the title 'New Decision Table' and 'Decision Table Parameters'. They both show a table with one row selected, labeled 'A.1'. The table has columns: Available data models, Selected parameters, Name, Type, I/O, and Amu. In the 'Available data models' column, there are two entries: 'A' and 'B'. In the 'Selected parameters' column, there is one entry: 'A.1' with Type 'A', I/O 'Both', and Amu checked. Below the table are buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. To the right of the dialog boxes is a tree view titled 'Decision Tables'. It shows a hierarchy: 'Decision Tables' -> 'A1' -> 'A.1 : A' (with sub-items 'one : String' and 'two : String'). It also shows 'A2' -> 'A.1 : A' and 'B.1 : B' (with sub-item 'four : String'). There is also a separate entry 'three : String'. At the bottom right of the slide is the text 'Software AG Training | 3 - 7'.

Notes:

...Processing Mode Inferential

- **Inferential Decision Tables:**
Order of rules does not imply order of execution
- **Inferential Rule Sets:**
The order of decision entities does NOT imply order of execution

The screenshot shows two windows side-by-side. The top window is titled 'New Decision Table' and has a sub-section titled 'Decision Table'. It contains fields for 'Decision table name' (set to 'DetermineDiscount'), 'Rule project' (set to 'DemoRuleProject'), 'Rule sets' (with a dropdown menu), and 'Processing mode' (with three radio button options: 'Inferential' (selected), 'Sequential all', and 'Sequential first'). Below this is a 'Description:' field with some placeholder text. The bottom window is titled 'New Rule Set' and has a sub-section titled 'Rule Set'. It contains fields for 'Rule set name' (set to 'CustomerRuleSet'), 'Rule project' (set to 'DemoRuleProject'), and 'Processing mode' (with two radio button options: 'Inferential' (selected) and 'Sequential'). At the bottom of both dialogs are 'Finish' and 'Cancel' buttons.

The default for both decision tables and rule sets is inferential processing mode; inferential processing means that the order of rules in a decision table or the order of decision entities in a rule set does not imply order of execution. Instead, the rules or decision entities are executed in arbitrary order.

Inferential Rule Processing - Basic Cycle of Rule Execution

- Parameters (Facts) are **inserted**
- Each Rule is **evaluated**; if its conditions are true it is **activated**
- All activated Rules **fire**; Parameter Elements may be modified by Rule results
- If a Parameter was modified, all rules that have a condition on this Parameter are **evaluated** again
 - A Rule cannot cause itself to re-evaluate
 - A Rule will not be re-evaluated if the condition Parameter Element was not changed (old value and new value are equal)
- This **cycle** continues until no Rules are activated anymore



Notes:

Inferential Processing - A Simple Example...

Evaluate
Activate
Fire

- Data Models:

Data Models

- A
 - one: String
 - two: String
- B
 - four: String
 - three: String

- Decision Tables

Decision Tables

- A1
 - A_1 : A
 - one: String
 - two: String
 - B_1 : B
 - four: String
 - three: String
- A2
 - A_1 : A
 - B_1 : B

- Input Data:

Name	Value
B_1	
three	
four	
A_1	
one	abc
two	

The screenshot shows the webMethods Business Rules interface. On the left, there's a tree view of 'Data Models' containing objects A and B with their respective attributes (one, two for A; four, three for B). Below that is a tree view of 'Decision Tables' with entries for A1 and A2. A1 contains decision tables for A_1 and B_1, each with two rows. A2 contains entries for A_1 and B_1. To the right, there are two decision table editor windows. The top window for A1 shows a rule with condition 'A.one = abc' and consequence 'B.three = def'. The bottom window for A2 shows a rule with condition 'B.three = def' and consequence 'B.four = ghi'. At the bottom, there's a table titled 'Input Data' with columns 'Name' and 'Value', listing attributes from B_1 to two. A large watermark 'Internal Use Only' is diagonally across the page.

Software AG Training | 3 - 10

Notes:

...Inferential Processing - A Simple Example

The diagram illustrates the execution flow of inferential processing:

- Evaluate:** Both rules in Decision Tables A1 and A2 are evaluated.
- Activate:** A1₁ is activated because its condition is true; A2₁ is not activated because its condition is false.
- Fire:** A1₁ fires: B.three is set to "def".
- Re-evaluation:** A2₁ is re-evaluated because a field of B_1 was changed; A1₁ is not re-evaluated because rules cannot cause itself to re-evaluate.
- Activation:** A2₁ is activated because its condition is true.
- Fire:** A2₁ fires: B.four is set to "ghi".
- Re-evaluation:** A1₁ is not re-evaluated because no condition field of parameter A_1 was changed.
- Activation:** A2₁ is not re-evaluated because a rule cannot cause itself to re-evaluate.
- Completion:** No rules are activated; execution stops and the result is returned.

Decision Tables (A1 and A2):

	A1	B.three
1	= abc	= def

	A2	B.four
1	= def	= ghi

Service Result:

Name	Value						
A_1	abc						
B_1	<table border="1"> <tr> <td>one</td> <td>abc</td> </tr> <tr> <td>four</td> <td>ghi</td> </tr> <tr> <td>three</td> <td>def</td> </tr> </table>	one	abc	four	ghi	three	def
one	abc						
four	ghi						
three	def						

STOP

Notes:

Rules Inference and "Any"

Evaluate
Activate
Fire

- To enforce inference in a Rule Set using different facts, check Decision Entity parameter with "Any":

The screenshot shows the webMethods Business Rules interface. At the top right, there are three buttons: Evaluate, Activate, and Fire. Below them, a sidebar titled 'Decision Tables' lists 'A1' and 'A4'. Under 'A1', it shows 'A_1:A' with parameters 'one: String' and 'two: String'. Under 'A4', it shows 'B_1:B' with parameters 'four: String' and 'three: String'. The main area displays two 'New Decision Table' configuration windows. The left window for 'A1' has 'Selected parameters' with 'A_1' checked and 'Any' unchecked. The right window for 'A4' has 'Selected parameters' with 'B_4' checked and 'Any' checked. Below these windows, under 'Inputs:', is a table showing various fact names and their values. Under 'Results:', is a table showing the inferred facts and their values.

Name	Value
B_4	
abc three	
abc four	
B_1	
abc three	
abc four	
A_1	
abc one	abc
abc two	

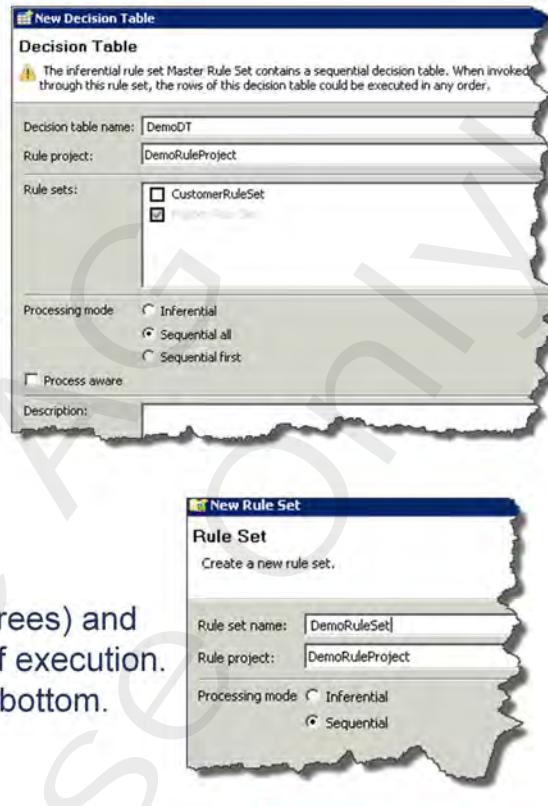
Name	Value
A_1	
abc one	abc
B_1	
abc four	ghi
abc three	def
B_4	

Software AG Training | 3 - 12

Notes:

Sequential Processing Modes

- **Decision Tables & Trees: Sequential all**
Order of rules in a Decision Table or Tree determines the order of execution.
All rules are executed from top to bottom.
- **Decision Tables & Trees : Sequential first**
Order of rules in a Decision Table or Tree determines the order of execution.
Evaluation and execution stops as soon as one rule fires.
- **Rule Sets: Sequential**
Order of Decision Entities (Decision Tables & Trees) and their rules in a Rule Set determines the order of execution.
The Decision Entities are executed from top to bottom.



Software AG Training | 3 - 13

For rule sets, a second processing mode can be specified: sequential processing. Sequential processing means that the order of decision entities in a rule set determines the order of execution. The decision entities are executed from top to bottom.

For decision entities (Tables and Trees), two sequential processing modes exist: sequential all and sequential first. In both cases, the order of rules determines the order of execution, and the rules are executed from top to bottom. But in sequential first processing, the evaluation and execution stops as soon as one rule fires.

Software AG

Sequential Processing Modes - Example

	customer.rating	status
1	0 <...<= 10	= Level 1
2	10 <...<= 20	= Level 2
3	20 <...<= 30	= Level 3
4	30 <...<= 40	= Level 4
5	40 <...<= 50	= Level 5
6	50 <...<= 60	= Level 6
7	60 <...<= 70	= Level 7
8	70 <...<= 80	= Level 8
9	80 <...<= 90	= Level 9
10	90 <...<= 100	= Level 10
11	< 200	= Level 0

	customer.rating	status
1	0 <...<= 10	= Level 1
2	10 <...<= 20	= Level 2
3	20 <...<= 30	= Level 3
4	30 <...<= 40	= Level 4
5	40 <...<= 50	= Level 5
6	50 <...<= 60	= Level 6
7	60 <...<= 70	= Level 7
8	70 <...<= 80	= Level 8
9	80 <...<= 90	= Level 9
10	90 <...<= 100	= Level 10
11	< 200	= Level 0

Software AG Training | 3 < 14

An example decision table demonstrates the difference between sequential all and sequential first processing. In the decision table, range values of customer ratings are evaluated. If a customer with a rating of 65 is evaluated, and the processing mode of this decision table is set to sequential all, all rules will be executed from top to bottom. In the end, two rules will have fired: rule 7 ($60 < x \leq 70$) and rule 11 ($x < 200$). If the processing mode of the decision table is set to sequential first, only rule 7 will fire as the evaluation and execution will stop afterwards.



Exercise 2

- In this exercise, you will add a second Decision Table to your Rule Project. The Decision Table determines a city by the ZIP code. To combine the execution of the Decision Table with your previous one, both will be contained in one common Rule Set.

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



4

Verifications

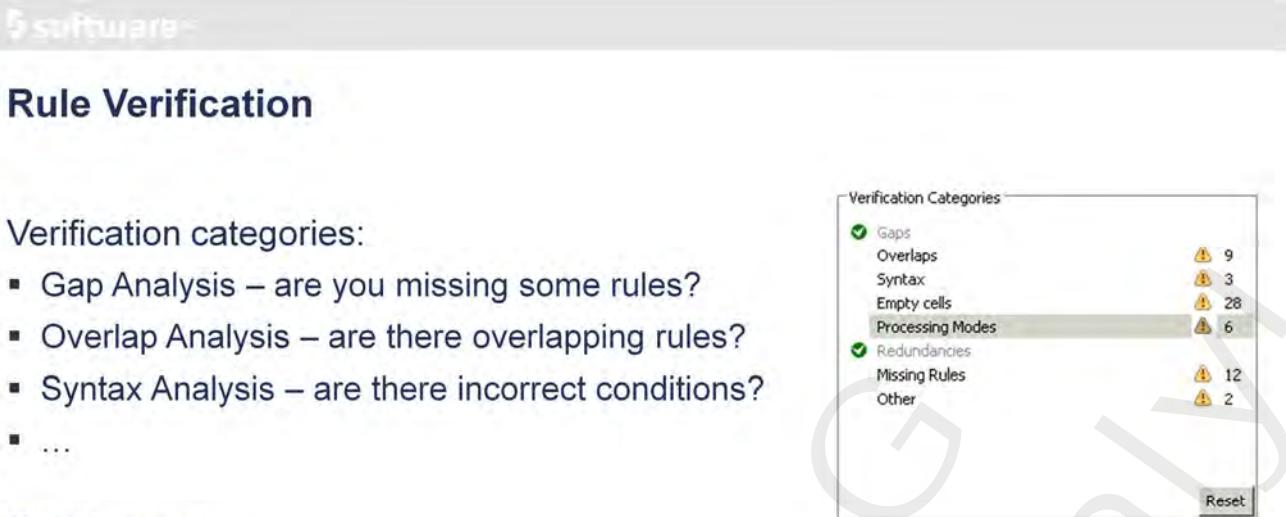
Notes:

Objectives

At the end of this chapter you ...

- Know about the different Verification types in webMethods Business Rules
- Used and customized Rule Verifications in Designer
- Know about Data Verification Services

Notes:



The screenshot shows a 'Verification Categories' panel with the following data:

Category	Count
Gaps	9
Overlaps	3
Syntax	28
Empty cells	6
Processing Modes	12
Redundancies	2
Missing Rules	
Other	

Tool Support:



Rule Developer

Verification support in Designer



Business User

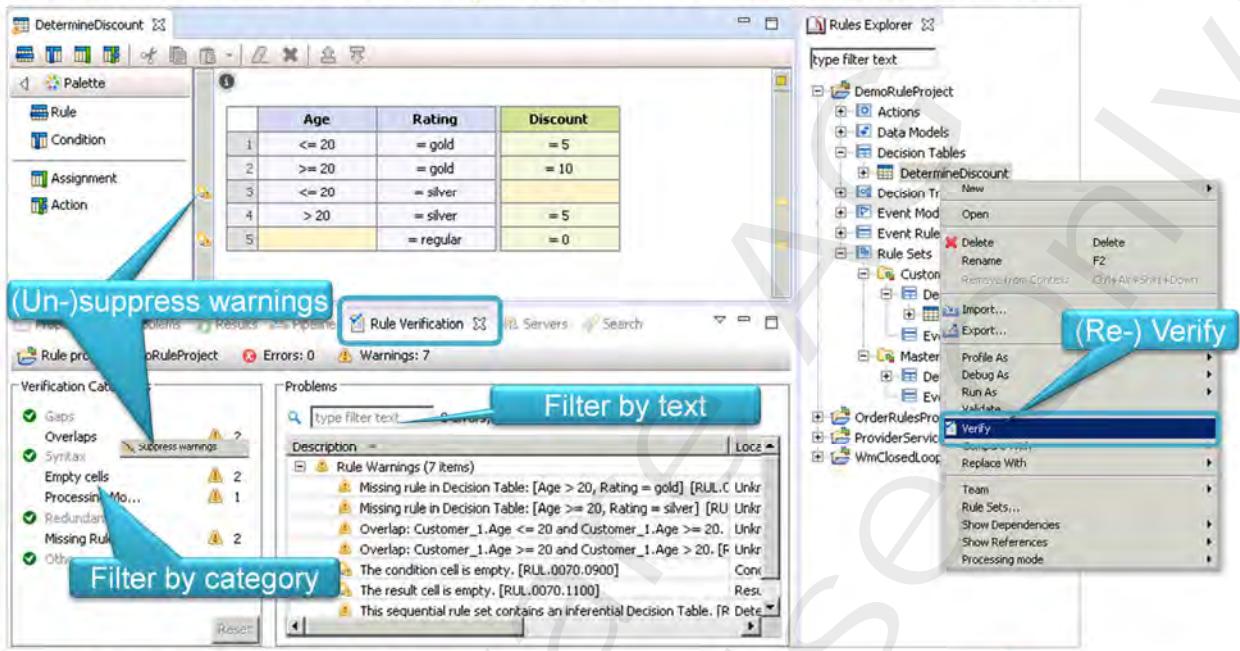
Verification support in Rules Management Console (RMC)

Software AG Training | 4 - 3

Notes:

Rule Verification in Designer

- Optional Verification of Decision Entities* and Rule Sets
 - Verification errors & warnings appear in the Rule Verification view



* Verification feature currently not available for Decision Trees and Event Rules

Software AG Training

8-4

Notes:

Verification Categories and Modes - Overview

- 8 Verification categories:
 - Gaps
 - Syntax
 - Processing Modes
 - Missing Rules
 - Overlaps
 - Empty cells
 - Redundancies
 - Other
- webMethods Rules Development in Designer supports two modes of verification:
 - automatic
 - manual

Notes:

Verification Categories - Gaps

- *Type:* Warning
- *Explanation:* A condition value or a range of condition values is not explicitly tested in a Decision Entity or Rule Set.
- *Example:*

	rating	status	notes
1	< 50	= Standard	= standard customers have a rating below 50
2	> 50	= Silver	= silver customers have a rating above 50

A gap will be reported, because the condition value “= 50” is not tested

We will now take a look at the first verification category: **Gaps**. A gap warning is reported, if a condition value or a range of condition values is not explicitly tested in a Decision Entity (currently Decision Trees do not offer Verifications yet) or a Rule Set.

As an example, I created the Decision Table CustomerStatusByRating that assigns a status to a customer based on a rating. I defined one rule for a rating lower than 50 and one rule for a rating higher than 50. If I verify the Decision Table, a gap warning will be reported, because a rating that equals 50 is not explicitly tested.

Verification Categories - Overlaps

- *Type:* Warning
- *Explanation:* A condition value or a range of condition values is tested multiple times in a Decision Entity or Rule Set
- *Example:*

	rating	status	notes
1	<= 50	= Standard	= standard customers have a rating below 50
2	>= 50	= Silver	= silver customers have a rating above 50

An overlap will be reported, because the condition value “= 50” is tested twice

We will now take a look at the second verification category: **Overlaps**. An overlap warning is reported, if a condition value or a range of condition values is tested multiple times in a Decision Entity or a Rule Set.

I modified our Decision Table and set the first rating value to “lower than or equal to 50” and the second to “higher than or equal to 50”. If I verify the Decision Table, an overlap warning will be reported, because the condition value “equal to 50” is tested twice.

Verification Categories - Gaps and Overlaps in Rule Sets

- CustomerStatusByRating:

	rating	status	notes
1	<= 50	= Standard	= standard customers have a rating below 50
2	> 50	= Silver	= silver customers have a rating above 50

- SeniorVIPCustomers:

	rating	age	status
1	> 80	> 50	= VIP

The rating condition values “> 50” and “> 80” lead to an overlap warning

Software AG Training | 4 - 8

If you verify on Rule Project or Rule Set level, gaps and overlaps can be exposed, even if the individual Decision Tables do not contain any.

An example will demonstrate this: I created a second Decision Table, SeniorVIPCustomers, that assigns a status to a customer based on his or her rating and age. I defined a rule that assigns VIP status to a customer with a rating higher than 80 and an age higher than 50. I grouped both Decision Tables in a Rule Set called CustomerStatusRules.

If I verify this Rule Set, the rating “higher than 50” of the Decision Table CustomerStatusByRating and the rating “higher than 80” of the Decision Table SeniorVIPCustomers will lead to an overlap warning.

Software AG Training | 4 - 9

Verification Category - Syntax

- **Type:** Warning and Error
- **Explanation:** Syntax errors and warnings are of different kinds. Warnings are reported, if data is lost due to truncation.
Errors are reported, if only operators are specified.
- **Example:**

	rating	status	notes
1	=	= Standard	= standard customers have a rating below 50
2	> 50	= Silver	= silver customers have a rating above 50

A syntax error will be reported, because only an operator is defined for the first rating value

The third verification category is called **Syntax**. There are syntax warnings and errors of different kinds. A syntax error will be reported for example, if a condition or result value is only specified by an operator.

To demonstrate this, I modified our Decision Table, deleted the rating value in the first row, specified a new operator, but forgot to assign a literal value, parameter element, action or constant. Verifying the Decision Table will expose a syntax error. The Decision Table cannot be exported, deployed and executed.

Syntax warnings are for instance reported, if data is lost due to truncation. This may happen, if you assign a value of the data type long to a condition or result of the data type short.

Verification Category - Empty Cells

- *Type:* Warning
- *Explanation:* An empty cell warning is reported, if a condition value or result value is not specified
- *Example:*

	rating	status	notes
1		= Standard	= standard customers have a rating below 50
2	> 50	= Silver	= silver customers have a rating above 50

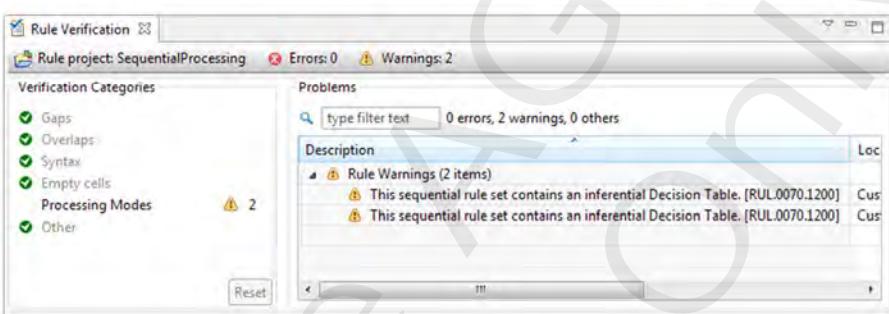
An empty cell warning will be reported, because the first rating value is not specified

The fourth verification category are **Empty cells**. An empty cell warning is reported, if a condition value or result value is not specified.

I modified our Decision Table and also deleted the operator in the first rating cell. If I verify the Decision Table again, an empty cell is detected and reported.

Verification Category - Processing Modes

- **Type:** Warning
- **Explanation:** A Processing Mode warning is reported if the Processing Mode of a Decision Table within a Rule Set differs from the Processing Mode of this Rule Set
- **Example:**



Differences in Processing Modes can occur if you add an inferential Decision Table to a sequential Rule Set or vice versa; or if you modify the Processing Mode of a Rule Set or of a Decision Table within this Rule Set

Software AG Training | 4 - 11

Fifth verification category: **Processing Modes.** A Processing Mode warning is reported if the Processing Mode of a Decision Table within a Rule Set differs from the Processing Mode of this Rule Set, because the Processing Mode of the Rule Set overwrites that of the Decision Table.

Differences in Processing Modes can occur if you add an inferential Decision Table to a sequential Rule Set or vice versa; or if you modify the Processing Mode of a Rule Set or of a Decision Table within this Rule Set.

Verification Category - Redundancies

- *Type:* Warning
- *Explanation:* A redundancy warning is reported if parts of one rule, or rules of one Decision Table, or rules of several Decision Tables within one Rule Set are dispensable
- *Example:*

	order.value	country	discount
1	> 500		= 4
2	> 500	= Germany	= 4

A redundancy warning is reported, because as in the first rule no value is specified for the condition *country*, any value applies to this rule. This makes the second rule superfluous.

Sixth category is called **Redundancies**. A redundancy warning is reported if parts of one rule, or rules of one Decision Table, or rules of several Decision Tables within one Rule Set are dispensable.

To demonstrate this, I created a Decision Table that assigns a *discount* to a customer based on *order value* and *country*. I defined one rule for an *order value* “> 500”. For this rule I did not specify a *country*. I defined another rule for an *order value* “> 500” and “Germany” as *country*. The assigned *discount* in both rules is “4”.

Verifying the Decision Table will expose a redundancy warning. As in the first rule no value is specified for the condition *country*, any value applies to this rule. This makes the second rule superfluous.

Verification Category: Missing Rules

- *Type:* Warning
- *Explanation:* A missing rule warning is reported if a probable combination of conditions is not explicitly tested in a Decision Entity or a Rule Set
- *Example:*

	gender	olderThan45
1	= male	= true
2	= male	= false
3	= female	= true

A missing rule warning is reported, because the condition combination *gender*=“female” AND *olderThan45*=“false” is not explicitly tested

Verification category no.7 is **Missing Rules**. A missing rule warning is reported if a probable combination of conditions is not explicitly tested in a Decision Entity or a Rule Set.

To demonstrate this, I created a second Decision Table that uses *gender* and *olderThan45* as conditions. I defined one rule for *gender* is “male” and *olderThan45* is “true”. I defined a second rule for *gender* is “male” and *olderThan45* is “false”. For the third condition I specified “female” as *gender* and *olderThan45* is “true”.

Verifying the Decision Table will expose a missing rule warning, because the condition combination *gender*=“female” AND *olderThan45*=“false” is not explicitly tested.

Verification Category: Other

- *Type:* Warning and Error
- *Explanation:* All warnings and errors that do not fit into the other categories

The eighth and last verification category is called **Other**. The Other category includes all warnings and errors that do not fit into the other categories.

Automatic Verification...

- Automatic Verification is performed, when you save or modify a Decision Entity. It can reflect both errors and warnings
- Warnings are highlighted in the editor by yellow indicators, Errors are highlighted in the editor by red indicators



- Decision Entities with warnings or errors are additionally marked in the Rules Explorer view

webMethods Rules Development supports two kinds of verification: automatic and manual verification. Both verification types will be explained in the presentation.

All verification errors and warnings appear in a specifically created view, the Rule Verification view. This view will be covered in detail.

Warnings can be suppressed and restored. It will be demonstrated, how to suppress warnings and make them visible again.

Software AG Designer preferences now contain rule verification preferences.

Automatic verification is performed, when you save or modify a Decision Entity. It can reflect both errors and warnings.

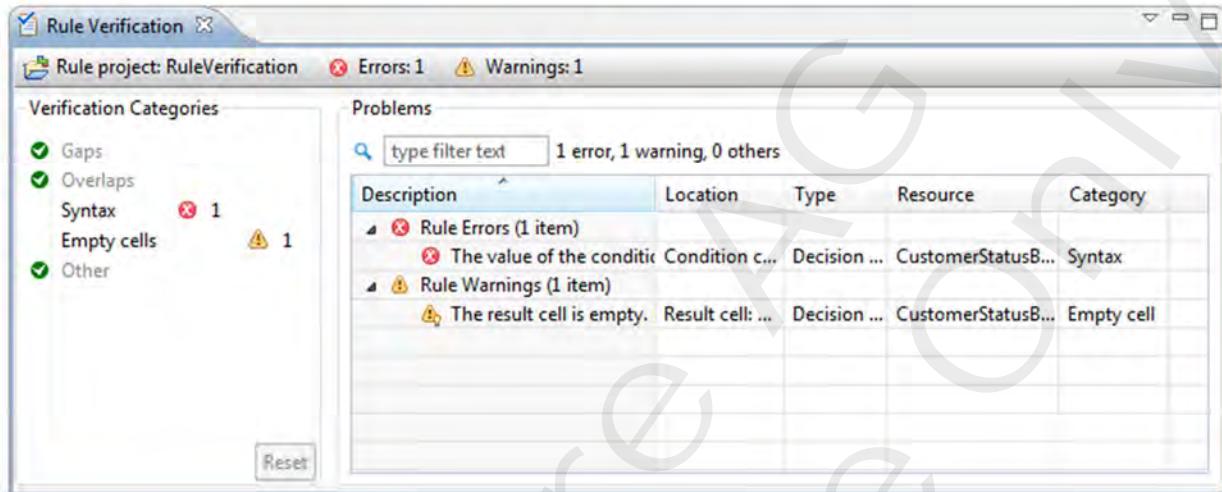
The warnings and errors are highlighted in the editor: The background of a cell with an error is highlighted in red; the background of a cell with a warning is highlighted in yellow. The left and right rulers of the editor are marked by specific red or yellow icons. If you click on an icon in the right ruler, the focus of the editor jumps to the according rule.

On saving, a Decision Entity with warnings or errors is marked by a yellow or red icon in the Rules Explorer view.



...Automatic Verification

- Detected warnings and errors are also shown in the Rule Verification view in the verification categories Syntax, Empty cells and Other.



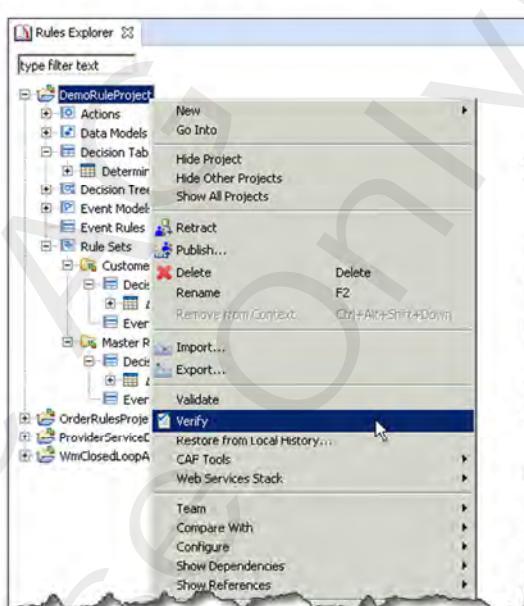
Software AG Training | 4 - 16

The detected warnings and errors are also shown in the Rule Verification view within the verification categories **Syntax, Empty cells or Other**.

To open this view, click Window > Show View > Other..., and select Software AG > Rule Verification from the list in the Show View pop-up window.

Manual Verification...

- Manual verification is performed on-demand on Rule Project, Rule Set or Decision Entity level. It is designed to detect potential logic problems in decision entities and only creates warnings
- Rule Projects, Rule Sets or Decision Entities are manually verified using the **Verify** item from the Rules Explorer context menu
- Rule Projects, Rule Sets or Decision Entities with warnings are marked in the Rules Explorer view



Software AG Training | 4 - 17

Manual verification is performed on-demand on Rule Project, Rule Set or Decision Entity level. It is designed to detect potential logic problems in decision entities and only creates warnings. There are certain points to keep in mind when verifying rules manually. These are described in detail in the webMethods Rules Development Help.

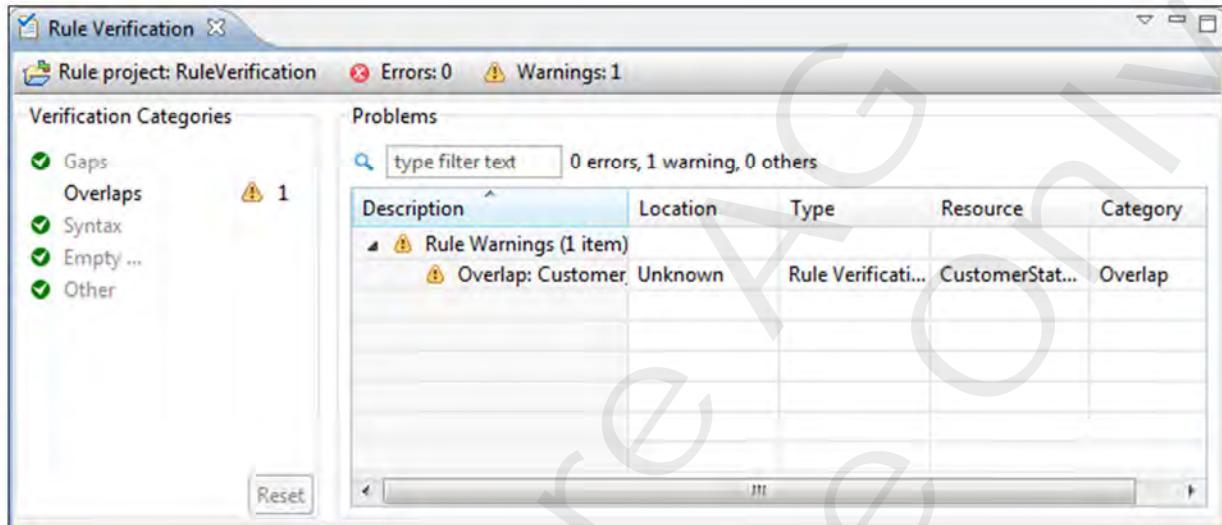
To manually verify a Rule Project, Rule Set or Decision Entity, right-click the respective asset in the Rules Explorer view and select Verify from the context menu. Note that the context menu item Validate is an Eclipse-specific item that is not related to rule verification.

After manual verification, a Rule Project, Rule Set or Decision Entity with warnings is marked by a yellow icon in the Rules Explorer view.



...Manual Verification

- Detected warnings are also shown in the Rule Verification view in all verification categories



Software AG Training | 4 - 18

Again, the detected warnings are also shown in the Rule Verification view within all verification categories.

Additional Rule Verification View Features

- You can filter for verification categories
- Warnings can be suppressed and restored
- The entries in the Problems column can be filtered
- The view features a pull-down menu to modify the view's settings

The screenshot shows the 'Rule Verification' view window. At the top, there's a toolbar with a 'Rule Verification' button, a 'Rule project: RuleVerification' dropdown, and a 'Warnings: 1' indicator. Below the toolbar is a 'Verification Categories' section with checkboxes for 'Gaps', 'Overlaps', 'Syntax', 'Empty cells', and 'Other'. To the right of this is a 'Items' table with columns: Description, Location, Type, Resource, and Category. The table contains two rows: one for 'Rule Errors' (1 item) and one for 'Rule Warnings' (1 item). A 'Filter by text' input field is above the table, and a 'Filter by category' button is below it. A 'Reset' button is at the bottom left. A 'Supress warnings' context menu is open over the 'Empty cells' category in the categories list. Callouts highlight the 'Filter by category' button, the 'Supress warnings' option, the 'Filter by text' input field, and the 'Items' table.

Description	Location	Type	Resource	Category
Rule Errors (1 item) The value of the condition c...	Decision ...	CustomerStatusB...	Syntax	
Rule Warnings (1 item) The result cell is empty. Result cell: ...	Decision ...	CustomerStatusB...	Empty cell	

Software AG Training | 4 - 19

There are some more features of the Rule Verification view that I would like to point out to you:

You can select a verification category on the left-most column labeled as Verification Categories to only show the associated warnings and errors in the Problems table on the right side.

Warnings can be suppressed and restored. Right-click the warning icon to do so.

The entries in the Problems table can be filtered. Provide a filter value in the corresponding text input field.

The Rule Verification view has a pull-down menu that allows you to specify the view's settings.

Software AG

Warning Suppression

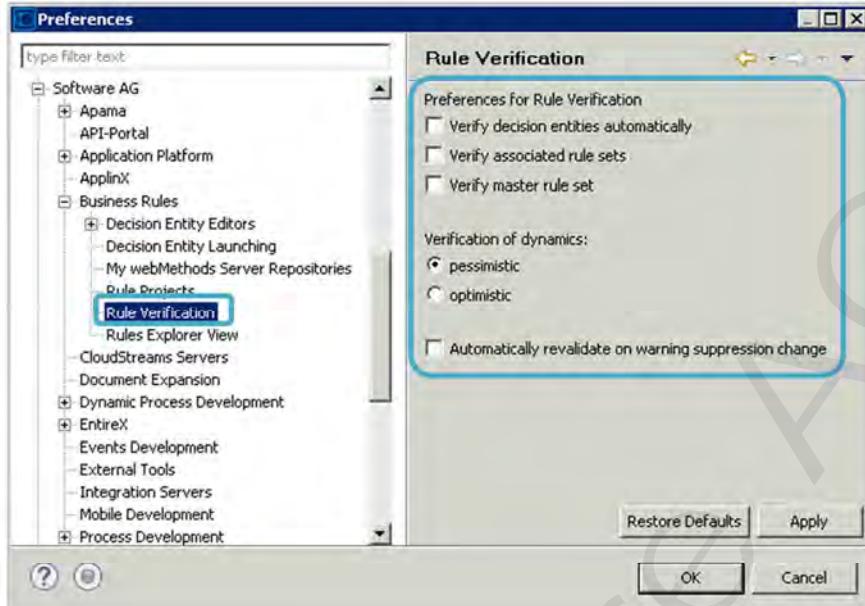
- Warnings can also be suppressed and restored in the Decision Entity editor

The screenshot shows a decision table with two rows. The first row has a 'rating' column containing ' ≤ 50 ' and a 'notes' column containing '= standard customers have a rating below 50'. A tooltip is displayed over the second row's 'status' column, which says 'Suppress warning 'The result cell is empty.' for [column: 2, row: 1]'. The 'notes' column for the second row is empty.

- The settings in the Rule Verification view overwrite the settings in the Decision Entity editor

Software AG Training | 4 - 20

As we've already learnt, warnings can be suppressed and restored with the help of the Rule Verification view. But you can also suppress and restore warnings in the Decision Entity editor. Note that the settings in the Rule Verification view overwrite the settings in the editor.



The screenshot shows the 'Preferences' dialog box for Software AG Designer. The left pane lists various categories under 'Software AG'. The 'Business Rules' category is expanded, and 'Rule Verification' is selected and highlighted with a blue border. The right pane displays the 'Rule Verification' settings. It includes sections for 'Preferences for Rule Verification' (checkboxes for 'Verify decision entities automatically', 'Verify associated rule sets', and 'Verify master rule set'), 'Verification of dynamics:' (radio buttons for 'pessimistic' and 'optimistic' with 'pessimistic' selected), and a checkbox for 'Automatically revalidate on warning suppression change'. Buttons at the bottom include 'Restore Defaults', 'Apply', 'OK', and 'Cancel'.

- Rule Verification settings can be modified in Software AG Designer preferences
- Perform Verification on demand or automatically
- Based on single Decision Entities or include Rule Sets
- Including/excluding master Rule Set

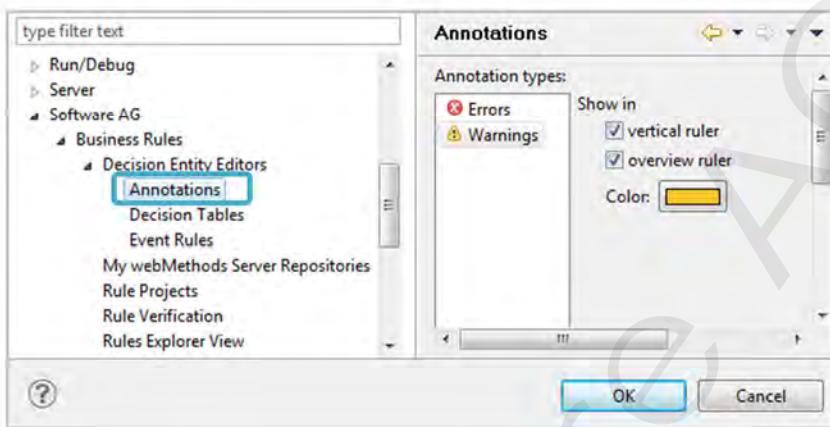
Software AG Training | 4 - 21

Verification of dynamics:

Pessimistic verification (default) assumes that the return values from actions or functions, or the input values from parameters will possibly lead to warnings and errors. Setting this preference increases the amount of verification warnings and errors but ensures that problems that may occur at runtime are detected when verifying rules. Optimistic verification assumes that the return values from actions or functions, or the input values from parameters will possibly not lead to warnings and errors. Setting this preference reduces the amount of verification warnings and errors but can lead to undetected problems that may occur at runtime.

...Rule Verification Properties in Designer

- You can modify the setting for the editor rulers, the highlighting colors of errors and warnings and general verification settings



Software AG Training | 4 - 22

Software AG Designer's preferences were extended to enable modification of rule verification settings. To open the preferences, select Preferences > Software AG > Business Rules.

You can modify the setting for the rulers, the highlighting colors of errors and warnings and general verification settings.

More Verifications

- Verifications also supported in Rules Management Console (see chapter 10)
 - Business user can verify their instant rule changes before saving
- RMC additionally offers invocation of a Business Verification service for Decision Table columns
 - Backend REST service has to be configured in Designer and RMC
 - Service checks that values in a decision tables column contain valid data only

The screenshot shows two windows. The top window is 'Decision Entity Editor - CustomerStatus' displaying a decision table with columns 'age', 'zipcode', 'status', and 'In Effect'. The bottom window is 'BusinessVerification - CustomerStatus - Verification' showing validation results. A red box highlights a validation error: 'Invalid zip code. Only zip codes starting with D4 are allowed.' This error is associated with 'CustomerStatus.decisiontable_1' and 'Condition cell: column 2, row 3'.

Software AG Training | 4 - 23

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



5

Internal Event Rules

Notes:

Objectives

At the end of this chapter you ...

- Can explain the usage of Internal Event Rules
- Created internal Event Rules using Designer
- Tested internal Event Rules in Designer

Notes:

Using Business Rules in the Digital Business Platform

The diagram illustrates four methods for integrating business rules into the Digital Business Platform:

- From a Business Process:** Shows a BPMN diagram with a "Rule Task Activity" step.
- From an Event source (BPM or other):** Shows an event definition for "OrderReceivedEM" with results for "CreateOrderReceivedDocument()" and "writeToLog()".
- For a User Task Assignment:** Shows a decision table mapping "Discount" to "HandledBy" for various percentages (20%, 10%, 15%, 5%) and team names (VIPTeam, StandardTeam).
- From Integration Server:** Shows a service structure for "WmBusinessRules" including "pubs", "businessrules", "client", "genericinvoke", "invoke", "invSelInput", and "invokeOutput".

DIGITAL BUSINESS PLATFORM

Software AG Training | 5 of 3

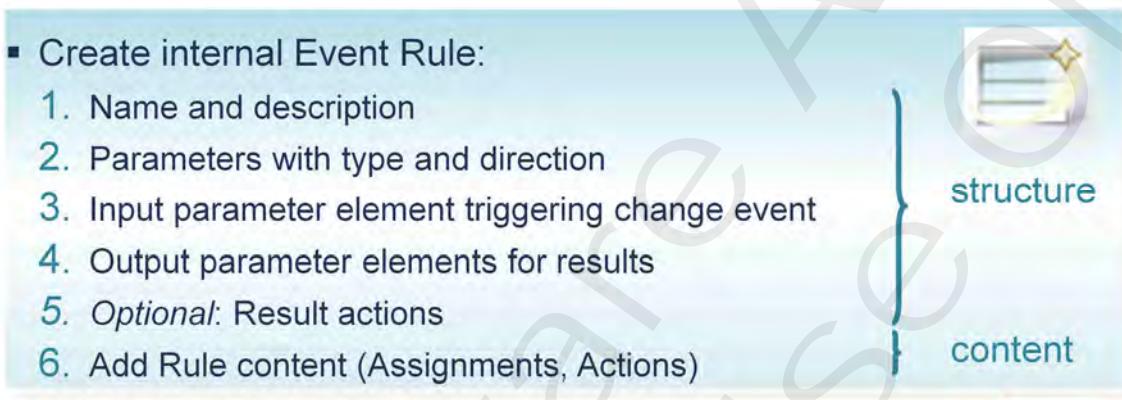
Notes:



Internal Event Rules

- Event Rules are Decision Entities
 - Belong to a Rule Project
 - May belong to a user-defined Rule Set (optional)
 - Can be internal or external*
- Internal Event Rules return result(s) if a parameter element is changed
- Create internal Event Rule:
 1. Name and description
 2. Parameters with type and direction
 3. Input parameter element triggering change event
 4. Output parameter elements for results
 5. *Optional:* Result actions
 6. Add Rule content (Assignments, Actions)

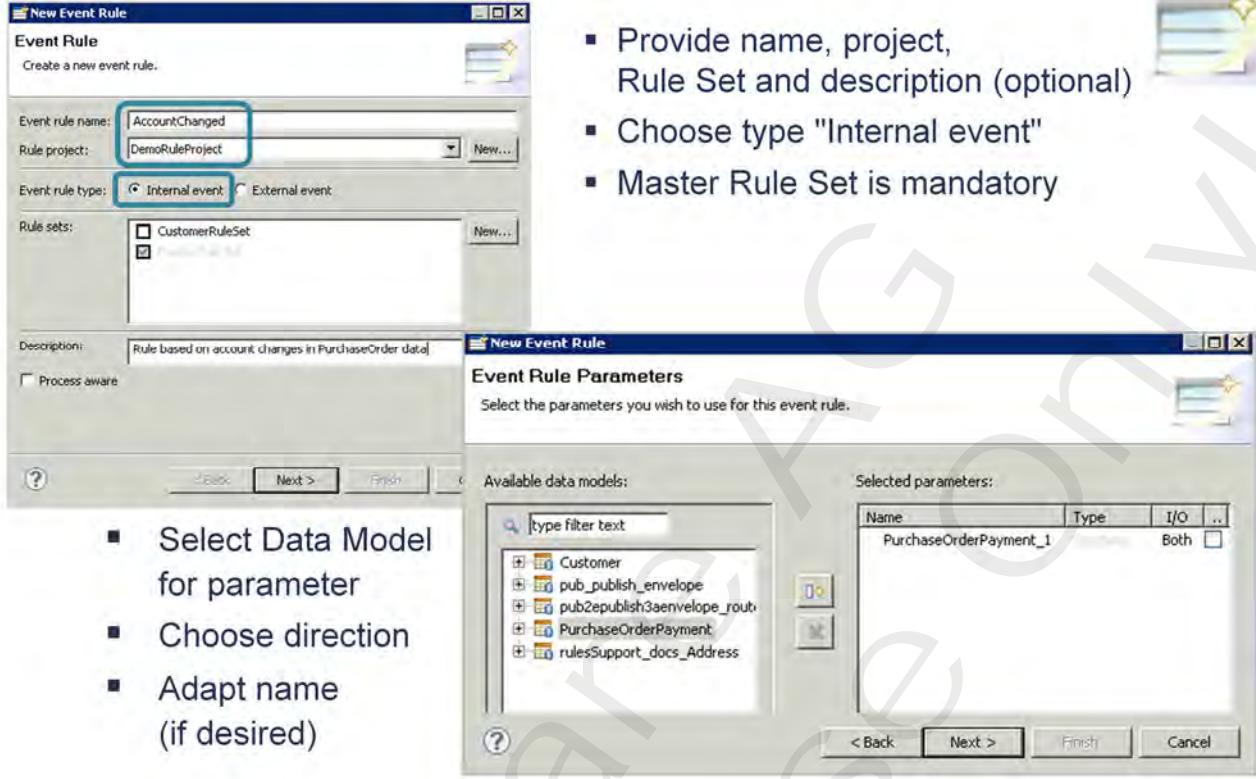
* external Event Rules will be handled in a later topic



Software AG Training | 5 - 4

Notes:

1-5: Create Internal Event Rule Structure...



The screenshot shows the "New Event Rule" wizard with two windows open:

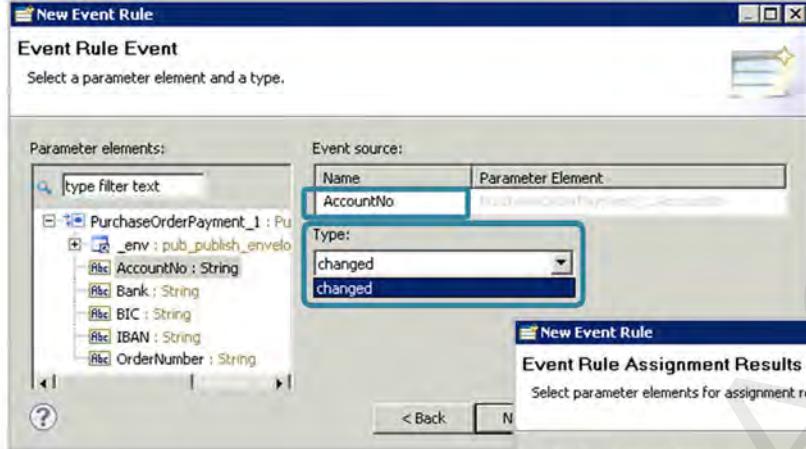
- Event Rule (Step 1):** Shows fields for "Event rule name" (AccountChanged), "Rule project" (DemoRuleProject), and "Event rule type" (Internal event). A callout points to the "Event rule name" field.
- Event Rule Parameters (Step 2):** Shows "Available data models" (Customer, pub_publish_envelope, pub2publish3aenvelope_rout, PurchaseOrderPayment, rulesSupport_docs_Address) and a selected parameter "PurchaseOrderPayment_1" with direction "Both".

List of steps:

- Provide name, project, Rule Set and description (optional)
- Choose type "Internal event"
- Master Rule Set is mandatory
- Select Data Model for parameter
- Choose direction
- Adapt name (if desired)

Notes:

...1-5: Create Internal Event Rule Structure



The screenshot shows the "New Event Rule" wizard. Step 1: "Event Rule Event". It displays a list of "Parameter elements" (PurchaseOrderPayment_1, _env) and an "Event source" table with "Name" (AccountNo) and "Parameter Element" (AccountNo). A dropdown "Type" is set to "changed". Step 2: "Event Rule Assignment Results". It shows "Available parameter elements" (PurchaseOrderPayment_1, _env) and a "Selected parameter elements" table with "Label" (IBAN, BIC).

- Select one Parameter Element for "is changed" trigger
- Rename Name, if desired

- Select Parameter Elements used as Rule results*
- Rename label, if desired

*Additionally, Actions to be executed as a result could be defined here, too (see later topic)

Notes:

6. Adding Content to Event Rule – Event Rule Editor

Add new Event Rule Assignment to the end of the Results list

Add a new Action to the end

Clear selected elements

Delete selected elements

Collapse/Show/
Edit description

Cut, copy
and paste

Description
Checks for a change in the account number. If changed, IBAN and BIC will be reset.

Event
AccountNo changed

Results
1 IBAN Operator...
2 BIC Operator...

ONE rule

Palette
Assignment
Action

Assignment – Add a new Assignment

Action – Add a new Action

Event Rule Editor

Palette

- Use appropriate icons in Event Rule Editor or use icons from Palette to customize an Event Rule:
 - Add/modify Rule
 - Add/remove/rename results (Assignments, Actions)

Software AG Training | 5 - 7

Notes:

Save and Test Event Rules in Designer

- Saved Event Rules can be tested standalone in Designer – no IS required
- Possibilities:
 1. Run As -> Run Event Rule
 - (Will prompt you for a Decision Table to trigger your Event Rule)
 2. Add Event Rule to a Rule Set containing triggering Decision Table and run Rule Set

The screenshot shows the 'Rules Explorer' window with a project tree. A right-click context menu is open over a 'CustomerRuleSet' node, with 'Run As' highlighted. Above the main window, a modal dialog titled 'Launch AccountChanged' asks 'Choose a decision entity'. It lists 'DemoRuleProject' with 'Decision Tables' expanded, showing 'DetermineDiscount'.

Notes:



Exercise 3

- In this exercise, you will add an Event Rule to your Rule Project. The Event Rule will be fired whenever a city of the billing address is changed. In this case the Event Rule copies the changed city value to the shipping address. To enforce inference, the Event Rule belongs to your common customer Rule Set.

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



6

Decision Trees

Notes:

Objectives

At the end of this chapter you ...

- Created a Decision Entity of type Decision Tree
- Tested your Decision Tree in Designer
- Used Run Configurations
- Can explain the differences between a Decision Table and a Decision Tree

Notes:

Decision Tables vs. Decision Trees

Decision Table:

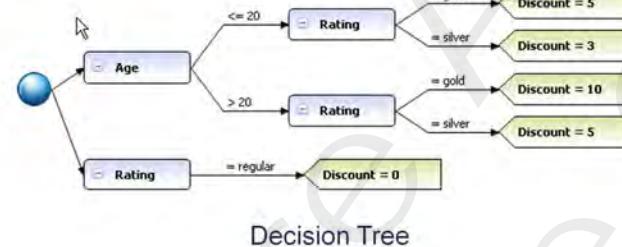
- Tabular decision entity
- Very structured
- Ensuring completion and that you have analyzed every possible combination of conditions
- Compact method to view all of the available decisions at once
- Can deal with large amount of data

	Age	Rating	Discount
1	<= 20	= gold	= 5
2	> 20	= gold	= 10
3	<= 20	= silver	= 3
4	> 20	= silver	= 5
5		= regular	= 0

Decision Table

Decision Tree*:

- Graphical decision entity
- Easy to create
- Easy to read and explain
- Can easily deal with missing data: Conditions that do not matter are absent
- Quickly become very large and hard to read



Decision Tree

* Currently, Decision Trees do not offer:
Verifications in Designer and RMC, In-effect Dates in Designer, Rule Verification Services, Data Provider Services , Principal Picker in RMC

Notes:

Software AG

Create a Decision Entity of Type Decision Tree

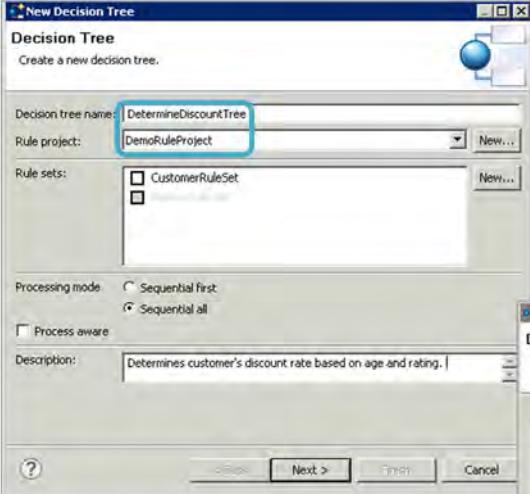
1. Create/identify Rule project to use
2. Create/identify Data Model(s) to use
3. Create Decision Entity of type Decision Tree
 - a. Name and description
 - b. Parameters with type and direction
 - c. Top level input parameter element for first condition node
 - d. Add and configure additional condition nodes
 - e. Add and configure resulting Assignment nodes
 - f. Optional: Add and configure resulting Action nodes
 - g. Copy, paste and configure additional Rules as sub-trees
4. Save and test in Designer

The diagram illustrates the structure of a Decision Tree. It shows a top-level window icon labeled 'initial structure'. Below it, a blue circle represents a condition node, with several white rectangles representing assignment nodes branching from it. A large curly brace on the right side groups these two elements under the heading 'structure & content'.

Software AG Training | 6 - 4

Notes:

3.a-b: Create Initial Decision Tree Structure



- Provide name, Rule Set, Processing mode* and (optional) description
- Master Rule Set is mandatory

Decision Tree Parameters

Select the parameters you wish to use for this decision tree.

Available data models:

- Customer
- docTypes3aPerson_Capabilities
- docTypes3aPerson_Privileges
- Person
- pub_publish_envelope
- pub2epublish3aenvelope_route
- PurchaseOrderPayment

Selected parameters:

Name	Type	I/O	Any
Customer_1	Customer	Both	<input checked="" type="checkbox"/> Input <input checked="" type="checkbox"/> Output <input checked="" type="checkbox"/> Both

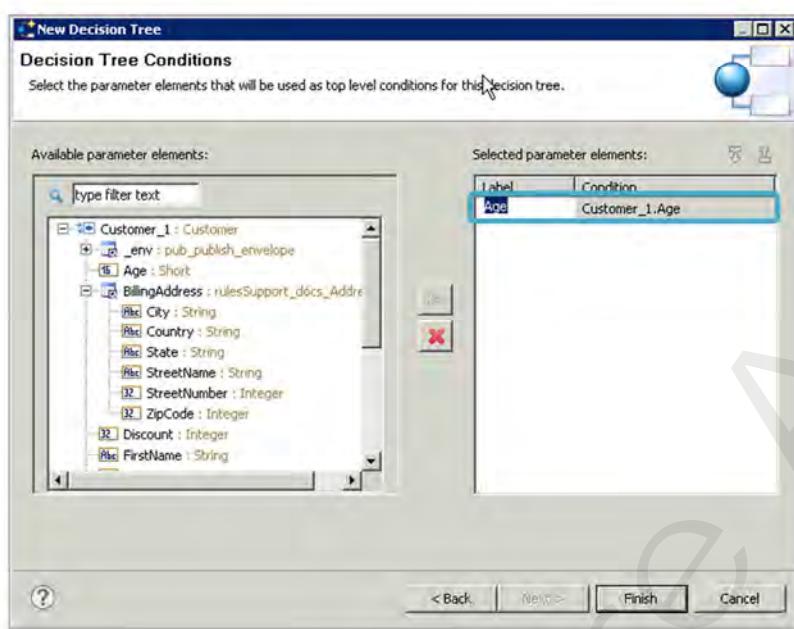
Named Instance of a Data Model with I/O direction (JESS: Fact)

*Inferential rule processing not available for Decision Trees

Software AG Training | 6 - 5

Notes:

3.c: Create Initial Decision Tree Structure

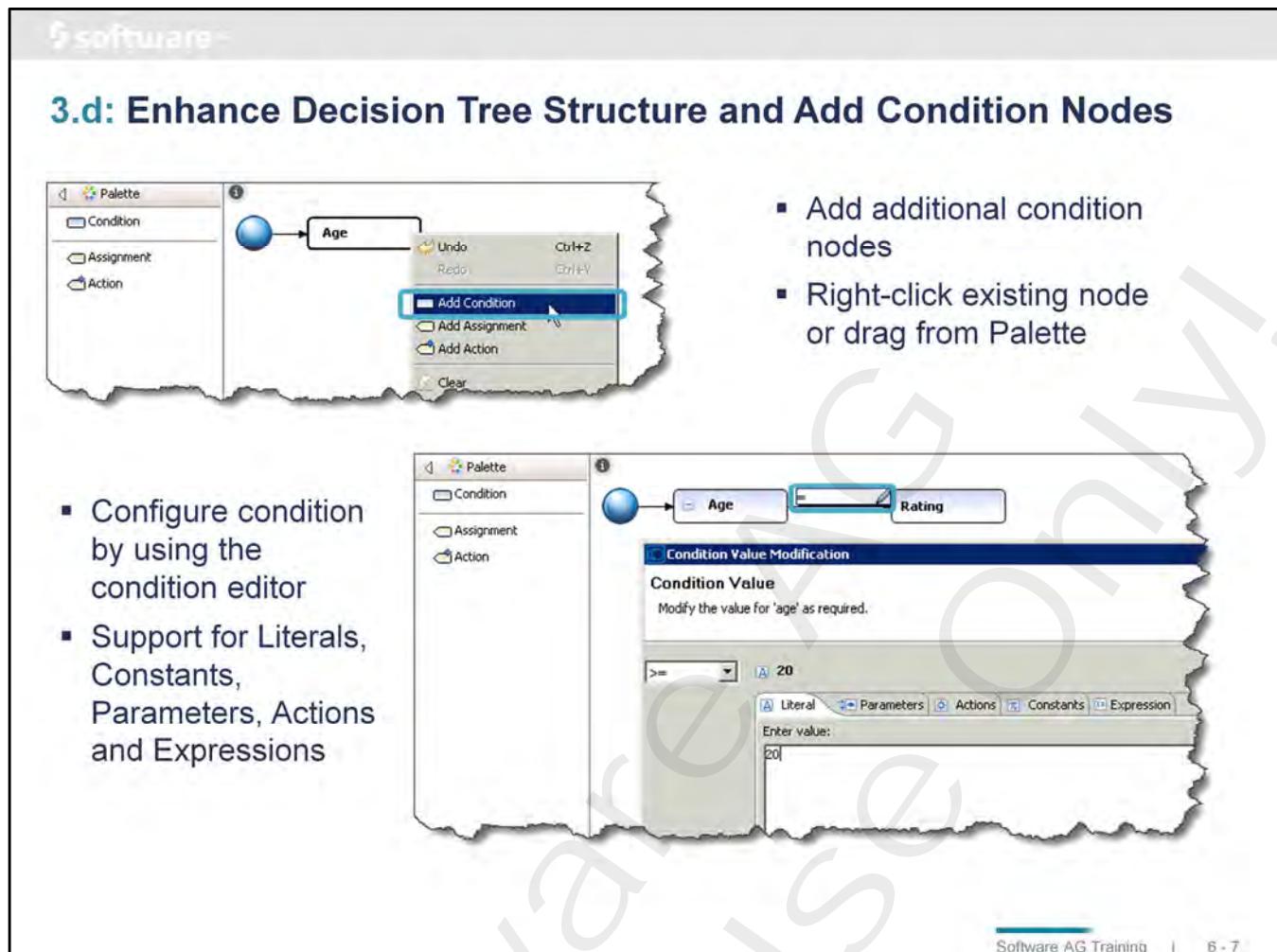


The screenshot shows the 'New Decision Tree' dialog box. In the 'Available parameter elements' tree view, under 'Customer_1 : Customer', there is a node 'Age : Short'. In the 'Selected parameter elements' table, there is one row: 'Age' with 'Condition' 'Customer_1.Age'. The dialog has buttons for '?', '< Back / Next >', 'Finish', and 'Cancel'.

- Select Parameter Element(s) used as top level conditions
- Rename label, if desired

Software AG Training | B - B

Notes:



The screenshot shows the webMethods Business Rules interface. On the left is a palette with categories: Condition, Assignment, and Action. A decision tree node labeled 'Age' is selected. A context menu is open over the node, with the 'Add Condition' option highlighted. Below the palette, a decision tree diagram shows a start node connected to an 'Age' node, which then connects to a 'Rating' node.

3.d: Enhance Decision Tree Structure and Add Condition Nodes

- Add additional condition nodes
- Right-click existing node or drag from Palette

Condition Value Modification

Condition Value
Modify the value for 'age' as required.

\geq 20

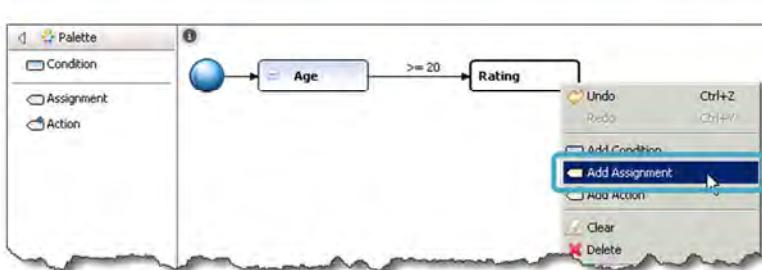
Literal Parameters Actions Constants Expression

Enter value:
20

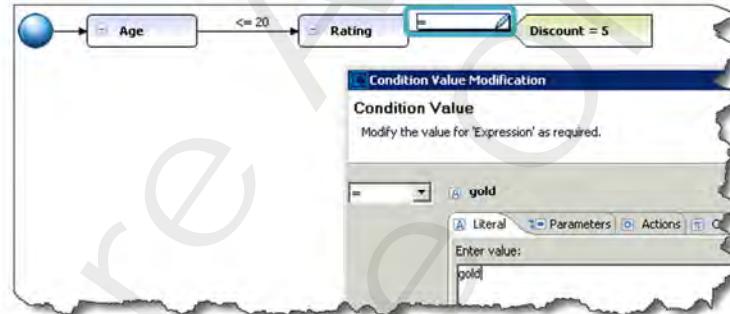
Software AG Training | 6 - 7

Notes:

3.e: Enhance Decision Tree Structure and Add Assignment Nodes



- Add Assignment nodes*
- Right-click existing node or drag from Palette



- Configure condition by using the condition editor
- Support for Literals, Constants, Parameters, Actions and Expressions

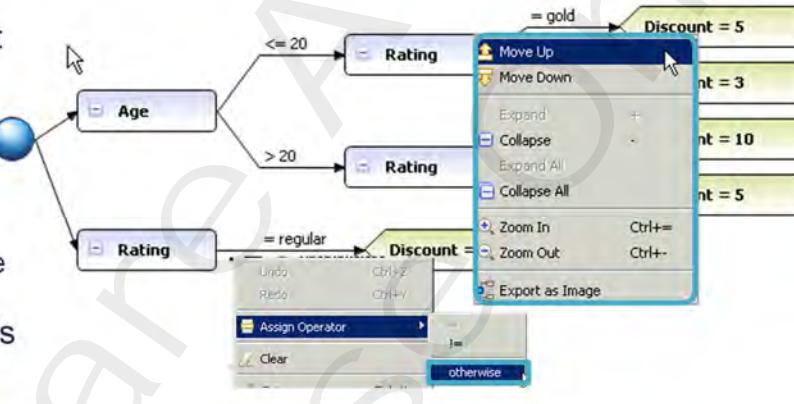
*Additionally, Actions to be executed as a result could be added as nodes here, too (see later topic)

Notes:

3.g: Copy, Paste and Configure Additional Rules as Sub-trees



- To create additional rules, copy, paste & configure existing nodes
- All descendant nodes will be copied, too



- Node context menu allows to:
 - Collapse and expand nodes
 - Reorder the sequence of nodes
 - Zoom in/out
 - Export Decision Tree as image
- Consider using "Otherwise" as condition for last tree branch

Software AG Training | 6 - 9

Notes:

Decision Tree in the Decision Tree Editor

Add a Condition to a node Add an Assignment to a node Add an Action to a node Cut, copy & paste node Clear selected element Delete selected element

Add a Condition to a node

Add an Assignment to a node

Add an Action to a node

Collapse/show description

Change node sequence

Palette Decision Tree Editor

- Use appropriate icons in Decision Tree Editor panel or use icons from Palette to customize Decision Tree:
 - Add/modify Rules (trees)
 - Cut, copy and paste Rules (nodes with descendant nodes)
 - Add/remove/rename nodes (Conditions, Assignments, Actions)
 - Change node sequence

Software AG Training | 6 - 10

Notes:

Handling Large Decision Trees in Designer

The screenshot shows the SAP BusinessObjects Decision Tree Designer interface. A large decision tree is displayed with various nodes and conditions. Annotations highlight specific features:

- Zoom in/out**: Points to the zoom controls in the top toolbar.
- Expand/collapse node incl. descendant nodes**: Points to a node in the tree and its associated expand/collapse icon. This feature is also shown in the Outline view below.
- Outline view**: Shows a detailed tree structure with many nodes and connections, illustrating the 'bird-eye view' of the entire Decision Tree.

Outline view features:

- Outline view additionally offers a bird-eye view of the entire Decision Tree.
- Blue box represents and influences the visible portion in the Decision Tree editor.

Notes:

Handling Large Decision Trees in Designer

▪ Use collapse and expand functionality to hide/show Decision Tree nodes

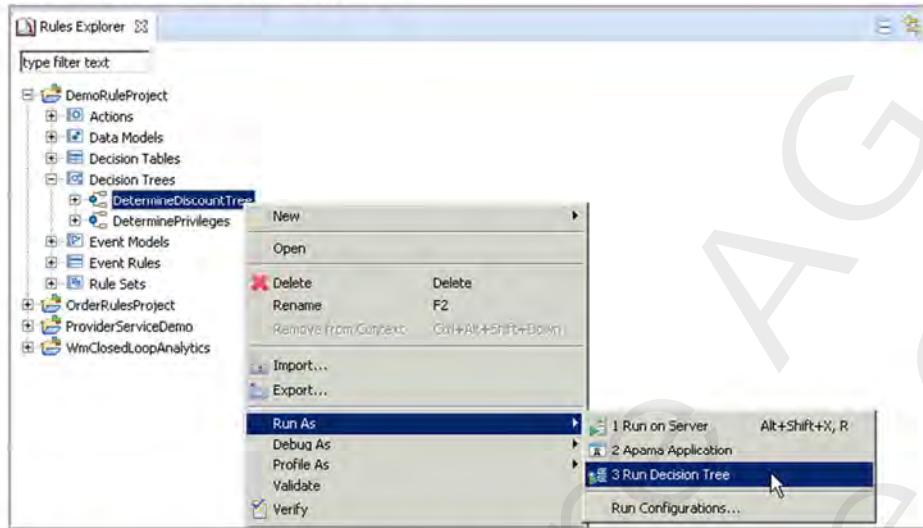
- Outline view additionally offers a bird-eye view of the entire Decision Tree

Software AG Training | 6 - 12

Notes:

4. Save and Test Decision Tree in Designer...

- Saved Decision Trees can be tested standalone in Designer – no IS connection required



- Prompts for Save if Decision Entity is "dirty"

Notes:

...4. Save and Test Decision Tree in Designer

- Provide inputs on appearing panel
 - Check "Include empty values for String Types", if necessary

The screenshot shows a Windows application window titled 'DetermineDiscountTree'. The main title bar says 'Enter Input for 'DetermineDiscountTree''. Below the title is a checked checkbox labeled 'Include empty values for String Types'. The main area is a table with two columns: 'Name' and 'Value'. The 'Name' column contains several entries with checkboxes to their left, some of which are checked. The 'Value' column contains the corresponding data. The table rows are as follows:

Name	Value
Customer_1	
FirstName	Joltawan
LastName	Barodscheff
Age	57
Rating	gold
Discount	
PurchaseHistoryCount	12
OpenInvoices	12
BillingAddress	
ShippingAddress	
_env	

At the bottom of the window are five buttons: a question mark icon, 'Load...', 'Save...', 'Clear', 'OK', and 'Cancel'. The 'Clear' button is highlighted with a blue border.

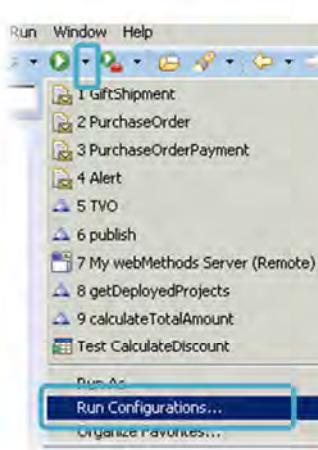
- Results are shown in Results view:

DemoRuleProject/Decision Trees/DetermineDiscountTree (Feb 7, 2017 2:15:30 PM)	
Name	Value
Customer_1	
Discount	10
OpenInvoices	12
PurchaseHistoryCount	12
Age	57
FirstName	Johanna
LastName	Barodscheff
Rating	gold
_env	
BillingAddress	
ShippingAddress	

Software AG Training | 6 - 14

Notes:

Test Decision Tree using Run Configuration



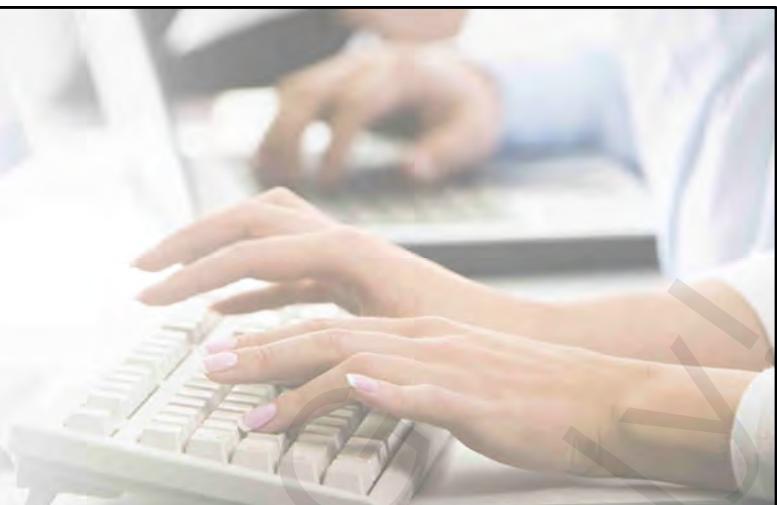
The screenshot shows the 'Run Configurations...' option highlighted in the toolbar. The main window displays the 'Run Configurations' dialog box, which is used to create, manage, and run configurations for various projects and entities. In the dialog, a configuration named 'DetermineDiscountTree' is selected under the 'Decision Entity' section. The 'Input' section shows a table with the following data:

Name	Type	Value
Customer_I	Customer	
FirstName	String	Joltawan
LastName	String	Barodscheff
Age	Integer	57
Rating	String	gold
Discount	Double	12
PurchaseHistoryCount	Integer	

At the bottom of the dialog, there are 'Save...', 'Load...', 'Clear', 'Run', and 'Close' buttons.

- Usage:
 - Recurring Tests
 - Logging

Notes:



Exercise 4

- In this exercise, you will add a Decision Tree to your Rule Project. The Decision Tree acts on the PurchaseOrder Data Model and identifies premium orders based on customer's number of open invoices and rating as well as the number of ordered items in an order. Premium orders are shipped free of charge, other orders will have assigned ranked shipping costs.

Notes:



7

Service Actions and New Data Actions

Notes:

Objectives

At the end of this chapter you ...

- Can name the different Actions types offered by webMethods Business Rules
- Can explain the usage of Service and Data Actions
- Created Service and Data Actions using Designer
- Invoked Service and Data Actions in Decision Entity results
- Used Service Actions in Decision Entity Assignments
- Tested Service and Data Actions via Decision Entities

Notes:

Actions - Overview

- Business Rules supports Actions to be defined in a Rule Project
- Supported Action types:
 - Service
 - New Data
 - Process
- Actions may return parameter elements as results
- Defined Actions can be used in a Decision Entity
(Decision Table, Decision Tree, Event Rule):
 - As result
 - At the right side of an Assignment creating the value to be assigned



Notes:

Software AG Training

Service Actions

- Service Actions invoke an IS service
- Service Actions require for "Business Rules Add-ons" license (WOKAA)
- *Example:* Calculation of a total amount based on a list of values

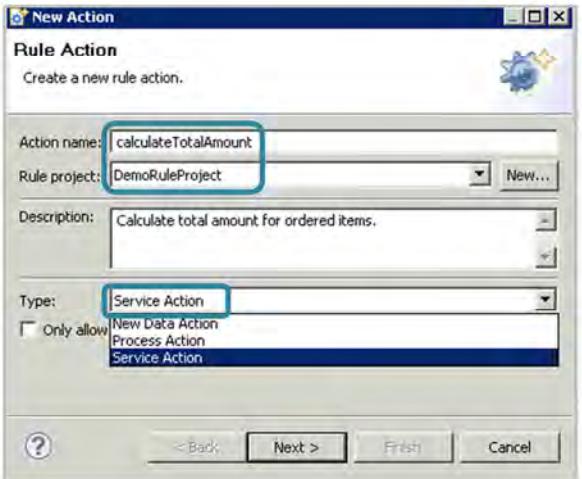
- Create a Service Action:
 1. Name, description, Action type (Service)
 2. Select IS service
 3. IS Service input default values (*optional*)
 4. Select IS service outputs as result



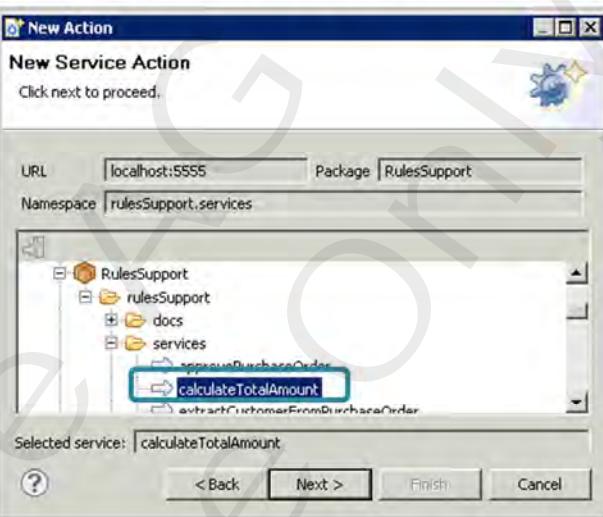
Software AG Training | 7 / 4

Notes:

1-2: Create Service Action



- Provide name and description (*optional*)
- Choose type "Service Action"



- Select IS service to be invoked from Package Navigator
- Requires an active IS connection

Software AG Training | 7 - 5

Notes:

3-4: Create Service Action

The screenshot shows two windows from a software application:

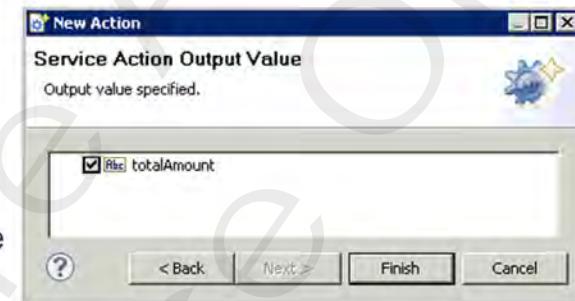
- New Action - Service Action Default Input Values:** A dialog box titled "Service Action Default Input Values" with the sub-instruction "Set any desired default values for inputs." It contains a table for "Action Input Parameters" with columns "Default Values" and "Include empty values for String Types". The table lists:
 - PurchaseOrder:
 - OrderNumber
 - NumberOfOrderedItems
 - ShippingCost
 - TotalAmount (selected, value 0)
 - Customer:
 - FirstName
 - LastName

Buttons at the bottom include "< Back", "Next >", "Finish", and "Cancel".
- New Action - Service Action Output Value:** A dialog box titled "Service Action Output Value" with the sub-instruction "Output value specified." It contains a table with one row:

<input checked="" type="checkbox"/> Rbc totalAmount

Buttons at the bottom include "?", "< Back", "Next >", "Finish", and "Cancel".

- Provide default values for IS service inputs (optional)



Software AG Training | 7 - 6

Notes:

Software AG

Invoke Service Actions - Overview

- Service Actions cannot be executed standalone
- Invoke Service Actions from a Decision Entity as
 - a result
 - within an assignment
- Service Actions can be picked and mapped:
 - Within creation wizard for a Decision Table or Event Rule
 - In the Decision Table or Event Rule Editor when adding an Action column
 - In the Decision Tree Editor when adding an Action node
 - In the Extended Cell Editor for an Assignment



Software AG Training | 7 / 7

Notes:

Add Service Action as Result to a Decision Table...

Add a new action to the end

Action – Add a new Action

drag

Description
determines total amount to be invoiced.

NumberofOrderedItems	Rating	TotalAmount
1	= gold	= calculateTotalAmount()
2	= silver	= calculateTotalAmount()
3	= regular	= calculateTotalAmount()
4	= fraud	= 0

type filter text

- DemoRuleProject
- HRBusinessRules
- OrderRulesProject
 - Actions
 - + alertIncompleteBillingAddress()
 - + calculateTotalAmount()
 - + cancelGiftShipmentForRegularCustomer()
 - + joinToOrderingWithCorrelationProcess()
 - + sendMailNotificationForPremiumCustomer()
 - + StartAlertHandlingProcess()
 - + startApproveMassOrderTask()
 - Data Models
 - Decision Tables
 - + CalculateDiscount
 - + CalculateDiscountSpecialOffer

▪ Selection via

- Drag from Rules Explorer OR
- Palette OR
- Toolbar

▪ Mapping editor will be offered after selection (see previous slide)

Software AG Training | 7 - 8

Notes:

...Add Service Action as Result to a Decision Table

The screenshot shows the 'Action Selection' window where a service action named 'calculateTotalAmount' is selected and set to 'active'. Below it, the 'Map Action Input' window shows the mapping of input parameters from a 'PurchaseOrder' pipeline to the 'calculateTotalAmount' action.

- Select the Service Action to be invoked by your Decision Entity as result
 - Action is pre-selected when dragging from Rules Explorer view
- Choose default status (active/inactive) for all rules

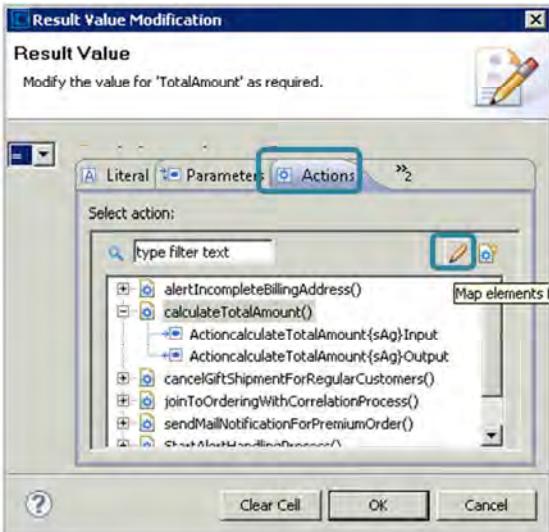
Map Action Input

Map elements from the decision entity's input parameters to the action's input.

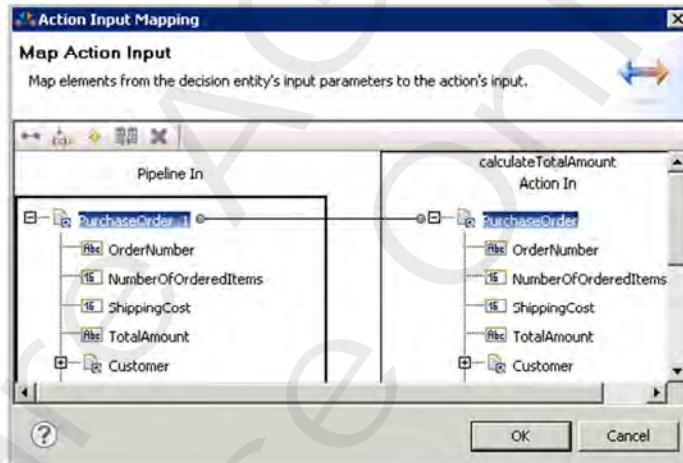
Pipeline In	Action In
<ul style="list-style-type: none"> - PurchaseOrder_1 <ul style="list-style-type: none"> - OrderNumber - NumberOfOrderedItems - ShippingCost - TotalAmount - Customer - OrderedItems - _env 	<ul style="list-style-type: none"> - PurchaseOrder <ul style="list-style-type: none"> - OrderNumber - NumberOfOrderedItems - ShippingCost - TotalAmount - Customer - OrderedItems - _env

Notes:

Use Service Action in an Assignment



- Click on green pencil in cell to open the Extended Cell Editor
- Select Service Action on Actions tab



- Click on orange pencil to open the Mapping editor
- Map Decision Entity parameter to Service Action

Software AG Training | 7 - 10

Notes:



Exercise 5 + 6

- In exercise 5, you will enhance the Decision Tree created in the previous exercise by the invocation of a Service Action. The invoked Service Action also acts on the PurchaseOrder Data Model and uses an IS service to send an email notification to the process administrator. It will be invoked by the Decision Tree as a result whenever a premium order has been detected.
- In exercise 6, you will add a Service Action that calculates the total amount to be invoiced for all items in an order. This Service Action will be used in a new Decision Table within an assignment. The execution of the Service Action depends on the customer's rating.

Notes:

Software AG

New Data Action

- New Data Actions create new named Parameter (Fact) of a certain Data Model with optional default values
- Can be used by other Decision Entities in the same Rule Set "catching" the Fact using inference
- Mainly intended for use with external Business Events
- New Data Actions require for "Business Rules Add-ons" license (WOKAA)

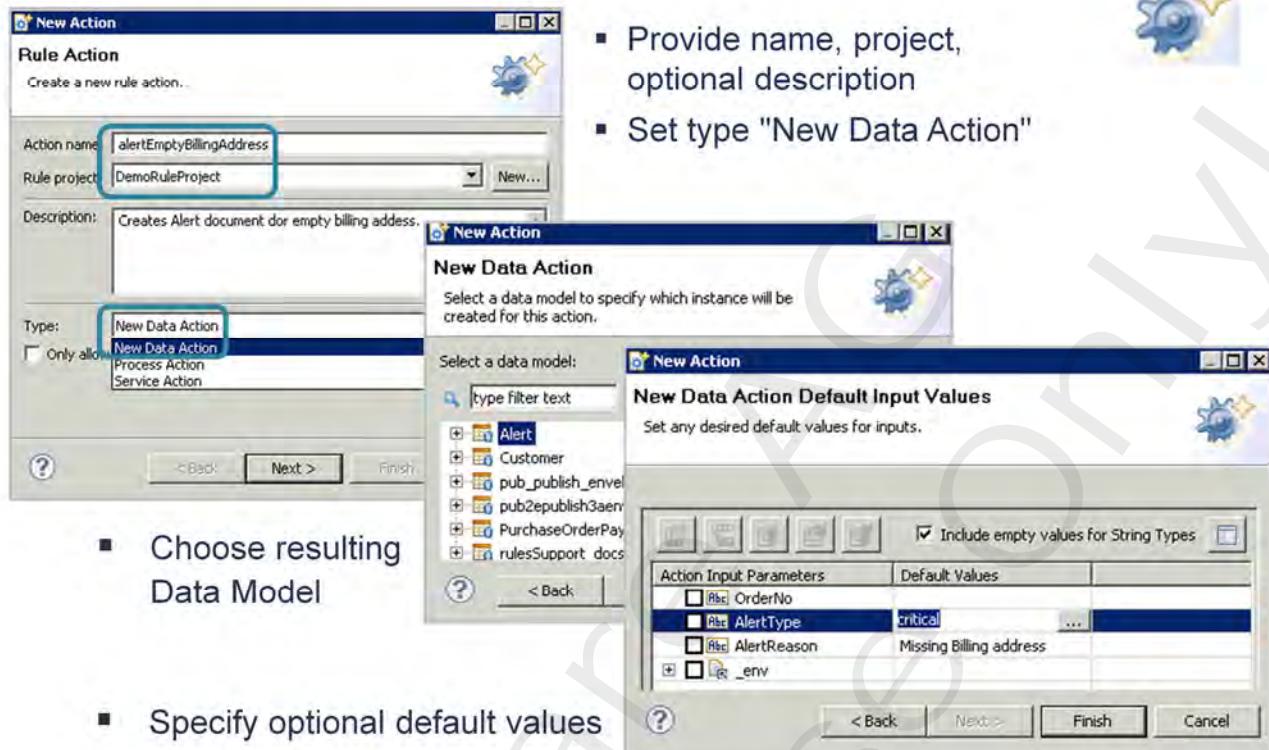
- Create a New Data Action:
 1. Name, description, Action type (New Data)
 2. Choose resulting Data Model
 3. Specify default values (*optional*)



Software AG Training | 7-12

Notes:

1-3: Create a New Data Action

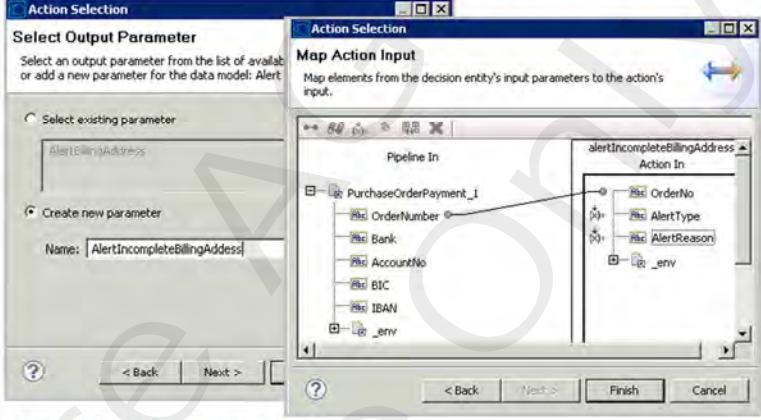


Notes:

Invoke New Data Actions

- New Data Actions cannot be executed standalone
- Invoke New Data Actions from a Decision Entity as a result
 - Decision Entity has to include the New Data Action's return parameter as an output parameter
 - When adding and it is missing, you will be prompted
 - Like for a Service Action, you have to map or assign values to the Action's input

- New Data Actions can be picked and mapped:
 - Within creation wizard for a Decision Table or Event Rule
 - In the Decision Table or Event Rule Editor when adding an Action column
 - In the Decision Tree Editor when adding an Action node



Software AG Training | 7 - 14

Notes:



Exercise 7

- In this exercise, you will add a Decision Table and a New Data Action to your Rule Project. The New Data Action returns an Alert parameter with some preconfigured parameter elements. This Action will be executed by the Decision Table when an incomplete billing address has been detected. Depending on the number of empty parameter elements, different alerts will be thrown.

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



8

Built-in Functions

Notes:

Objectives

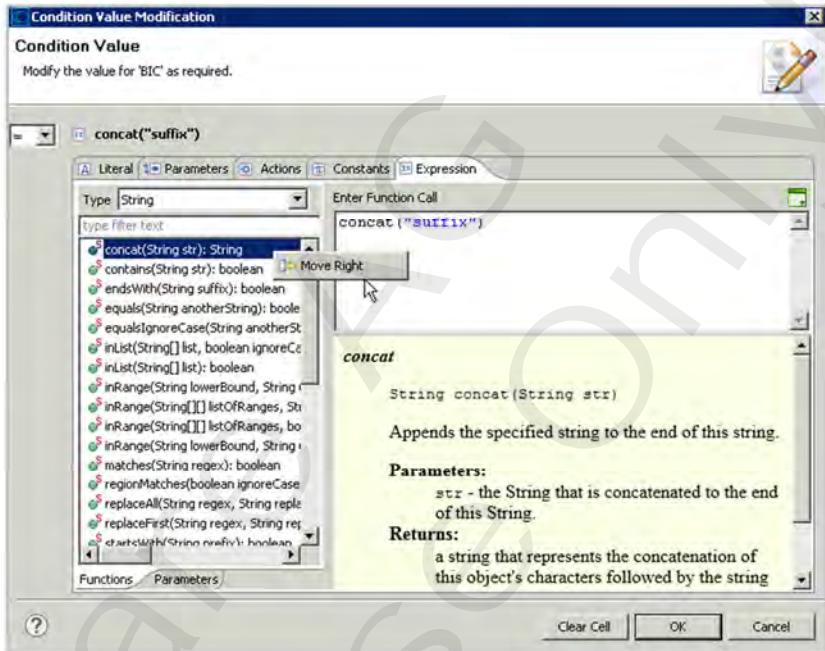
At the end of this chapter you ...

- Know how to invoke Build-in Functions for rule definitions in Designer
- Can invoke Build-in Functions returning boolean or Strings
- Are able to nest or chain Build-in Functions
- Can specify mathematical expressions in your rules
- Can work with dates using Build-in Functions

Notes:

Built-in Functions in Business Rules...

- Rule Editors allow to add and modify simple or complex Expressions
- Can be used in
 - Conditions
 - Assignments
 within
 - Decision Tables
 - Decision Trees
 - Event Rules
- To build an Expression, select function from the list and configure
- Extensive help and samples available



The screenshot shows the 'Condition Value Modification' dialog box. In the 'Enter Function Call' field, 'concat("suffix")' is typed. Below it, the 'concat' function is selected from a list of methods for the 'String' type. The right-click context menu is open over the 'concat' entry, with the 'Move Right' option highlighted. A tooltip for the 'concat' function is displayed, stating: 'String concat(String str) Appends the specified string to the end of this string.' It also specifies parameters: 'str - the String that is concatenated to the end of this String.' and returns: 'a string that represents the concatenation of this object's characters followed by the string'. At the bottom of the dialog are 'Clear Cell', 'OK', and 'Cancel' buttons.

Notes:

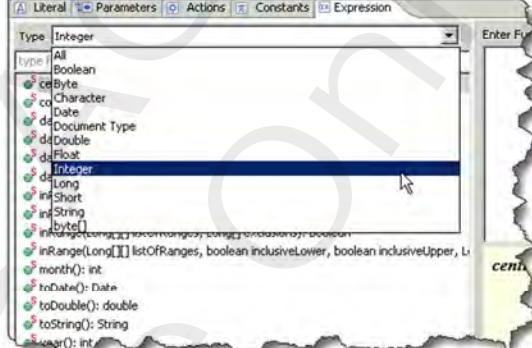
...Built-in Functions in Business Rules

- No need for service invocations
 - Functions are executed within the same Java Runtime Environment as the Rule Engine
 - Exist as static methods within a Java class
- Support for String, Boolean and mathematical expressions & functions
- Functions can:
 - receive decision entity's parameter elements as arguments
 - can be nested and/or chained
- Overloaded functions are supported
 - Same function names, different signatures

Explain what business functions are

Built-in Functions in Assignments

- For Assignments, Functions return type must match with column or node parameter type:
 - Functions that return Boolean can only be placed in columns/nodes of type Boolean
 - Functions that return String can only be placed in columns/nodes of type String
 - ...
- The return value of the Function will be assigned to the parameter element for the column or node
- Setting Function invocations in Event Rules can only be done for the Assignment cells



The screenshot shows a dropdown menu titled "Assignment" with the following options:

- Literal
- Parameters
- Actions
- Constants
- Expression

Below the menu, a "Type" dropdown is set to "Integer". The main list contains the following items:

- All
- Boolean
 - ceByte
 - coCharacter
 - dDate
 - dDocument Type
 - dDouble
 - dFloat
 - dInteger
 - dLong
 - dShort
 - dString
 - byte[]
 - inRange([Long[]]) listOfRanges, boolean inclusiveLower, boolean inclusiveUpper, Long[]
 - month(): int
 - toDate(): Date
 - toDouble(): double
 - toString(): String
 - year(): int

Notes:

Built-in String Functions that Return Boolean...

- boolean **contains**(String str)
- boolean **endsWith**(String suffix)
- boolean **equals**(String anotherString)
- boolean **equalsIgnoreCase**(String anotherString)
- boolean **inList**(String[] list)
- boolean **inList**(String[] list, boolean ignoreCase)
- boolean **inRange**(String lowerBound, String upperBound)
- boolean **inRange**(String lowerBound, String upperBound, String[] exclusions)
- boolean **inRange**(String[][] listRanges, String[] exclusions)
- boolean **inRange**(String[][] listRanges, boolean inclusiveLower, boolean inclusiveUpper, String[] exclusions)
- boolean **inRange**(String[][] listRanges, boolean ignoreCase, boolean inclusiveLower, boolean inclusiveUpper, String[] exclusions)

List String functions that return boolean

...Built-in String Functions that Return Boolean

- boolean **matches**(String regex)
- boolean **regionMatches**(boolean ignoreCase, int toffset, String other, int ooffset, int len)
- boolean **startsWith**(String prefix)

List String functions that return boolean

Built-in String Functions that Return Boolean – Remarks...

- The return value of the function will either be True or False
- When used in a condition cell or node, the return value of the Function will not be evaluated against the parameter element of the column/node in which it resides unless the column is of type Boolean
- In a condition cell or node of type String, the return value of the Function will be tested for True
 - In a condition cell or node, the operator can be = (equals) or != (not equals)
 - So the return value of the Function can be tested for = True and != True
- For assignments, Functions that return Boolean can only be placed in columns/nodes of type Boolean
 - The return value of the Function will be assigned to the parameter element for the column/node

Explain the behavior of String functions that return boolean and how they differ in condition cells vs. assignment cells.

...Built-in String Functions that Return Boolean – Remarks

- `inRange` is an overloaded Function that tests if a given String is within a given range or set of ranges
 - A list of exclusions can be provided. If the String to be checked is one of the provided exclusions, the `inRange` function will return `False`
 - Case sensitivity can be ignored if desired

Discuss the `inRange` function and how it also supports integer, short, long and double data types

Built-in Functions that Return Boolean for Integer and Short

- **inRange** Function also supports the following numeric data types:
 - Integer
 - Short
 - Long

- boolean **inRange**(Long lowerBound, Long upperBound)
- boolean **inRange**(Long lowerBound, Long upperBound, Long[] exclusions)
- boolean **inRange**(Long[][] listOfRanges, Long[] exclusions)
- boolean **inRange**(Long[][] listOfRanges, boolean inclusiveLower, boolean inclusiveUpper, Long[] exclusions)

List String functions that return String

Built-in Functions that Return Boolean for Long and Double

- `inRange` function also supports the following numeric data types:
 - `Double`
 - `Float`

- `boolean inRange(Double lowerBound, Double upperBound)`
- `boolean inRange(Double lowerBound, Double upperBound, Double[] exclusions)`
- `boolean inRange(Double[][] listOfRanges, Double[] exclusions)`
- `boolean inRange(Double[][] listOfRanges, boolean inclusiveLower, boolean inclusiveUpper, Double[] exclusions)`

List String functions that return String

Built-in String Functions that Return String

- String **concat**(String str)
- String **replaceAll**(String regex, String replacement)
- String **replaceFirst**(String regex, String replacement)
- String **substring**(int beginIndex)
- String **substring**(int beginIndex, int endIndex)
- String **toLowerCase**()
- String **toUpperCase**()
- String **trim**()

List String functions that return String

Built-in String Functions that Return String – Remarks

- The return value of the Function will be a String
- When used in a condition cell or node, the return value of the Function will be compared to the parameter element of the column/node in which it resides
- For assignments, Functions that return String can only be placed in cells/nodes of type String
 - The return value of the Function will be assigned to the parameter element for the cell/node

Explain the behavior of String functions that return String and how they behave in condition and assignment cells.

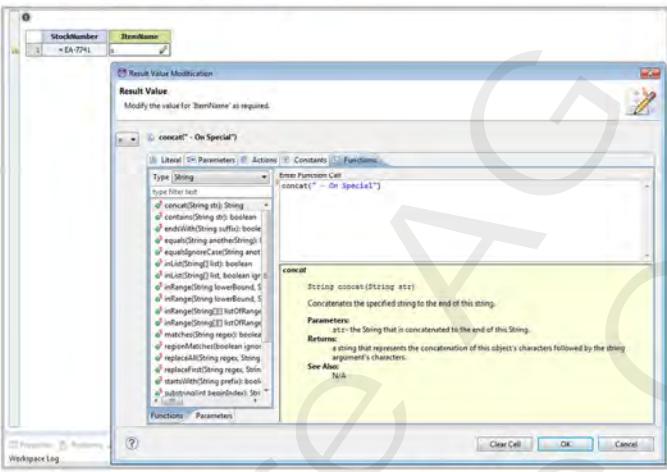
Implied Parameter

- All of the preexisting built-in Functions have an implied parameter
- A Function can be invoked by its name only (e.g., `concat ("suffix")`)
 - In this case, the implied parameter is the parameter element of the cell/node that the Function call resides within
- A Function can be invoked using a parameter element (e.g., `%"Item.StockNumber"%.concat ("suffix")`)
 - In this case, the implied parameter is the parameter element `Item.StockNumber`
 - Note the parameter must be enclosed in `%"..."%`
- The String value "suffix" is a parameter to the `concat ()` Function, but it is not implied

Explain/illustrate the inner workings of the implied parameter in function signatures

Implied Parameter – Sample 1

- If we have concat(" - On Special") in a cell within the assignment column Item.ItemName, then the String " - On Special" will be concatenated to the end of the value of Item.ItemName when the rule fires



The screenshot shows the 'Result Value Modification' dialog box. In the 'Enter Process Cell' field, the expression 'concat(" - On Special")' is entered. Below it, the 'concat' function is expanded, showing its definition: 'Concatenates the specified string to the end of this string.' The 'Parameters' section shows 'str1: the String that is concatenated to the end of this String.' The 'Returns' section states 'a string that represents the concatenation of this object's characters followed by the string argument's characters.'

- If at the time this rule fires Item.ItemName contains the value "Laser Projection Keyboard", the concat Function will change it to "Laser Projection Keyboard - On Special"

Software AG Training | 8 - 15

Give an example of the implied parameter when used in a standalone (no parameter element) function call

Implied Parameter - Sample 2

- If we have `%"Customer.ShippingAddress.City"%.toUpperCase()` in a cell within the condition column `Customer.CustomerAddress.City`, then the value of `Customer.ShippingAddress.City` will be converted to upper case, and this value will be compared to the value of `Customer.CustomerAddress.City` during rule evaluation

The screenshot shows the webMethods Business Rules interface with a function call dialog open. The dialog has the title 'Condition Value Modification' and 'Modify the value for 'City' as required.' The 'Enter Function Call' field contains the expression `'%"Customer.ShippingAddress.City"%.toUpperCase()`. The 'Functions' tab is selected in the dialog, and the 'concat' function is highlighted. The 'concat' function documentation is displayed below, showing its purpose of concatenating strings, parameters, and return type.

- During rule evaluation if `Customer.ShippingAddress.City` contains the value "New York", it will be converted to "NEW YORK" and compared to the value of `Customer.CustomerAddress.City`

Software AG Training | 8 - 16

Give an example of the implied parameter when a function call is used with a parameter element

Chained/Nested Built-in Function Invocations

Functions can be chained nested and/or chained*

- Chained function invocations:

- `toLowerCase().trim().concat("suffix")`
- `substring(1, 5).equals("%Item.StockNumber%")`

- Nested function invocations:

- `endsWith("%Item.StockNumber%").toLowerCase()`
- `endsWith("%Item.StockNumber%").concat("%Item.Name%").toLowerCase()`

- Nested and Chained

- `endsWith("%Item.StockNumber%").concat("%Item.Name%").toLowerCase().trim()`

Show some examples of chained, nested and both (chained and nested) function invocations

Available Math Functions...

- long **abs**(long value)
- double **abs**(long double)
- double **acos**(double val)
- double **asin**(double val)
- double **atan**(double val)
- double **ceil**(double val)
- double **cos**(double val)
- double **cosh**(double val)
- double **degreesToRadians**(double angdeg)
- double **exp**(double val)
- double **floor**(double val)
- double **log**(double val)
- long **max**(long val1, long val2)
- double **max**(double val1, double val2)
- long **min**(long val1, long val2)
- double **min**(double val1, double val2)

A list of Math functions.

...Available Math Functions

- long **mod**(long val1, long val2)
- double **mod**(double val1, double val2)
- double **pi()**
- double **pow**(double base, double exponent)
- double **radiansToDegrees**(double angdeg)
- long **round**(double val)
- double **round**(double val, int scale)
- double **round**(double val, int scale, int roundingMethod)
- double **sin**(double val)
- double **sinh**(double val)
- double **tan**(double val)
- double **tanh**(double val)

Notes:

Math Function – Samples

- `abs(%"Param_1.LongVar")`
 - If the value of parameter Param_1.LongVar is -50, then the Function will return +50
- `max(long -5, %"Param_1.LongVar")`
 - If the value of parameter Param_1.LongVar is -50, then the Function will return -5
- `round`
 - `round(1.001, 3, 0)` returns 1.001
 - `round(1.001, 2, 0)` returns 1.01
 - `round(1.001, 1, 0)` returns 1.1
 - `round(1.001, 1)` returns 1.0
- `pow(10.0, 3.0)` returns 1000.0

Some examples of Math functions

Build-in Functions for Dates...

- int **century()**
- int **compareDates**(Date date1, Date date2)
- Date **date()**
- Date **dateMinusDays**(int days)
- Date **datePlusDays**(int days)
- long **dateToInt()**
- int **dayOfMonth()**
- int **dayOfWeek()**
- int **dayOfYear()**
- int **daysInMonth**(int month, int year)
- long **diffInDays**(Date anotherDate)
- long **diffInMonths**(Date anotherDate)
- long **diffInYears**(Date anotherDate)

A list of Date functions.

...Build-in Functions for Dates

- Date **intToDate**(long millis)
- **isLeapYear**(int year)
- int **month**()
- String **time**()
- boolean **verifyDate**(int year, int month, int day)
- boolean **verifyIntDate**(long millis)
- boolean **verifyMonth**(int month)
- boolean **verifyYear**(int year)
- int **year**()
- Date **ymdToDate**(int year, int month, int day)

Notes:

Build-in Functions for Dates - Samples

- `%"AllTypes_1.DateVar"%.diffInYears('10/12/1992')`
 - If the value of the parameter `AllTypes_1.DateVar` is '10/12/1492', then the Function will return 500 (500 year difference between '10/12/1992' and '10/12/1492')
- `%"AllTypes_1.DateVar"%.diffInMonths('10/12/1992')`
 - If the value of the parameter `AllTypes_1.DateVar` is '10/12/1492', then the Function will return 60,000 (60,000 month difference between '10/12/1992' and '10/12/1492')
- `%"AllTypes_1.DateVar"%.diffInDays('10/12/1992')`
 - If the value of the parameter `AllTypes_1.DateVar` is '10/12/1492', then the Function will return 182,621 (182,621 day difference between '10/12/1992' and '10/12/1492')
Note: there are 121 leap years in this range
- `date().dateMinusDays(8)`
 - The Function `date()` returns the current system date, so if the date is '11/26/2014', then the Function will return '11/18/2014' (8 days earlier)
- `date().dateMinusDays(8).dayOfMonth()`
 - If the date is '11/26/2014', then the Function will return 18 (for the 18th day of November)

Some examples of Date functions.

Built-in Functions for Dates – inRange Function...

- inRange is an overloaded Function that also tests if a given Date is within a given range or set of ranges
- A list of exclusions can be provided
 - If the Date to be checked is one of the provided exclusions, the inRange Function will return False
 - When checking a date against exclusions, the time of day is ignored
- boolean **inRange**(Date lowerBound, Date upperBound)
- boolean **inRange**(Date lowerBound, Date upperBound, Date[] exclusions)
- boolean **inRange**(Date[][] listOfRanges, Date[] exclusions)
- boolean **inRange**(Date[][] listOfRanges, boolean inclusiveLower, boolean inclusiveUpper, Date[] exclusions)
- boolean **inRange**(Date[][] listOfRanges, boolean ignoreCase, boolean inclusiveLower, boolean inclusiveUpper, Date[] exclusions)

A list of new inRange functions (for Date only)

...Built-in Functions for Dates – inRange Function

Example:

```
inRange('11/3/2015 9:00:00 AM', '11/7/2015 5:00:00 PM',
{'11/5/2015 11:00:00 AM', '11/6/2015 12:00:00 AM'})
```

- In the above example, the exclusions are 11/5/2015 11:00:00 AM and 11/6/2015 12:00:00 AM
- The time of day in the exclusions is not necessary, and will be ignored when checking
- So the date for the condition cell/node value must be \geq '11/3/2015 9:00:00 AM' (time of day included) and \leq 11/7/2015 5:00:00 PM; but if it falls on the dates 11/5/2014 or 11/6/2015 then the rule will not fire

An example of the inRange function

Date Formatting

- Internally, dates will be persisted as a long integer value
 - This way, the decision entities can be shared across locales
- The date format will not be persisted
- Dates may be entered using a wide variety of formats, but once persisted in a decision entity, upon reloading a predefined format will be used to display
- Dates used within Function calls will be displayed in a locale specific format
 - Examples:
 - English, United States: 2/4/14 2:08:28 PM
 - English, Great Britain 04/12/14 14:08:28
 - German, Germany: 04.12.14 14:08:28
 - Hindi, India: ४/१२/१४ २:०८:२८ अपराह्न
 - Chinese, Taiwan: 2014/12/4 下午 02:08:28

Date formatting within function calls

Build-in Conversion Functions

The following conversion functions are available:

- **toBoolean()**
- **toDate()**
- **toDate(String dateFormat)**
- **toDouble()**
- **toLong()**
- **toString()**
- **toString(String dateFormat)**

Invocation using chaining!

A list of Conversion functions.

Build-in Conversion Function - Samples

- Param1.BoolStr.ToBoolean()
 - If the parameter Param1.BoolStr contains the value "true" or "TRUE", then the Function will return "true"
 - If the parameter Param1.BoolStr is null, then a runtime error will occur
 - If the parameter Param1.BoolStr contains any other text, then the Function will return false
- Param1.NumStr.ToDouble()
 - If the parameter Param1.NumStr contains the text "3.14159", then the Double value 3.14159 will be returned from the Function
- Param1.LongVal.ToDouble()
 - If the parameter Param1.LongVal contains the value 5280, then the Double value 5280.0 will be returned from the Function
- Param1.DoubleVal.ToInt()
 - If the parameter Param1.DoubleVal contains the value 3.14159, then the Long value 3 will be returned from the Function

Some examples of Conversion functions.

Mathematical Expressions

- An Expression may contain:
 - Parameters references
 - Literals
 - Operators (+, -, *, /)
 - Parentheses – ()

Operators may be used with or without functions

Mathematical Expressions - Samples

- Compute the area of a rectangle:
 - `%"Shapes_1.Rect_Height"% * %"Shapes_1.Rect_Width"%`
- Compute the perimeter of a rectangle:
 - `(2 * %"Shapes_1.Rect_Height"%) + (2 * %"Shapes_1.Rect_Width"%)`
- Compute the circumference of a circle:
 - `pi() * %"Shapes_1.Circle_Diameter"%`
 - `pi() * 2.0 * %"Shapes_1.Circle_Radius"%`
- Compute the area of a circle:
 - `pi() * pow(%"Shapes_1.Circle_Diameter%" / 2, 2.0)`
- Compute the area of a circle rounded:
 - `round(pi() * pow(%"Shapes_1.Circle_Diameter%" / 2, 2.0))`
- Compute the length of a side of a right triangle:
 - `sin(%"Shapes_1.AngleDegrees") * %"Shapes_1.Hypotenuse"%`

Operators may be used with or without functions



Exercise 8

- Sales department likes to offer a special discount rate for customers of a particular region in Europe regardless of their status and rating. In this exercise you will created an enhanced Decision Table based on the existing CalculateDiscount Decision Table that returns the special discount rate for a selected customer subset.

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



9

Deployment to IS and MWS

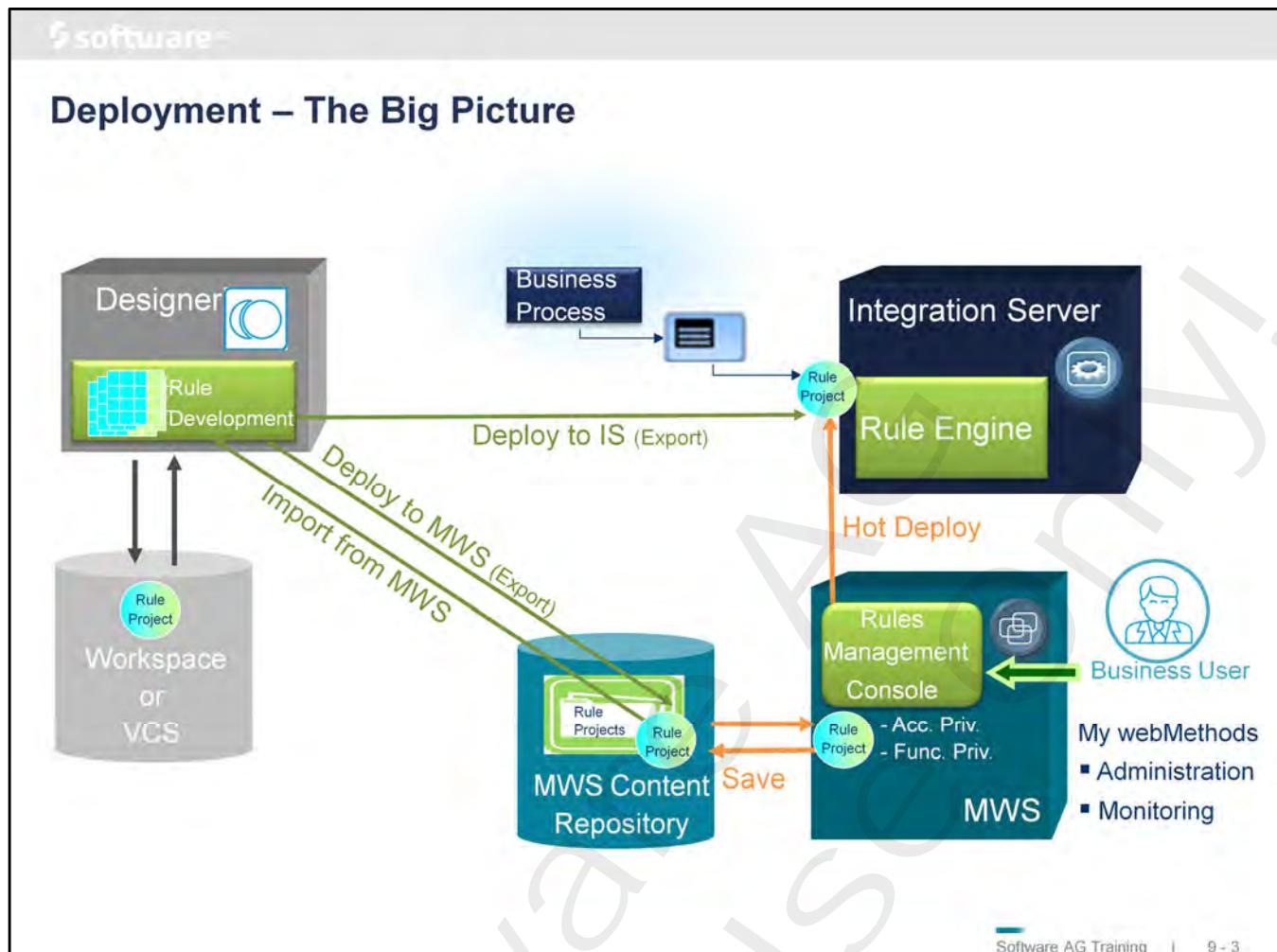
Notes:

Objectives

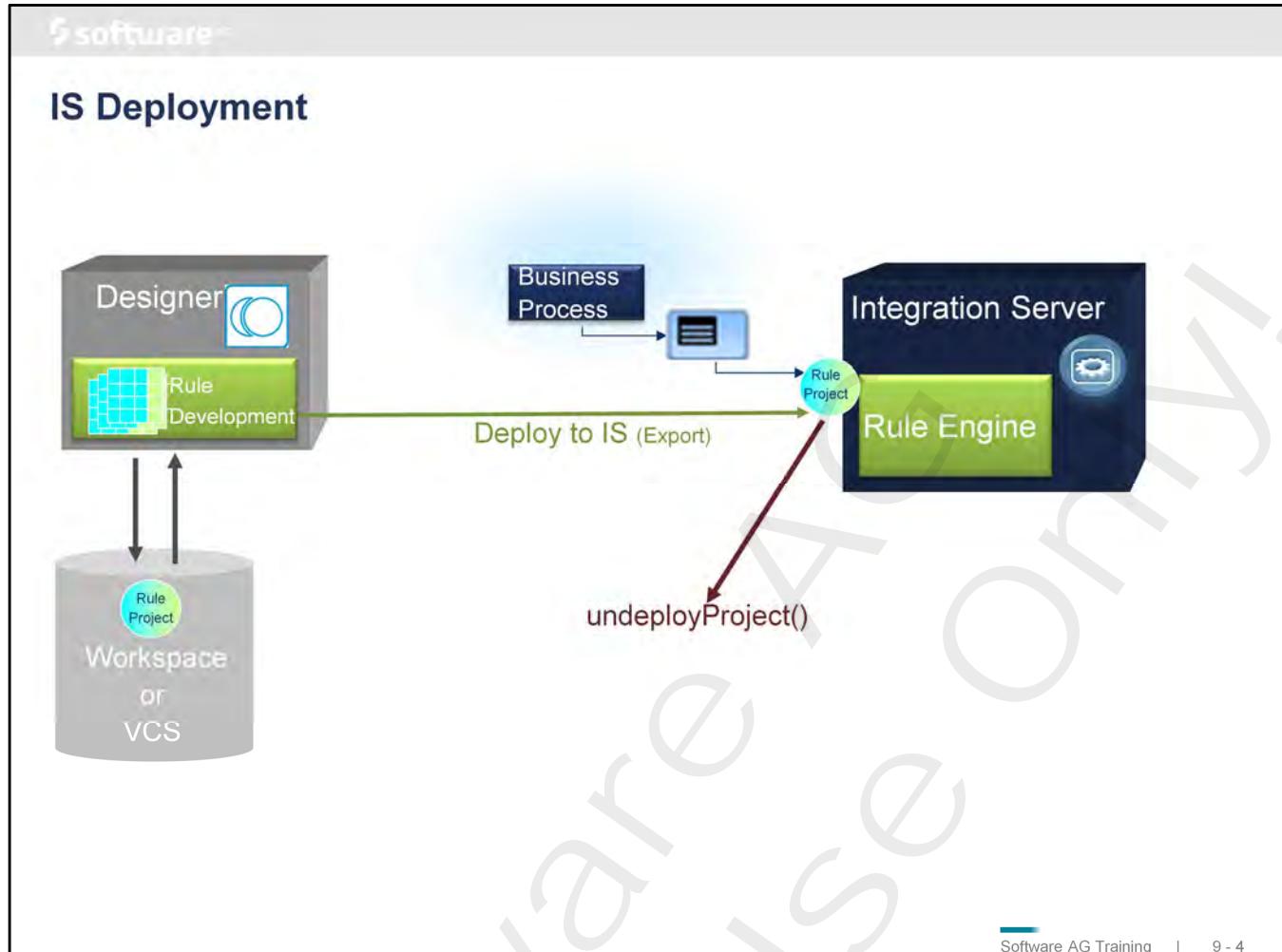
At the end of this chapter you ...

- Can describe Rule runtime environments and deployment
- Deployed Rule projects to IS runtime environments
- Configured MWS and My webMethods for Rule deployment
- Deployed a Rules Management Console to MWS

Notes:



Notes:



Notes:

Integration Server as Rule Runtime Environment

- For a production environment, Rule Projects have to be deployed to an Integration Server
- Integration Server hosts webMethods Business Rule Engine (like Designer)
- Deployment can be performed by
 - Designer (Export...)
 - Deployer



Notes:

The screenshot displays three main panels of the webMethods Integration Server interface:

- Left Panel (File Explorer):** Shows the package structure under 'Default'. The 'WmBusinessRules' package is highlighted. It contains sub-folders 'pub' and 'wm', each with 'businessrules' sub-folders. 'wm/businessrules' contains 'admin', 'deployer', 'deployerSPI2', 'docTypes', 'eda', 'runtime' (with 'archive', 'consumer', 'info', 'pool'), and 'spi2services'.
- Middle Panel (Server Management):** Shows the navigation menu with 'Scheduler', 'Service Usage', and 'Statistics' under 'Server'. Under 'Management', it lists 'Publishing', 'Subscribing', 'Adapters', 'Security', and 'Settings'.
- Right Panel (Package Management):** Shows the details for the 'WmBusinessRules' package. The package information table includes:

Package Name	WmBusinessRules
Version	9.12.0.1.291
Build	291
Minimum Version of JVM	1.8
Package List ACL	Developers
Patches Included	
Description	Support services for webMethods Business Rules
Publisher	Software AG
Created on	
Elements Loaded	192
Elements Not Loaded	0
Startup Services	wm.business.rules.adminsStartup
Shutdown Services	wm.business.rules.adminsshutdown
Replication Services	None
Packages on which this package depends	None
Packages that depend on this package	None
Subscribers	None

 Below the package information are sections for 'Load Errors' (No errors), 'Load Warnings' (No Warnings), and 'Patch History' (No Patches).

Bottom Left: A note states: "Rule Engine in IS can also be monitored and administered using Command Central".

Notes:

Rule Project Deployment from Designer to IS...

- To deploy a Rule Project to an IS hosting the Rule Engine use Export from Designer's:
 - File menu
 - Rules Explorer view
 - Solutions view

The screenshot illustrates the deployment process. On the left, the 'Rules Explorer' window shows a project structure with several nodes like 'DemoRuleProject', 'OrderRulesProject', and various rule sets. A context menu is open over the 'Event Rules' node under 'OrderRulesProject', with the 'Export...' option highlighted by a red box. On the right, the 'Export' dialog box is displayed, titled 'Select'. It contains a tree view under 'Select an export destination:' with various options. The 'Rule Project to Integration Server runtime' option is selected and highlighted by a blue box. At the bottom of the dialog, there are 'Back', 'Next >', 'Finish', and 'Cancel' buttons. The status bar at the bottom right of the dialog box shows 'Software AG Training | 9 - 7'.

Notes:

... Rule Project Deployment from Designer to IS

- Requires a configured Integration Server connection in Designer

The screenshot shows two windows side-by-side. On the left is the 'Integration Servers' configuration window, which lists a single server named 'Default' with 'localhost' as the host name and port '5555'. The status is 'Connected'. On the right is the 'Export Rule Project' dialog, which is titled 'Export Rule Project to Integration Server Runtime Environment'. It shows the 'Integration Server' dropdown set to 'Default' and the 'Rule project' dropdown containing several options: 'OrderRulesProject', 'DemoRuleProject', 'OrderRulesProject' (which is selected), 'ProviderServiceDemo', and 'WmClosedLoopAnalytics'. The 'Version (generated)' dropdown shows '2017-03-27'. At the bottom of the dialog are 'Back', 'Next', 'Finish', and 'Cancel' buttons.

Notes:

Optional: Check IS Deployment...

- Deployed Rules projects are stored as .jar files in the projects folder within package WmBusinessRules:

Name	Date modified	Type
deployedProjects	11/18/2014 1:50 PM	File folder
HRBusinessRules.jar	11/18/2014 1:50 PM	Executable Jar File
OrderRulesProject.jar	3/10/2015 11:33 AM	Executable Jar File
WmClosedLoopAnalytics.jar	12/9/2014 2:13 PM	Executable Jar File

Notes:

... **Optional:** Check IS Deployment

- Deployed projects could also be checked by running the (internal) IS service `wm.businessrules.runtime.info:getDeployedProjects`
- Service has no input

The screenshot shows the webMethods Business Rules interface. On the left, the 'Package Navi...' tab is selected in the outline view, displaying a tree structure of projects and services. A service named 'getDeployedProjects' under the 'info' folder of the 'WmBusinessRules' project is highlighted with a red box. On the right, the 'Properties' tab is active in a results panel. The URL shown is '[localhost:5555] wm.businessrules.runtime.info:getDeployedProjects (Mar 11, 2015 3:05:55 PM)'. The results table lists three deployed projects:

Name	Value
Project Information List[0]	
Project Name	HRBusinessRules
Project Version	2014-11-18T13:50:27_D
Project Information List[1]	
Project Name	OrderRulesProject
Project Version	2015-03-10T11:33:13_M
Project Information List[2]	
Project Name	WmClosedLoopAnalytics
Project Version	2014-12-09T14:13:37_D

Notes:

Undeploy Rules Project from IS

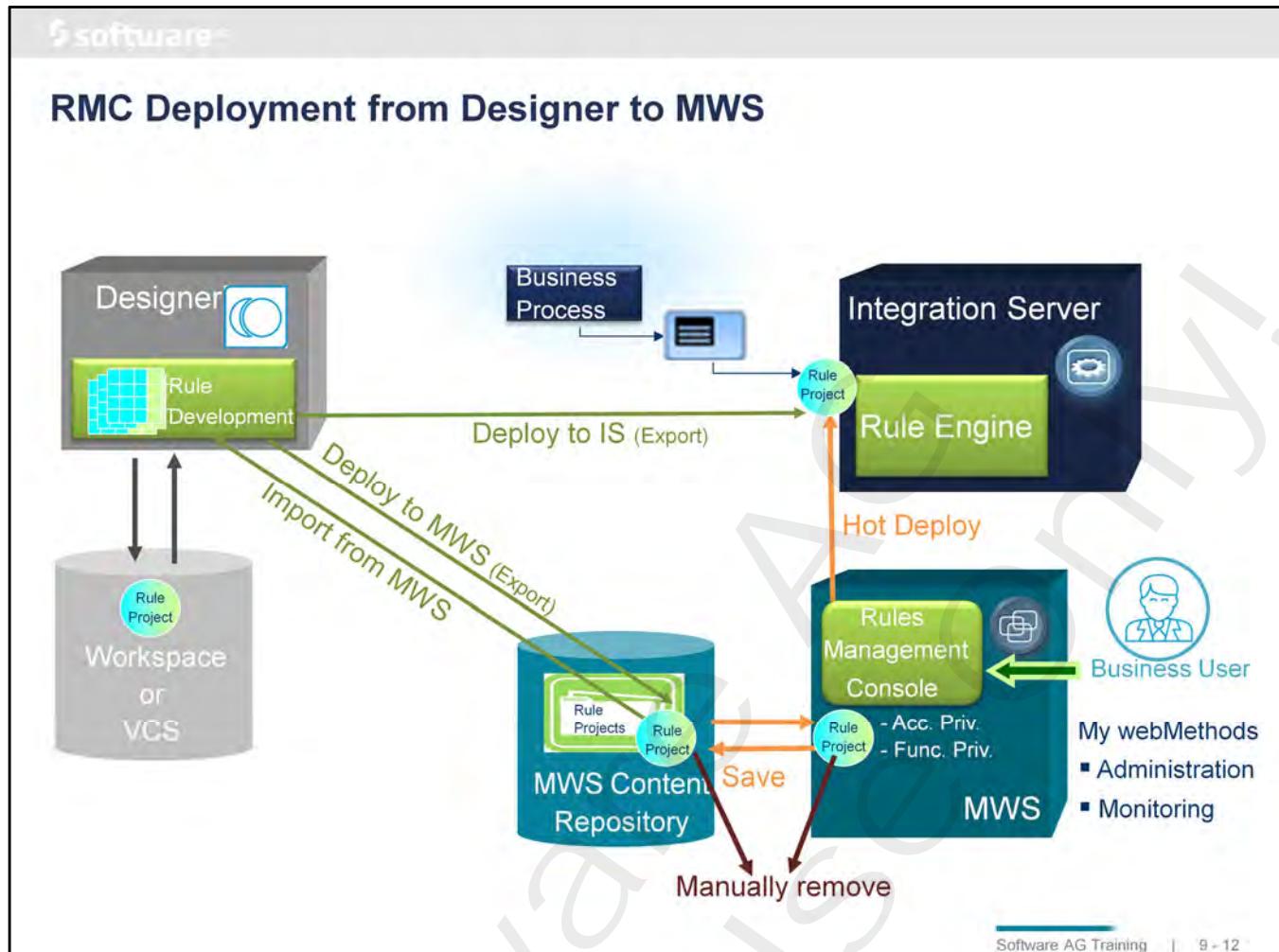
- Use (internal) IS service:
`wm.businessrules.runtime.archive:undeployProject`

Necessary inputs can be retrieved by running IS service:
`wm.businessrules.runtime.info:getDeployedProjects`

The screenshot shows the webMethods interface. On the left, there is a tree view of a project structure under 'WmBusinessRules'. A node named 'undeployProject' is highlighted with a blue box. On the right, a dialog box titled 'Enter Input for 'undeployProject'' is open. It contains a table with three rows: 'Project Information', 'Project Name' (set to 'OrderRulesProject'), and 'Project Version' (set to '2015-03-10-T11:33:13_M'). A blue arrow points from the text 'Necessary inputs can be retrieved by running IS service:' to this dialog. Another blue box highlights the 'Project Name' row in the table.

Software AG Training | 9 - 11

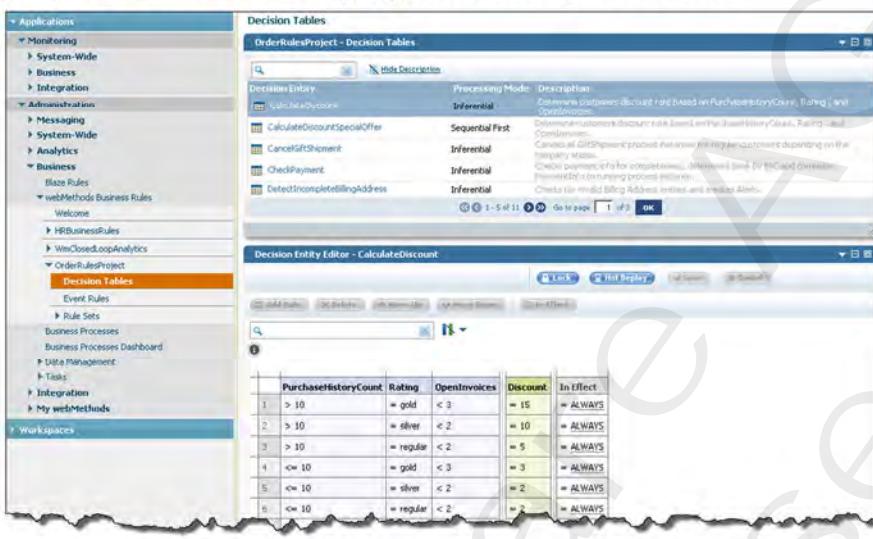
Notes:



Notes:

Rules Management Console (RMC)

- My webMethods offers access to every deployed Rules Management Console (RMC) - one RMC* for each Rule project
- Enables Business Users to dynamically work on rule assets of deployed rule projects without changing business processes



* Rules Management Console in MWS can be monitored and administered using Command Central

Software AG Training | 9 - 13

Notes:

Prerequisites for Using an RMC in My webMethods- Overview

1. MWS security setup for target Rule Projects folder in MWS Content Repository
2. MWS security setup for accessing the RMC and the Rule Projects in My WebMethods
3. Configure MWS Repository Connection in Designer Preferences
4. Deployment of Rule Project from Designer to MWS Content Repository

- See also documentation
"Working with Business Rules in My webMethods"



Notes:

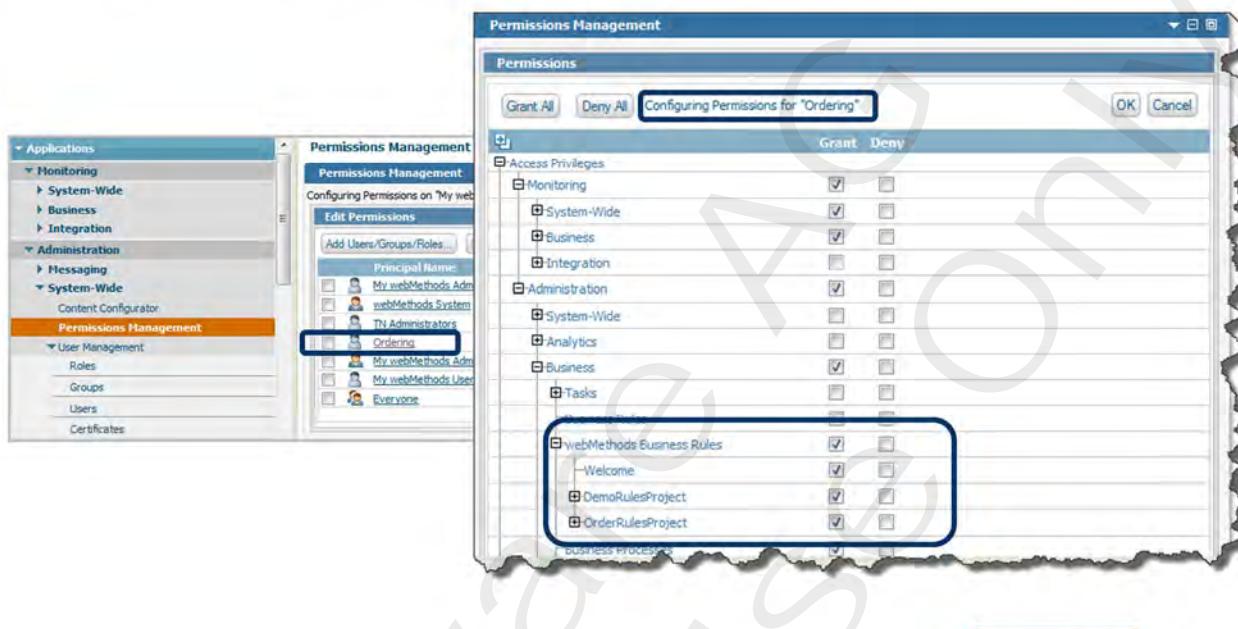
1. MWS Security Setup for Target Rule Projects Folder

- Login to MWS as Sysadmin and grant Permissions for Designer users to deploy Rule Projects into MWS Content Repository folder:
Folders > My webMethods Applications > webMethods Application Data > Rule Projects

Notes:

2. MWS Security Setup for Accessing the RMC

- Login to My webMethods as Administrator. Grant permissions for Business Users to access the RMCs and the Rule Projects in the RMC:



Software AG Training | 9 - 16

Notes:

3. Configure MWS Repository Connection in Designer Preferences

- Create My webMethods Server Content Repository Connection in Designer Preferences

The screenshot shows the 'My webMethods Server Repositories' configuration window. On the left, a tree view lists various software components under 'Software AG'. The 'My webMethods Server Repositories' node is selected and highlighted with a blue border. On the right, a table lists a single repository entry: 'Name' is 'MWS Content Repository', 'Host' is 'localhost', and 'MWS HTTP Port' is '8585'. Below this table is an 'Edit Server' button, which is also highlighted with a blue border. A modal dialog box titled 'Edit My webMethods Server Repository' is displayed over the main window, containing fields for 'Name' (set to 'MWS Content Repository'), 'Host' (set to 'localhost'), 'Port' (set to '8585'), 'User' (set to 'Administrator'), and 'Password' (represented by five asterisks). At the bottom of the modal are 'OK' and 'Cancel' buttons.

Notes:

4. Deployment of Rule Project to MWS Content Repository

- Rules Projects could be deployed to an MWS Content Repository from Designer's:
 - File menu
 - Rules Explorer view
 - Solutions view

Notes:

Software AG

Check MWS Content Repository Deployment using My webMethods

- After an export from Designer, a Rules Management Console becomes available in My webMethods for the corresponding Rule project
- Business Rules with sufficient permissions can access the Rules Management Console(s) via the Navigation pane
- If necessary, refresh the entries in Navigation pane first



The screenshot shows the MY WEBMETHODS interface. The navigation pane on the left is titled 'Applications' and contains the following structure:

- Monitoring
- Administration
 - Messaging
 - System-Wide
 - Analytics
- Business
 - webMethods Business Rules
 - WmClosedLoopAnalytics
 - OrderRulesProject (highlighted)
 - Decision Tables
 - Decision Trees
 - Event Rules
 - Rule Sets
 - ProviderServiceDemo
 - Business Processes
 - Data Management
 - Tasks
 - Integration
 - My webMethods
- Workspaces

Software AG Training | 9 - 19

Notes:

Optional: Check MWS Content Repository Deployment in MWS

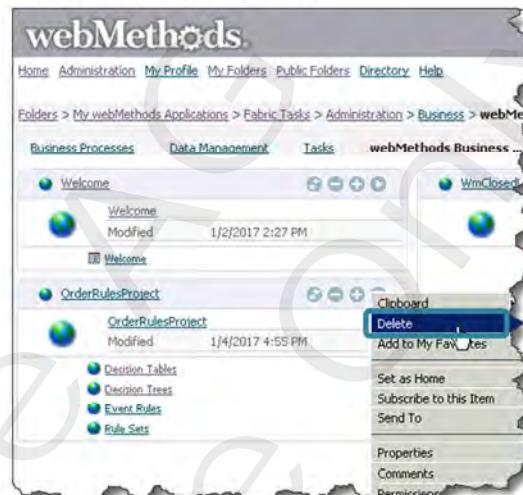
- Deployed Rules projects become visible in Rule Projects folder:
Folders > My webMethods Applications > webMethods Application Data > Rule Projects

The screenshot shows the webMethods application interface. At the top, there is a navigation bar with links: Home, Administration, My Profile, My Folders, Public Folders, Directory, Help, Advanced, Search, and a user dropdown (Hello, Sys Admin! Logout). Below the navigation bar, the main content area shows the 'Rule Projects' section. A breadcrumb navigation path is displayed: Folders > My webMethods Applications > webMethods Application Data > Rule Projects. Under this path, there are two project entries: 'DemoRulesProject' and 'OrderRulesProject'. Each project entry has a thumbnail icon, the project name, a modified date (4/6/2011 3:18 PM for Demo and 4/6/2011 3:20 PM for Order), and a list of sub-folders: Actions, bin, Data Models, Decision Tables, Event Models, Event Rules, and Rule Sets.

Notes:

Undeployment from MWS...

- Has to be performed manually (so far):
 - Logon to MWS as **Sysadmin** for MWS Content Administration
 - Navigate to:
Folders >
My webMethods Applications >
Fabric Tasks >
Administration >
Business >
webMethods Business Rules
 - Delete Rule Project entry



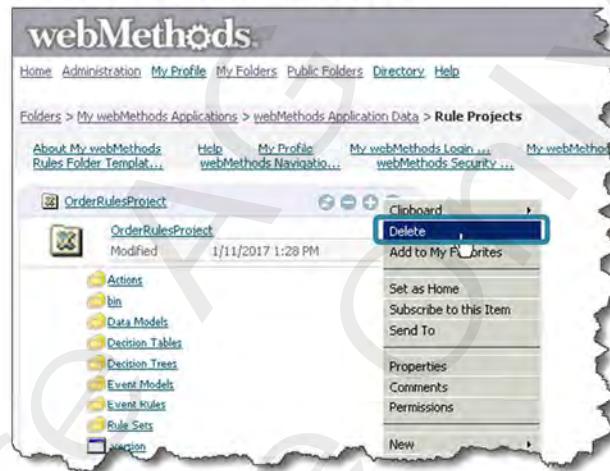
→ Removes Rule project entry from Navigation tree in My webMethods

Software AG Training | 9 - 21

Notes:

...Undeployment from MWS

- Has to be performed manually (so far):
 - Still in MWS as user **Sysadmin** for MWS Content Administration
 - Navigate to:
Folders >
My webMethods Applications >
webMethods Application Data >
Rule Projects
 - Delete Rule project entry



→ Removes Rule project content from MWS Content Repository

Notes:



Exercise 9

- In this exercise, you will deploy your rules project to your Integration Server first. Then, by using a configured MWS Repository connection, you will deploy your Rule Project to be used by the Rules Management Console (RMC) in My webMethods. This requires configuration of several MWS security settings in advance.

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



10

Working with the Rules Management Console

Notes:

Objectives

At the end of this chapter you ...

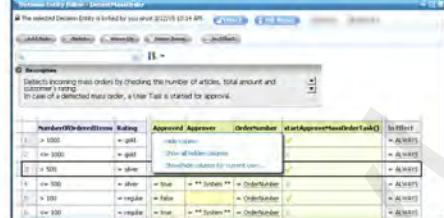
- Logged into the Rules Management Console in My webMethods
- Used the RMC for instant modifications and verifications of Decision Entities
- Saved and “hot-deployed” Rule assets from the RMC to the IS runtime environment
- Synchronized modified Rule Projects back to Designer

Notes:

Rules Management Console (RMC) in My webMethods

- Intended for Business Users to perform web-based instant rule modifications using My webMethods
- RMC offers different capabilities compared with Designer's Rules Development perspective:
 - User can lock/unlock to protect/unprotect a Decision Entity
 - User can add*/delete* rules, modify and save the content of a Decision Entity
 - User can invoke a custom Data Provider Service to dynamically provide cell values*
 - User can modify the sequence of rules in a Decision Entity*
 - User can perform verifications**
 - User can invoke a custom Rule Verification service for column values*
 - User can set the "In effect" property for single rules or entire Decision Entity
 - User can hot deploy a Rule Project to IS runtime(s)
 - User can change the structure of a Decision Entity
 - User can create new or delete existing Decision Entities
- How to work with an RMC is described in the document "Working with Business Rules in My webMethods "

* currently not supported for Decision Trees in the RMC
 ** currently not supported for Decision Trees in Designer and RMC



RMC only
Designer only
RMC and Designer

Notes:

Rules Management Console - Navigation

- For a Rule Project deployed to MWS, Decisions Entities of type **Decision Table**, **Decision Tree**, **Event Rule**, and **Rule Set** become available in the left-hand Navigation pane
- Selecting a Decision Entity type opens a page listing all entries of that type in a portlet

Decision Entity	Processing Mode	Description
CalculateDiscount	Inferential	Determine customers discount rate based on OpenInvoices.
CalculateDiscountSpecialOffer	Sequential First	Determine customers discount rate based on OpenInvoices.
CancelGiftShipment	Sequential First	Cancels all gift shipment process instances for regular customers depending on the company status.
CheckPayment	Sequential First	Checks PaymentInfo for completeness, determines Bank by BIC, and correlates PaymentInfo to running process instance.
DetectIncompleteBillingAddress	Sequential First	Checks for invalid billing address entries and creates alerts.

Software AG Training | 10 - 4

Notes:

Decision Entity Editor for a Decision Table

- Lock Decision Table before modifying

The screenshot shows the 'Decision Entity Editor' interface for a decision table. The table has columns: NumberofOrderedItems, Rating, Approved, Approver, OrderNumber, and startApproveMassOrderTask(). The rows define rules based on the number of ordered items:

NumberofOrderedItems	Rating	Approved	Approver	OrderNumber	startApproveMassOrderTask()	In Effect
1 > 1000	= gold	= false			= ** System ** = OrderNumber	= ALWAYS
2 <= 1000	= gold	= true			= OrderNumber	= ALWAYS
3 > 500	= silver	= false			= ** System ** = OrderNumber	= ALWAYS

Annotations on the interface include:

- Append new last rule (points to the 'Add Rule' button)
- Delete/ move selected rows(s) (points to the 'Delete' and 'Move Up/Down' buttons)
- Lock/unlock Decision Table (points to the 'Unlock' button)
- Minimize/ maximize portlet (points to the minimize/maximize button in the top right)
- Restore portlet (points to the restore button in the top right)
- Hide/show description (points to the 'x' icon in the description section)
- Modify description text (points to the 'Description' text area)
- Click to hide/unhide columns (points to the column visibility dropdown menu)
- Click to edit value (points to the edit icon in the 'Approved' column of the third row)
- Open Advanced Edit Dialog (points to the edit icon in the 'Approved' column of the third row)

Notes:

Support for Large Decision Tables in Decision Entity Editor

- Pagination
 - Improves displaying and handling large Decision Tables
 - Every page shows 100 lines of the Decision Table
 - The pagination significantly speeds up the initial display of large Decision Tables

- Filtering
 - Providing a filter value only shows rules containing the filter value



Row	Condition	Action	Result
88	= 00088	= 04420	= kunde88
89	= 00089	= 04420	= kunde89
90	= 00090	= 04420	= kunde90
91	= 00091	= 04420	= kunde91
92	= 00092	= 04420	= kunde92
93	= 00093	= 04420	= kunde93
94	= 00094	= 04420	= kunde94
95	= 00095	= 04420	= kunde95
96	= 00096	= 04420	= kunde96
97	= 00097	= 04420	= kunde97
98	= 00098	= 04420	= kunde98
99	= 00099	= 04420	= kunde99
100	= 00100	= 04420	= kunde100
			= bearbeiter88 = vertreter88 = ALWAYS
			= bearbeiter89 = vertreter89 = ALWAYS
			= bearbeiter90 = vertreter90 = ALWAYS
			= bearbeiter91 = vertreter91 = ALWAYS
			= bearbeiter92 = vertreter92 = ALWAYS
			= bearbeiter93 = vertreter93 = ALWAYS
			= bearbeiter94 = vertreter94 = ALWAYS
			= bearbeiter95 = vertreter95 = ALWAYS
			= bearbeiter96 = vertreter96 = ALWAYS
			= bearbeiter97 = vertreter97 = ALWAYS
			= bearbeiter98 = vertreter98 = ALWAYS
			= bearbeiter99 = vertreter99 = ALWAYS
			= bearbeiter100 = vertreter100 = ALWAYS

Software AG Training | 10 - 6

Notes:

Hide/Show Decision Table Columns in Decision Entity Editor

- Decision Entity Editor allows to show/hide columns
 - in general
 - for a particular MWS principal

Right-click to hide/show columns

Press here to open options

Select "Show/hide columns"

Select columns to hide

Notes:

Decision Entity Editor for a Decision Tree

- Lock Decision Tree before modifying
- Saving performed by inline or advanced editor

The screenshot shows the 'Decision Entity Editor - PropagateAlerts' interface. At the top right are 'Minimize/maximize portlet' and 'Restore portlet' buttons. Below them are 'Unlock' and 'Hot Deploy' buttons. The main area shows a decision tree node with a 'Description' field containing placeholder text. There are 'Expand All' and 'Collapse All' buttons. A 'Condition Node Configuration' dialog is open, showing fields for 'Label' (Medium OrderNo) and 'Default Value'. Other visible nodes include 'AlertType' (medium) and 'OrderNo'. Annotations with callouts explain various features:

- 'Collapse/ expand all nodes'
- 'Modify description text'
- 'Lock/unlock Decision Tree'
- 'Hide/show description'
- 'Click to edit a condition value'
- 'Click to edit a Condition Node Configuration'
- 'Open Advanced Edit Dialog'

Software AG Training | 10 - B

Notes:

Decision Entity Editor for an Event Rule

- In the RMC, Event Rules can be locked, modified, verified, saved and unlocked like Decision Tables and Decision Trees
- Advanced Edit Dialog available for resulting values

The screenshot shows the 'Decision Entity Editor - SynchronizeShippingAddress' interface. At the top, there's a message about the entity being locked by the user. Below it are buttons for 'Unlock', 'Hot Deploy', and others. A 'Description' section contains a tooltip for 'Hide/show description'. To the right are buttons for 'Minimize/maximize portlet', 'Lock/unlock Event Rule', and 'Restore portlet'. The main area shows an 'Event' section with a 'City' field set to 'changed'. Below it is a 'Results' table with six rows, each showing a variable name and its mapping. A tooltip for the first row says 'Click to edit a result value'. The table has a green header row.

Notes:

Decision Entity Editor – Advanced Edit Dialog

- opens Open Advanced Edit Dialog for modifications of
 - Condition Values or Result Values (Decision Table, Event Rule)
 - Links (Decision Trees)

Available operators, depending on data type

The screenshot shows the 'Condition Value Modification' dialog box. It has fields for 'Select an operator*' (with '<=' selected) and 'Select an option*' (with 'Expression' selected). Below these are tabs for 'Functions' and 'Parameters'. A large list of methods is displayed under the 'Integer' category, including: appendToList(Object[] list, Object item), century(): int, compareDates(Date date1, Date date2): int, createList(Object[] list): Object[], dayOfMonth(): int, dayOfWeek(): int, dayOfYear(): int, daysInMonth(int month, int year): int, inRange(Long lowerBound, Long upperBound): boolean, and inRange(Long lowerBound, Long upperBound): int. At the bottom are 'Finish' and 'Cancel' buttons.

Notes:

"In Effect" Dates

- You can specify an "In Effect" date for*:
 - Decision Tables
 - Decision Trees
- "In Effect" is specified only in RMC
 - "In Effect" date can be later viewed, but not changed in Designer**
- By default, the rules of a Decision Table/Tree are always in effect

The screenshot shows the Decision Entity Editor for a Decision Table named DT_EffectivenessGeneral. The interface includes tabs for Add Rule, Details, and Verification Portlet. A button labeled 'In Effect' is visible. The main area displays a grid of rules with columns for name, dateInput, result, and In Effect. The In Effect column contains entries like '= ALWAYS', '= NEVER', and ranges such as '< 9/8/14 12:00 AM' and '> 11/13/14 12:00 AM'. A blue callout bubble points to the In Effect column with the text "'In Effect' column/node is always visible'. Another blue callout bubble points to the grid with the text 'Expired "In Effect" cells are marked in yellow and listed in the verification portlet'.

* In Effect dates not available for Event Rules
 ** currently "In Effect" dates of Decision Trees are not displayed in Designer

Software AG Training | 10 - 11

Notes:

Decision Entity Editor in RMC – Set "In Effect" Dates

- For Decision Entities of type Decision Table and Decision Tree “In Effect” Dates can be specified for entire Decision Entity or at rule level
- My webMethods user must have “Add Rule” privileges to specify “In Effect” dates

The screenshot shows the 'Decision Entity Editor - DetectMassOrder *' interface. On the left, a modal dialog titled 'Edit in effect dates' is open, showing options for setting in-effect dates: 'Always', 'Never', or 'Time frame'. A callout bubble says 'Select - Always - Never - Time frame for "In Effect"' and points to the 'Time frame' option. Another callout bubble says 'Set "In Effect" dates of all rules' and points to the 'In Effect' button in the dialog. On the right, a decision table named 'orderTask()' is displayed. The 'In Effect' column contains rules like '>= 3/13/15 12:00 AM', '>= 3/13/15 12:00 AM', and '>= 3/13/15 12:00 AM'. A callout bubble says 'Set "In Effect" dates of selected row/rule' and points to one of these rules.

Notes:

"In Effect" Dates in Designer

- After saving the Decision Entity in the RMC and re-importing the rule project from MWS into Designer, "In Effect" dates are displayed in Designer*
- "In Effect" dates can be viewed, but not changed in Designer

The screenshot shows the webMethods Business Rules Designer interface. On the left, there's a decision table with columns for name, dateInput, and result. A tooltip over the first row indicates the "In Effect" time frame as Sep 1, 2014 12:00:00 AM <...<= Sep 4, 2014 12:00:00 AM. The "Properties" tab is selected, showing a list of verification categories: Gaps, Overlaps, Syntax, Empty cells (115), Processing..., Redundant..., Missing Rule..., and Other (2). The "Problems" tab shows two rule warnings: "The rule has expired. [RUL.0070.1850]" and "The rule has expired. [RUL.0070.1850]". A callout box points to this list with the text: "Expired rules are shown in the 'Other' category of Rule Verifications".

Clock indicates that "In Effect" has been used in this Decision Entity

"In Effect" time frames is shown as a tooltip by hovering over the row number, but cannot be edited

Expired rules are shown in the "Other" category of Rule Verifications

* currently "In Effect" dates of Decision Trees are not available in Designer

Software AG Training | 10 - 13

Notes:

Rules Management Console – Rule Sets

- A Rule Set in the RMC lists all contained Decision Entities
- Each Decision Entity can be opened, modified and verified from here, too

	Country	ZipCode	City	In Effect
1	= DE	= 64297	= Darmstadt	= ALWAYS
2	= DE	= 80331	= Munich	= ALWAYS
3	= US	= 20199	= Reston	= ALWAYS
4	= US	= 80201	= Denver	= ALWAYS
5	= MY	= 50470	= Kuala Lumpur	= ALWAYS
6	= MY	= 62000	= Putrajaya	= ALWAYS

Click Decision Entity to open it in the lower Decision Entity Editor portlet

Software AG Training | 10 - 14

Notes:

Rules Management Console - Rules Verification

Decision Entities with errors or warnings are marked

Decision Entity Editor - CancelGiftShipment

	GlobalCustomerSatisfaction	MonthlyTurnaround	StockRating
1	= well	100000.0 <=...< 500000.0	= well
2		< 100000.0	= well
3	= well		= bad
4	= bad		

In Effect

cancelGiftShipmentForRegularCustomers() In Effect
= ALWAYS
= ALWAYS
= ALWAYS
= ALWAYS

Apply filters

Press to Re-verify

OrderRulesProject - CancelGiftShipment - Verification

Filter by: All Errors/Warnings CancelGiftShipment.decisionable

Category	Severity	Description	Resource	Location
Gaps				
Overlaps	⚠	Overlap: 100000.0 <= CompanyStatus_1.MonthlyTurnaround < 500000.0 and CompanyStatus_1.MonthlyTurnaround >= 100000.0. [RUL.0070.0020]	CancelGiftShipment.decisionable	
Syntax				
Empty cells				
Processing Modes				
Redundancies				

Hide/show warnings suppressed in Designer

Show all suppressed warnings

Software AG Training | 10 - 15

Notes:

Business Validation using a Verification Service

- Verify the current values provided in a Decision Table column before save and hot deploy
- Requires for a REST service assigned to a condition or assignment column in Designer and configured in the RMC

The screenshot shows two windows. The top window is the 'Decision Entity Editor - CustomerStatus' showing a decision table with columns 'age', 'zipcode', and 'status'. The bottom window is the 'BusinessVerification - CustomerStatus - Verification' tool. A callout from the verification tool points to the 'zipcode' column in the decision table, stating: 'Zipcode values verified by a custom REST service'. Another callout from the verification tool says: 'I want to allow valid ZIP codes only.' A user icon is also present in the verification tool.

Notes:

Setting Up a Verification Service in Designer and My webMethods

1. Create a REST resource (e.g. in IS) implementing the Verification Service*

The screenshot shows the Outline view with a tree structure. A blue callout points to a node under 'RulesSupport' labeled 'zipCode'.

2. In My webMethods, define and enable base REST URL for RMC**

The screenshot shows the 'My webMethods' settings page. A blue callout points to the 'Business Verification' section where 'Enabled' is checked and the 'Protocol' is set to 'http'. The 'Base Path' is set to 'rest' and the 'Url' is 'http://localhost:5555/rest/services/'.

3. In Designer, add and configure Verification Service for Decision Table column

The screenshot shows a decision table with columns 'Age', 'Rating', 'ZipCode', and 'Discount'. A blue callout points to the 'Configure Verification Service' option in the context menu for the 'Discount' column.

4. Save and deploy Rule Project to IS and MWS

The screenshot shows the 'Business Verification Configuration' dialog with the 'Service URI' field containing 'rulesSupport/RESTprovider/zipCode'. A blue callout points to the 'OK' button.

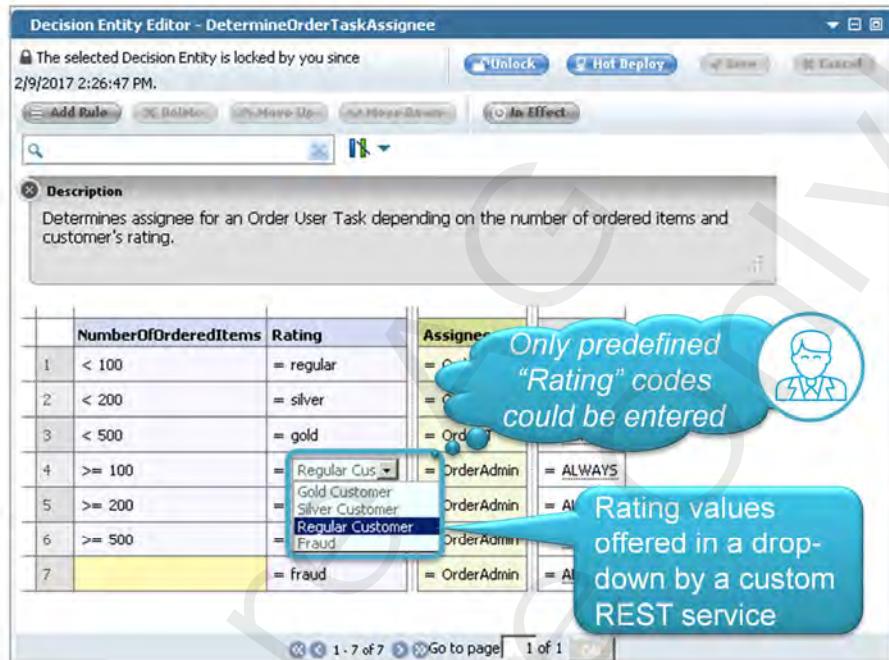
* For more details about implementing a Verification Service see document "webMethods Business Rules Reference"
** Base path of Business Verification REST URL can be also configured using Command Central

Software AG Training | 10 - 17

Notes:

Data Provider Service for Decision Tables

- To avoid later errors, only predefined values are offered in the RMC when modifying an condition or assignment value of a Decision Table
 - Requires for a REST service assigned to the column in Designer and configured in the RMC



Software AG Training | 10 - 18

Notes:

Setting Up a Data Provider Service in Designer and My webMethods

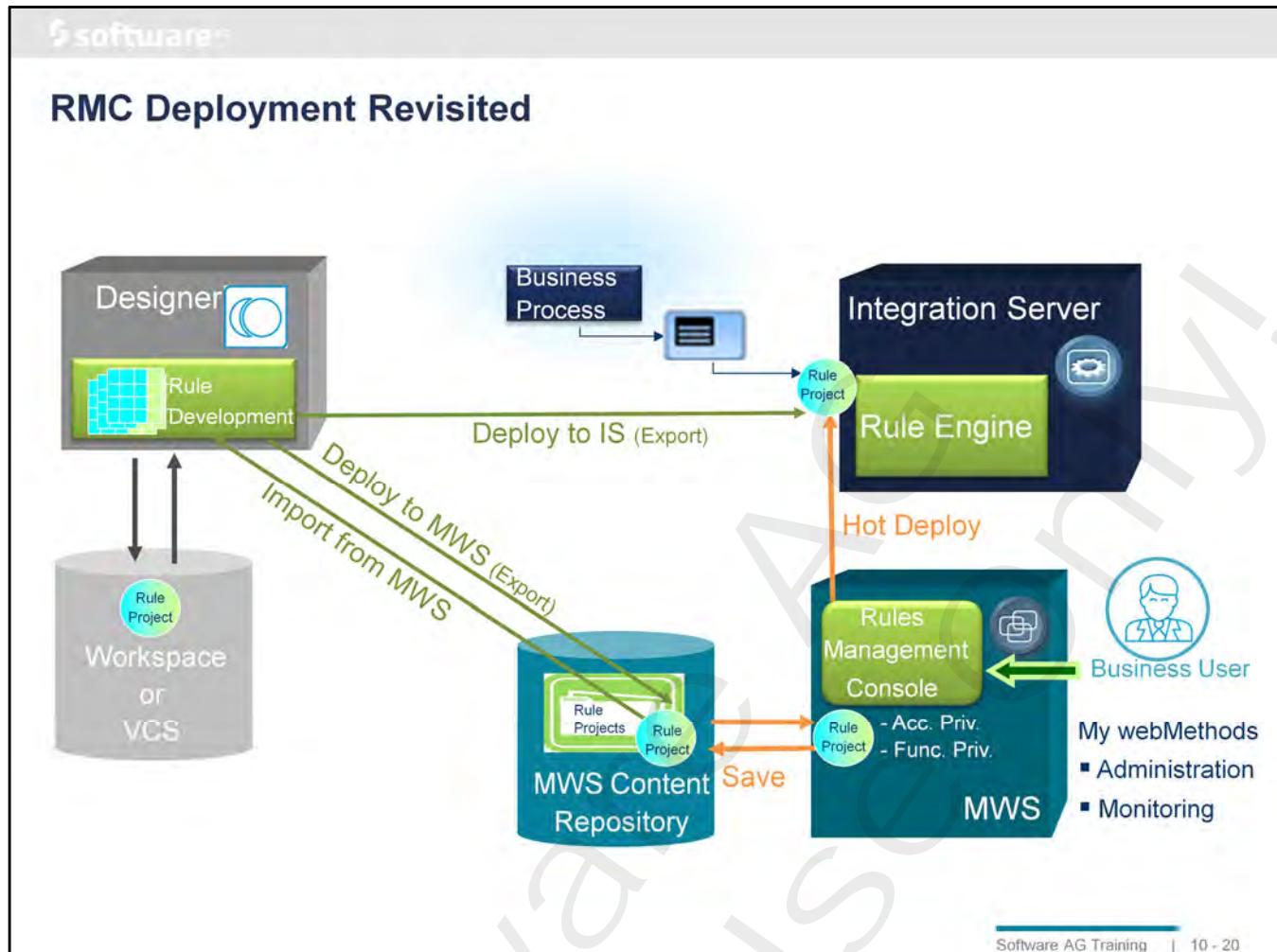
The screenshot shows the webMethods Business Rules interface with several windows open:

- Outline View:** Shows a tree structure with nodes like 'Definitions', 'RulesSupport', 'RESTprovider', and 'rating'. A callout bubble labeled "1. Create a REST resource (e.g. in IS) implementing the Data Provider Service*" points to the 'rating' node.
- Applications Menu:** Shows options like Monitoring, Business, Integration, and My webMethods.
- My webMethods - webMethods Business Rules Settings:** A configuration window for the DATA PROVIDER. It has checkboxes for 'Enabled' and 'Protocol' (set to http), and fields for 'Host' (localhost), 'Port' (5555), 'Base Path' (rest), and 'Url' (http://localhost:5555/rest/service). A callout bubble labeled "2 In My webMethods**, define and enable base REST URL for RMC" points to the 'Url' field.
- Data Provider Service Configuration Dialog:** A modal dialog titled "Data Provider Service Configuration" with the instruction "Enter a REST service path.". It contains a text input field with the value "Service URL: rulesSupport/RESTprovider/rating". A callout bubble labeled "3 In Designer, add and configure Data Provider Service for Decision Table column" points to this dialog.
- Decision Table View:** Shows a table with columns "NumberOfOrderedItems" and "Rating". Rows 1 and 2 have conditions like "< 100" and "< 200" respectively, with corresponding actions like "regular" and "silver". A callout bubble labeled "3 In Designer, add and configure Data Provider Service for Decision Table column" points to this table.
- Buttons at the Bottom:** Buttons for "Add Rule", "Add Condition", "Add Assignment", "Add Action", "Configure Verification Service", and "Configure Data Provider Service". A callout bubble labeled "4 Save and deploy Rule Project to IS and MWS" points to the "Configure Data Provider Service" button.

* For more details about implementing a Data Provider Service see document "webMethods Business Rules Reference"
** Base path of Data Provider REST URL can be also configured using Command Central

Software AG Training | 10 - 19

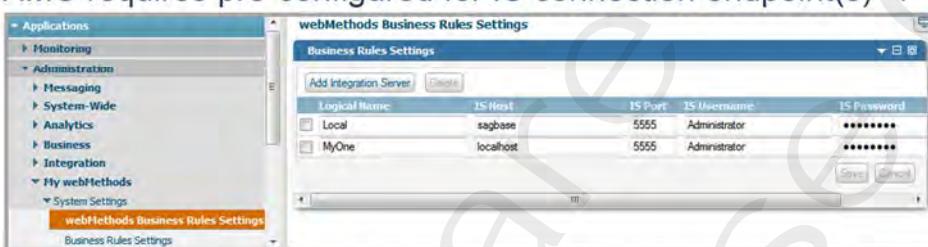
Notes:



Notes:

Decision Entity Editor in RMC – Unlock, Save, and Hot Deploy

- **Save***: Persists changes in MWS Content Repository
 - No automatic sync-back with Designer workspace/VCS
 - Requires for a later import from MWS Content Repository into Designer
 - Always save before unlock, otherwise changes get lost
- **Unlock**: Removes the lock for Decision Entity
- **Hot Deploy**: Deploys Rule Project from RMC to one or multiple IS runtimes
 - Immediate project replacement in IS runtime
 - RMC requires pre-configured for IS connection endpoint(s)**:



* Only available for Decision Table and Event Rule. Modifications at Decision Trees are saved when leaving the node editor
** Hot Deploy endpoints can also be configured using Command Central

Software AG Training | 10 - 21

In case of using a clustered IS, the Integration Server endpoint(s) to be defined under webMethods Business Rules Settings are so called master IS(s). Using a database table, a master IS knows its slave ISs in its cluster and internally distributes the hot deployed Rule project to all known slaves in the cluster.

Re-Deployment of Rule Project from MWS Content Repository

- Rules Projects modified by the RMC could be imported to overwrite the outdated Rule Project still in Designer. Use Import from Designer's:
 - File menu
 - Rules Explorer view
 - Solutions view

The screenshot shows the webMethods Business Rules interface. On the left, there is a navigation pane with 'Solutions' and 'Navigator' tabs, and a main workspace with various project types like 'Default Solution', 'Dynamic Processes', 'Processes', 'Rules', etc. A context menu is open over the 'Rules' node, with 'Import...' selected. In the center, two windows are displayed: 'Import' (Select) and 'Import Rule Project'. The 'Import' window lists options like 'ARIS Solution design', 'CloudStreams Governance Project', etc., and 'Rule Project files from My webMethods Server repository' is highlighted. The 'Import Rule Project' window shows 'My webMethods Server' set to 'MWS Content Repository' and 'Remote rule project' set to 'OrderRulesProject'. Other projects like 'OrderRulesProject', 'WmClosedLoopAnalytics', and 'ProviderServiceDemo' are listed below.

Software AG Training | 10 - 22

Notes:

Rules Management Console - REST API...

- Functionality of a Rules Management Console can also be invoked via REST calls
- Available predefined REST services:

REST SERVICE	DESCRIPTION	REST SERVICE	DESCRIPTION
GET<base URL>/projects	Returns a list of rule projects that are currently available on My webMethods Server.	PUT<base URL>/project/%rule ProjectName%/decisiontable/%decisionTableName%/unlock	Unlocks the given decision table.
GET<base URL>/project/%rule ProjectName%	Returns the content and metadata for a given rule project.	POST<base URL>/project/%rule ProjectName%/deploy	Hot deploys the given rule project to an Integration Server that was configured on the My webMethods Server.
PUT<base URL>/project/%rule Project Name%/lock	Locks the given rule project.	GET<base URL>/project/%rule Project Name%	Retrieves the rule project from the My webMethods Server.
PUT<base URL>/project/%rule Project Name%/unlock	Unlocks the given rule project.	PUT<base URL>/project/%rule Project Name%/unlock	Stores the given rule project on the My WebMethods Server
GET<base URL>/project/%rule ProjectName%/decisiontable/%decisionTableName%	Retrieves the given decision table.	DELETE<base URL>/project/%rule ProjectName%	Deletes the given rule project from the My WebMethods Server
PUT<base URL>/project/%rule ProjectName%/decisiontable/%decisionTableName%	Writes the changes for the given decision table to the My webMethods Server.		
PUT<base URL>/project/%rule ProjectName%/decisiontable/%decisionTableName%/lock	Locks the given decision table.		

Software AG Training | 10 - 23

Notes:

Rules Management Console - REST API

- For detailed information on available REST services, see the WADL URL:
`<MWS_host>:<MWS_port>/wm_rma/rest/application.wadl`
- All REST services use the base URL:
`<MWS_host>:<MWS_port>/wm_rma/rest/content`
- Packaged within feature “Business Rules Public API”
 - “Business Rules Add-ons” license is needed (WOKAA)
- For further details see document:
[*"webMethods Business Rules Reference"*](#)



Software AG Training | 10 - 24

Notes:



Exercise 10

- In this exercise, you will use the Rules Management Console (RMC) in My webMethods to modify your deployed Rule Project as a Business User. You will change a Decision Table and a Decision Tree and see the advantage of a Data Provider Service. Finally you will hot deploy the entire Rule Project to your Integration Server runtime, and synchronize your Rule Project with Designer.

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



11

BPM invokes Decision Entities



Notes:

Objectives

At the end of this chapter you ...

- Can explain the interaction between Business Processes and Business Rules
- Invoked deployed Decision Entities and Rule Sets from a Rule Task Activity in a Business Process
- Tested and ran the invoking Business Process

Notes:

Using Business Rules in the Digital Business Platform

The diagram illustrates four methods for integrating Business Rules into the Digital Business Platform:

- From a Business Process:** Shows a BPMN-like diagram where a "Decision Entity" leads to two parallel paths: "Offer" and "Offer". Each path contains a "DecisionTable" activity.
- From an Event source (BPM or other):** Shows a screenshot of a BPMN interface with two event definitions: "OrderReceived(1)" and "CreateOrder(InvoiceId:DocumentId)".
- For a User Task Assignment:** Shows a screenshot of a rule table titled "Discount | HandledBy" with the following data:

Discount	HandledBy
= 20%	ITP Team
= 10%	StandardTeam
= 15%	ITP Team
= 5%	StandardTeam
- From Integration Server:** Shows a file structure for "WmBusinessRules" under "pub/businessrules/client". It includes "genericinvoke", "invoke", "InvokeInput", and "InvokeOutput" sub-folders.

Below the diagrams are descriptive notes:

- Put a Rule Task Activity step into a Business Process**
- An Event (int./ext.) triggers an Event Rule and starts related Rule processing**
- A Decision Entity is called when a User Task Assignment is performed**
- Call a Decision Entity using a public IS service**

DIGITAL BUSINESS PLATFORM

Software AG Training | 11 - 3

Notes:

webMethods Business Rules invoked by Business Process...

The configuration dialog shows:

- Integration Server Name: Default
- Rule Type: webMethods Business Rule Blaze Rule
- Rule Set Decision Table Decision Tree
- Rule Project Name: OrderRulesProject
- Decision Tree Name: DetectPremiumOrder
- Generated Service Name: [empty]
- Allow parallel execution
- Compensating

- BPMN Rule Task Activity allows synchronous invocation of
 - webMethods Business Rules Decision Entity (Decision Table or Tree)
 - webMethods Business Rules Rule Set
 - Blaze Rule service in IS
- To create a configured Rule Task Activity for webMethods Business Rules:
 - Drag Decision Entity or Rule Set from Rules Explorer onto Task or into whitespace or
 - Use Browse buttons

Software AG Training | 11 - 4

Notes:

...webMethods Business Rules invoked by Business Process

- Data Mapping of Rule Task Input/Outputs to
 - Named input/output parameters of Decision Entity to be invoked
 - Aggregated named input/output parameters of all Decision Entities contained in Rule Set to be invoked
- Use to refresh parameter names from Decision Entity or Rule Set
- ***Important:*** Parameter names MUST match the parameter names defined in Decision Entities (except for parameters checked as "Any")

Software AG Training | 11 - 5

Notes:



Exercise 11 + 12

- In exercise 11, you will enhance a preconfigured process model by adding a Rule Task Activity. The Rule Task Activity has to be configured to invoke your Decision Tree DetectPremiumOrder. You will debug the process to see the returned results of your invoked Decision Tree.
- In exercise 12, you will enhance a preconfigured process model by adding two Rule Task Activities. The first Rule Task Activity has to be configured to invoke your Rule Set CustomerRuleSet, the second Rule Task Activity should invoke your Decision Tree DetectPremiumOrder. You will debug the process to see the returned results of your invoked Rule entities.

Notes:



12

Business Rule interacts with BPM (Process Actions)



Notes:

Objectives

At the end of this chapter you ...

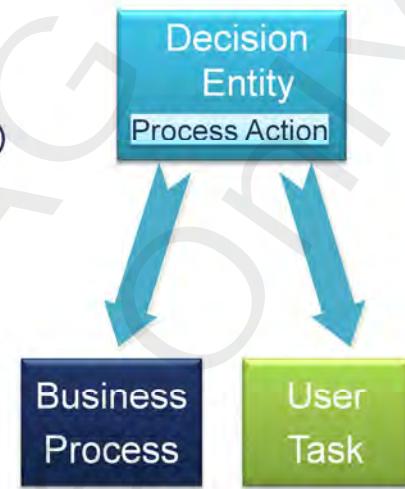
- Can name the various possibilities to interact with Business Processes from a Decision Entity using Process Actions
- Developed Process Actions in Designer
- Started and joined Business Processes using Process Actions
- Canceled, suspended, resumed, and failed running Business Processes using Process Actions
- Dynamically started User Tasks by using Process Actions of type Manual
- Tested Process Actions in conjunction with their related Business Processes

Notes:

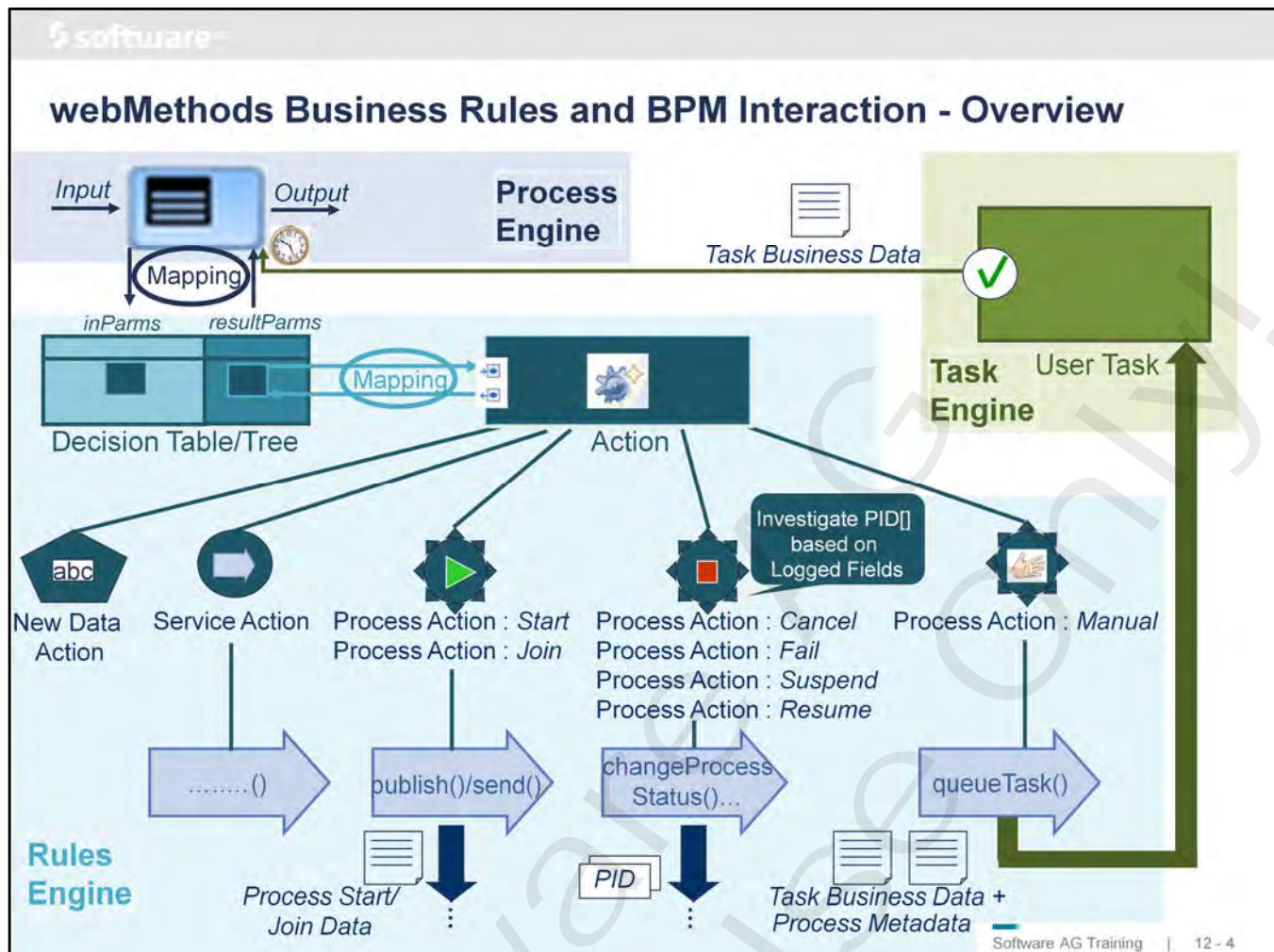
Process Actions

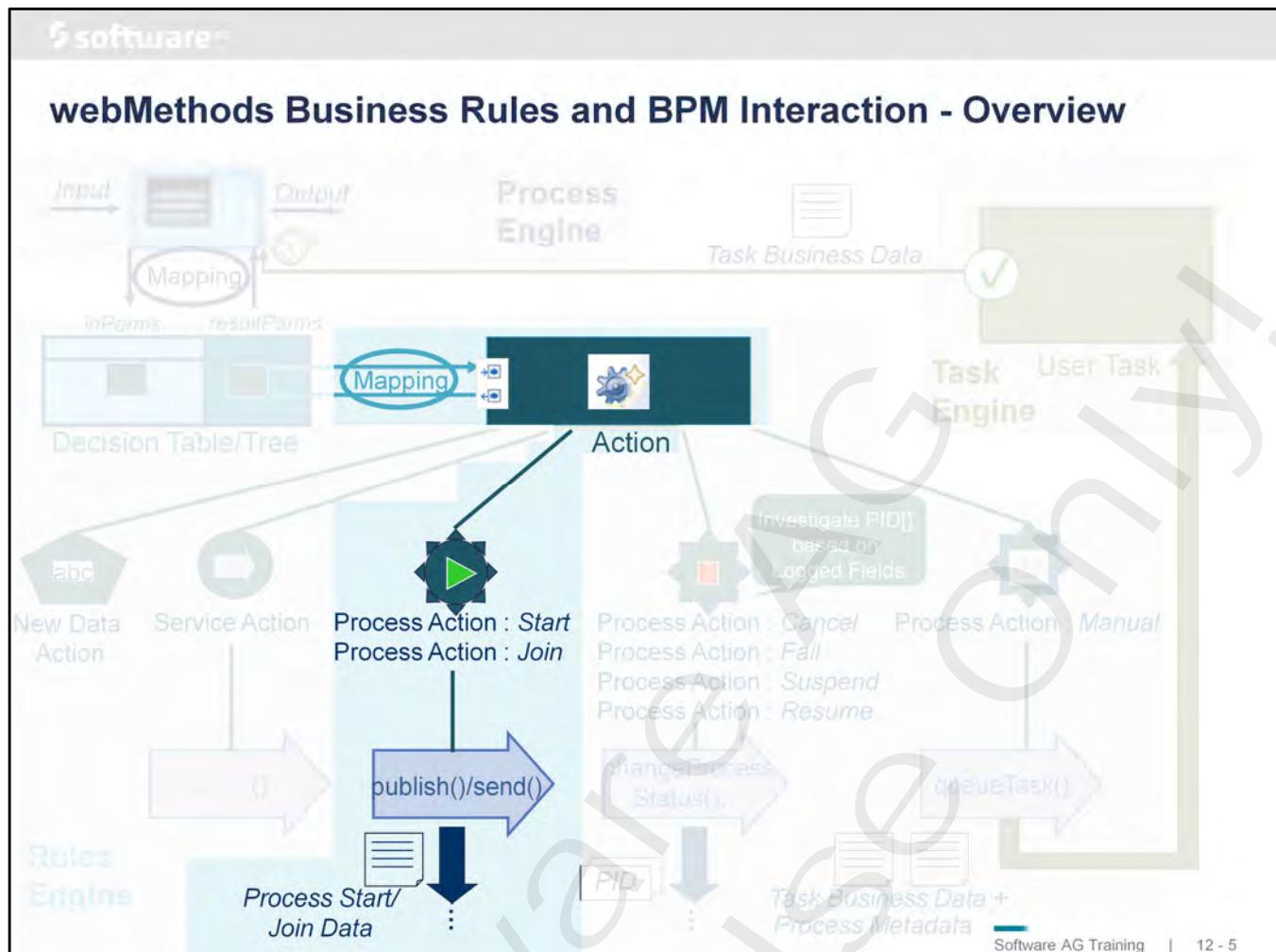
Process Actions enable a Decision Entity to interact with BPM process(es) and User Tasks. You can:

- Start and initiate a new process instance
- Join a running process instance
- Suspend one or more running process instance(s)
- Cancel one or more running process instance(s)
- Fail one or more running process instance(s)
- Resume one or more suspended process instance(s)
- Start and initiate a User Task instance(s)



Notes:





Notes:

Creating a Process Action of Type Start or Join

- Creation of a Process Action supported by wizard
- Add Process Action in Rules Explorer view (*New -> Action*)
- Process Actions require for “Business Rules Add-ons” license (WOKAA)

- Create a Process Action of type Start or Join:
 1. Name, description, Action type (*Process*)
 2. Select Process Action type (*Start or Join*)
 3. Select process model to start or join
 4. Select document type required by the process to start or join
 5. Provide default input for document used to start or join (optional)
 6. Check/uncheck Process Action's output as result



Notes:

1-2: Create a Process Action of Type Start or Join

■ Provide name, rule project and description (optional)

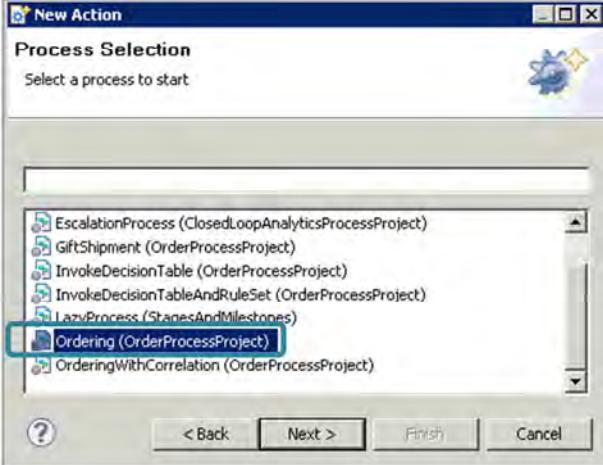
■ Choose type "Process Action"

■ Allow/disallow to execute Process Action multiple times

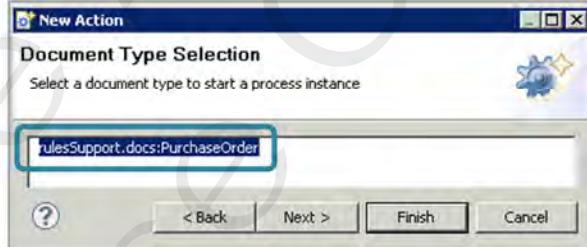
Software AG Training | 12 - 7

Notes:

3-4: Create a Process Action of Type Start or Join



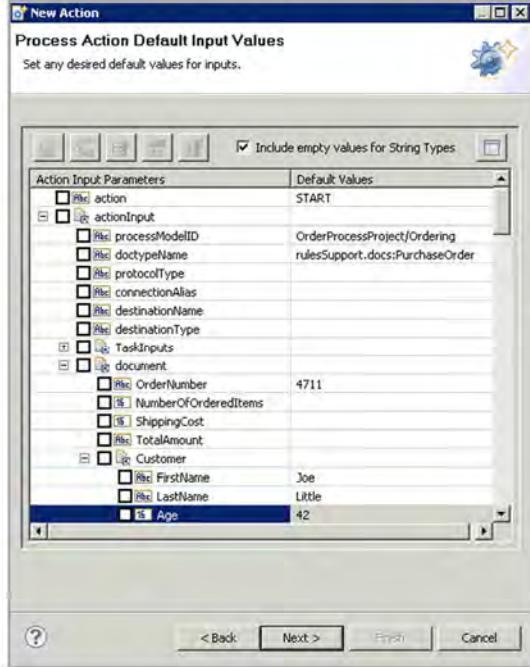
- Select existing process model to start or join
 - Process models retrieved from Designer workspace or VCS



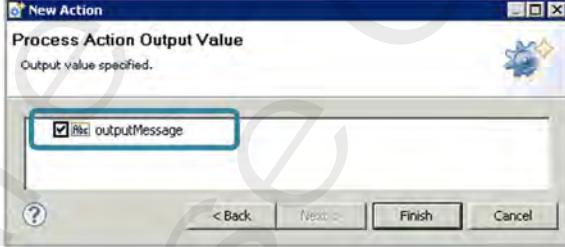
Software AG Training | 12 - 8

Notes:

5-6: Create a Process Action of Type Start or Join



- *Optional:* Provide default input for the document that starts or joins a process instance
- Data for various fields can be static values or dynamic bindings to Data Model elements
- protocolType ("Subscription |JMS") and JMS parameter connectionAlias, destinationName, destinationType are retrieved from the process model's protocol settings



- Check/uncheck action output value

Notes:

Process Actions of Type Start and Join - Remarks

▪ Start:

- When the process action is invoked, action emits the required document by running IS service pub.publish.publish() or pub.prt.jms:send() (depending on process receive protocol), which in turn, starts the process instance
- Currently no support for BPMN Signals

▪ Join:

- When the process action is invoked, action emits the required document by running IS service pub.publish.publish() or pub.prt.jms:send() (depending on chosen protocol) which in turn, is received by the corresponding intermediate Message Event or Receive Task of the process instance to be joined
- Correlation has to be configured in process' receiving process steps
- Currently no support for BPMN Signals

Notes:

Invoke Process Actions of Type Start and Join

- Like other Actions, Process Actions cannot be executed standalone
- Invoke Process Actions from a Decision Entity as
 - a result
 - within an assignment (seldom used)
- Like other Actions, Process Actions can be picked:
 - Within creation wizard for a Decision Table or Event Rule
 - In the Decision Table/ Decision Tree/ Event Rule Editor as a result column
 - In the Extended Cell Editor for an Assignment
- Additionally map Rule Parameter Elements to document fields



see chapter 7

see next slide

Notes:

Invoke Process Actions of Type Start and Join - Mapping

- Map Rule Parameter Elements to document fields:

The 'Map Action Input' dialog displays a tree view of input parameters for a process action named 'PurchaseOrder_1'. The parameters include OrderNumber, NumberOfOrderedItems, ShippingCost, TotalAmount, and Customer. The 'Customer' node has further sub-nodes like AddressLine1 and _env. On the right side of the dialog, there is a 'Map to document' section which contains a tree view of document fields: document, OrderNumber, NumberOfOrderedItems, ShippingCost, and TotalAmount. Arrows indicate the mapping from the process action parameters to the document fields.

- Action's default input values can be overwritten

Software AG Training | 12 - 12

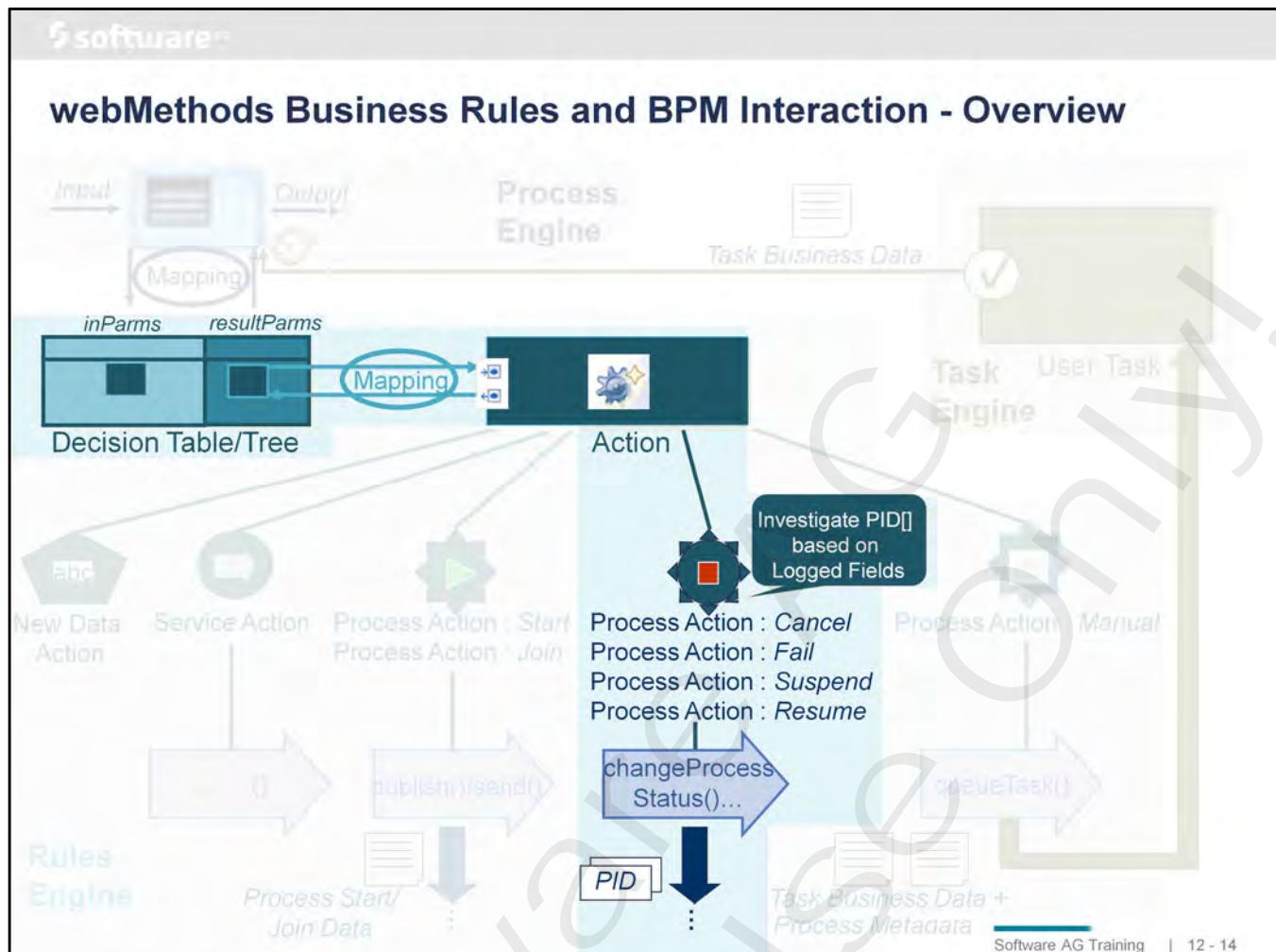
Notes:



Exercise 13 + 14

- In exercise 13, you will create a Process Action to start a process instance by publishing Alert documents. The Process Action will be invoked by a Decision Table catching Alert documents. The process to be started contains a User Task Activity to show and confirm the alert data.
- In exercise 14, you will create a Process Action to join a process instance by publishing a PurchaseOrderPayment document. The action will be invoked by a Decision Table. The process has to be started before you run your Decision Table to wait for the incoming missing PurchaseOrderPayment document.

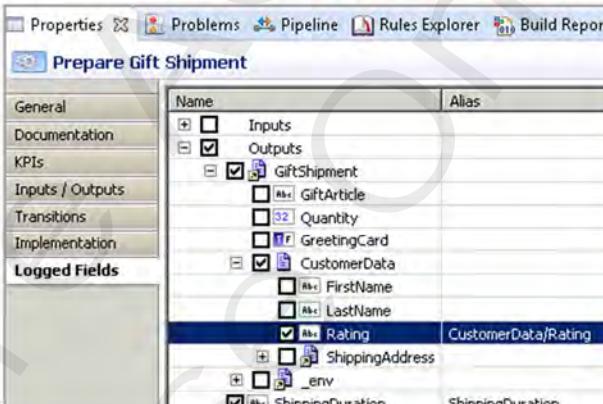
Notes:



Notes:

Process Actions of Type Cancel, Fail, Suspend and Resume

- Process Actions of type Suspend, Cancel, Fail, and Resume require for an IS service retrieving a list of affected ProcessIDs. Service can be:
 - Build-in service
 - Custom service
- Service requires for Logged Fields defined in corresponding process model at any step
 - Logged Field values are logged in additional tables that can be queried
 - Used also by Optimize and KPIs
 - Don't mix up with Process logging level for monitoring and restart – those values cant be queried



Software AG Training | 12 - 15

Notes:

Creating a Process Action of Type Cancel, Fail, Suspend or Resume

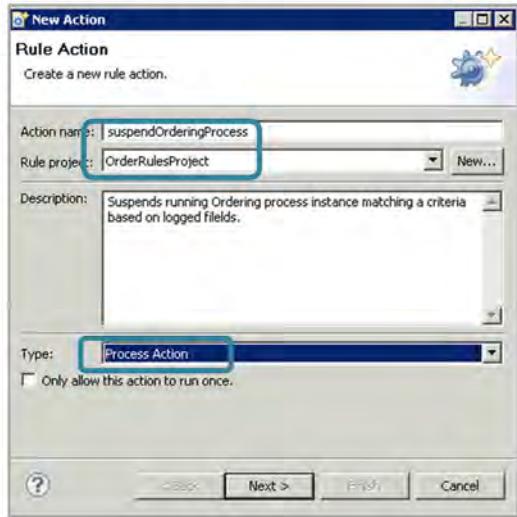
- Creation of a Process Action supported by wizard
- Add Process Action in Rules Explorer view (New -> Action)
- Process Actions require for “Business Rules Add-ons” license (WOKAA)

- Create a Process Action of type Cancel, Fail, Suspend or Resume :
 1. Name, description, Action type (Process)
 2. Select Process Action type (Cancel, Fail, Suspend or Resume)
 3. Select process model to affect
 4. Select service type to derive list of process instances
 5. a) Build-in service: Create search criteria based on Logged Fields
b) Custom service: select custom service
 6. Check/uncheck Process Action's output as result

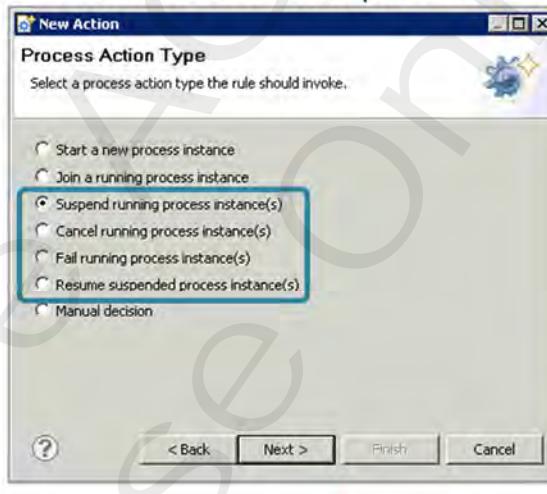


Notes:

1-2: Creating a Process Action of Type Cancel, Fail, Suspend or Resume



- Provide name, rule project and description (optional)
- Choose type "Process Action"
- Allow/disallow to execute Process Action multiple times



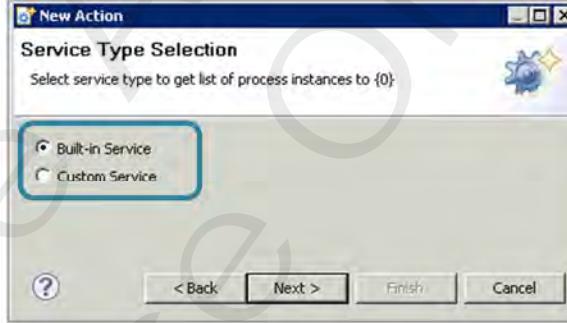
Software AG Training | 12 - 17

Notes:

3-4: Create a Process Action of Type Cancel, Fail, Suspend or Resume



- Select existing process model to affect
 - Process models retrieved from Designer workspace or VCS



- Select service type to derive list of process instances type to affect
 - a) Build-in Service
 - b) Custom Service

Software AG Training | 12 - 18

Notes:

5a: Create a Process Action of Type Cancel, Fail, Suspend or Resume

Build-in Service:
Create search criteria based on Logged Fields of any process step

- Typed Field Names must correspond to Alias names of Logged Fields

Query String:
(NumberOfOrderedItems >= 100 or OrderNumber = 4711)

Properties Problem Map PurchaseOrder

Name	Alias
Inputs	
PurchaseOrder_1	
OrderNumber	OrderNumber
NumberOfOrderedItems	NumberOfOrderedItems
ShippingCost	
TotalAmount	
Customer	
OrderedItems	
_env	

Software AG Training | 12 - 19

Notes:

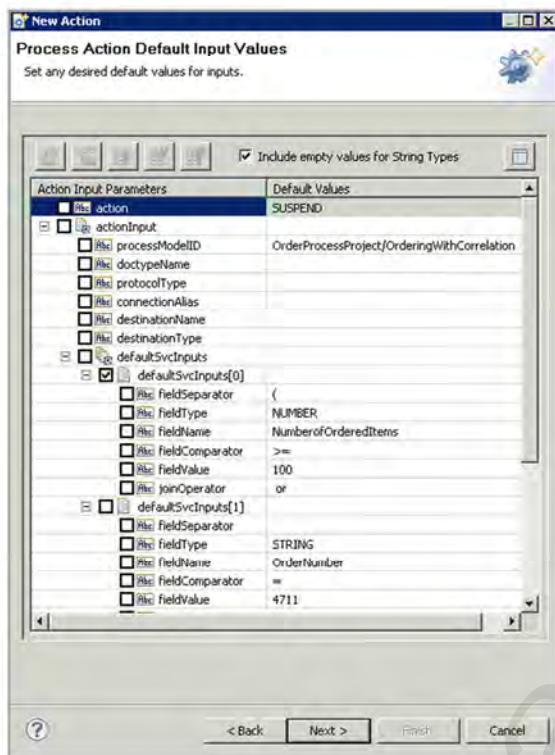
5b: Create a Process Action of Type Cancel, Fail, Suspend or Resume

Custom service:

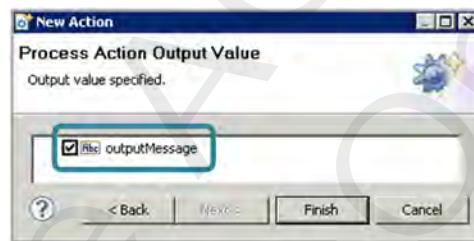
- User-written IS service
- Service has to return this output document

Notes:

6: Create a Process Action of Type Cancel, Fail, Suspend or Resume



- protocolType** (" "**=Subscription |JMS**)
and JMS parameter **connectionAlias**,
destinationName, **destinationType**
are retrieved from the process model's
protocol settings



- Check/uncheck action output value

Software AG Training | 12 - 21

Notes:

Invoke Process Actions of Type Cancel, Fail, Suspend or Resume

- Like other Actions, Process Actions cannot be executed standalone
- Invoke Process Actions from a Decision Entity as
 - a result
 - within an assignment (seldom used)
- Like other Actions, Process Actions can be picked:
 - Within creation wizard for a Decision Table or Event Rule
 - In the Decision Table/ Decision Tree/ Event Rule Editor as a result column
 - In the Extended Cell Editor for an Assignment
- As all Action input fields are predefined by the wizard, no additional mapping is required here



see chapter 7

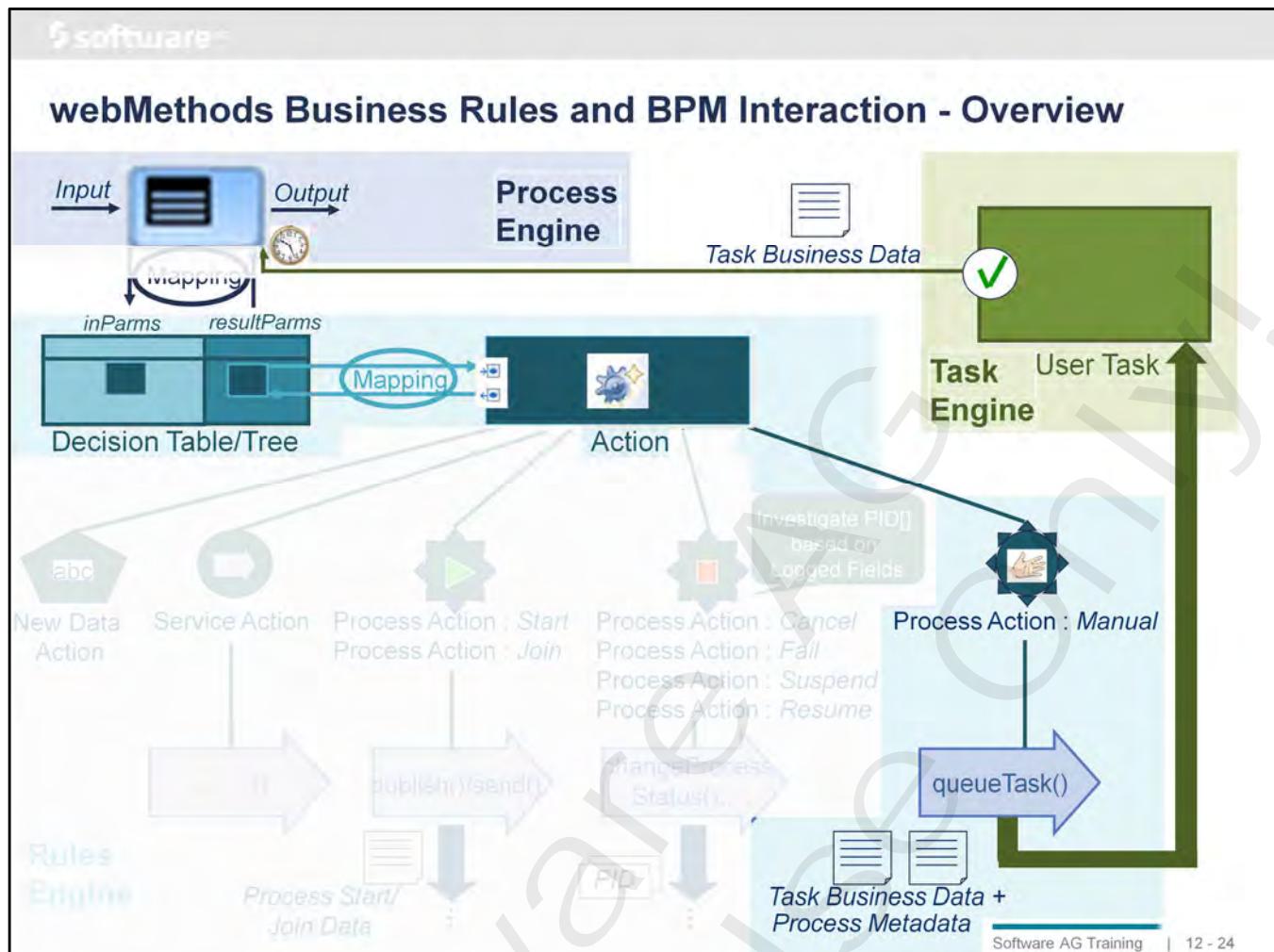
Notes:



Exercise 15

- In this exercise, you will create a Process Action that cancels running process instances of a certain model type and a matching condition based on logged process data fields. The action will be invoked conditionally by a Decision Table.

Notes:



Notes:

Process Actions of Type Manual

- Usage Scenario
 - A manual decision needs to be taken
 - Dynamic BPM
- Configuration
 - Select User Task to instantiate
 - Depending on the User Task selected, bind inputs and outputs to the data model elements
- Semantics
 - When the action is triggered, a User Task gets queued – and depending on User Task configuration, is assigned to a user
 - The user completes the User Task
 - The Task Engine calls back the Process Engine with result and the original call context

Salary	Score	Rate
>200	>800	3%
<=200	700-800	4%
<=200	>700	Manual

Software AG Training | 12 - 25

Notes:

Creating a Process Action of Type Manual

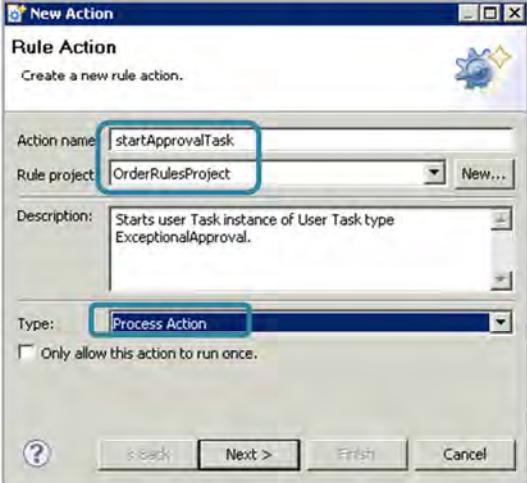
- Creation of a Process Action supported by wizard
- Add Process Action in Rules Explorer view (*New -> Action*)
- Process Actions require for “Business Rules Add-ons” license (WOKAA)

- Create a Process Action of type Manual:
 1. Name, description, Action type (Process)
 2. Select Process Action type (Manual)
 3. Select User Task type to start
 4. Provide default input for User Task data
 5. Check/uncheck Process Action's output as result

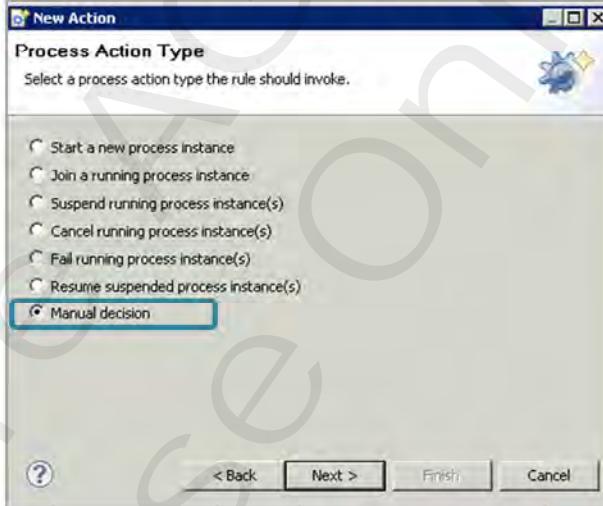


Notes:

1-2: Creating a Process Action of Type Manual



The screenshot shows the 'New Action' dialog for creating a rule action. The 'Action name' field contains 'startApprovalTask'. The 'Rule project' dropdown is set to 'OrderRulesProject'. The 'Description' field states 'Starts user Task instance of User Task type ExceptionalApproval.' The 'Type' dropdown is set to 'Process Action', which is highlighted with a blue box. A checkbox below it is unchecked. At the bottom are buttons for '?', '< Back', 'Next >', 'Finish', and 'Cancel'.



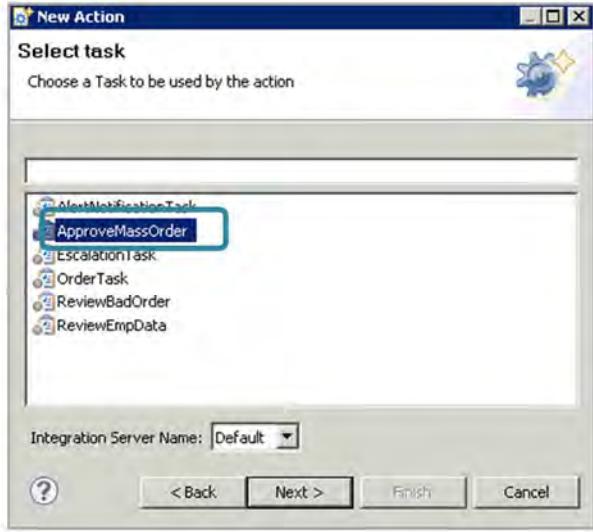
The screenshot shows the 'Process Action Type' dialog. It asks to select a process action type. Several options are listed as radio buttons, with the last one, 'Manual decision', being selected and highlighted with a blue box. At the bottom are buttons for '?', '< Back', 'Next >', 'Finish', and 'Cancel'.

- Provide name, rule project and description (optional)
- Choose type "Process Action"
- Allow/disallow to execute Process Action multiple times

- Select Action type "Manual decision"

Notes:

3: Create a Process Action of Type Manual

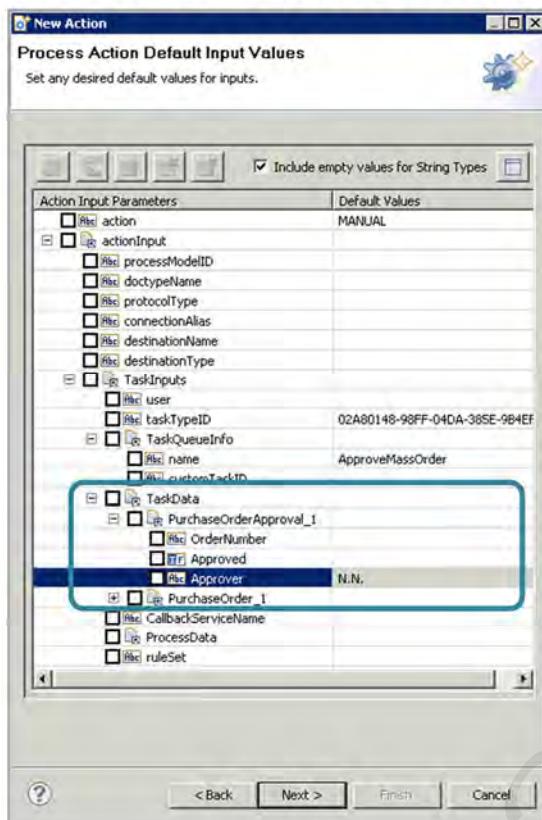


- Select existing User Task type published to MWS
- User Task types retrieved from Designer workspace or VCS

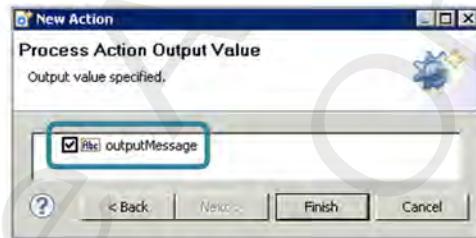
Software AG Training | 12 - 28

Notes:

4-5: Create a Process Action of Type Manual



- *Optional:* Provide default input for the User Task data
 - Data for various fields can be static values or dynamic bindings to Data Model elements
 - Don't change preconfigured field taskTypeID



- Check/uncheck action output value

Software AG Training | 12 - 29

Notes:

Invoke Process Actions of Type Manual

- Like other Actions, Process Actions cannot be executed standalone
- Invoke Process Actions from a Decision Entity as
 - a result
 - within an assignment (seldom used)
- Like other Actions, Process Actions can be picked:
 - Within creation wizard for a Decision Table or Event Rule
 - In the Decision Table/ Decision Tree/ Event Rule Editor as a result column
 - In the Extended Cell Editor for an Assignment
- Invoking Decision Entity MUST be checked as "Process Aware"
- Additionally map Rule parameter elements to Task Business data



see chapter 7

} see next slides

Notes:

Invoke Process Actions of Type Manual – Mapping...

■ Check invoking Decision Entity to be "Process aware"

■ Process Metadata of calling Rule Task Activity become available as Decision Entity Parameter

Automatically pre-selected

Notes:

The screenshot shows the 'Action Input Mapping' dialog and a decision table interface.

Action Input Mapping Dialog:

- Panel: Map Action Input
- Description: Map elements from the decision entity's input parameters to the action's input.
- Tree View: Pipeline In, PurchaseOrder_1, PurchaseOrderApproval_1, TaskInputs, TaskQueueInfo, TaskData, PurchaseOrderApproval_1, PurchaseOrder_1.
- Right Panel: startApprovalTask: Action In, action, actionInput, processModelID, doctypeName, protocolType, connectionAlias, destinationName, destinationType, TaskInputs, user, taskTypeID, TaskQueueInfo, TaskData, PurchaseOrderApproval_1, OrderNumber, Approved, PurchaseOrder_1, CallbackServiceName, ProcessData.
- Buttons: OK, Cancel.

Decision Table Action Results:

- Panel: New Decision Table
- Description: Select actions for action invocations.
- Table: Available actions (startApprovalTask selected), Selected actions (startApprovalTask).
- Buttons: Back, Next, Finish, Cancel.

Notes:

- Open mapping editor to map Rule Parameter Elements to Manual Process Action
 - Action's default input values can be overwritten

Notes:



Exercise 16

- In this exercise, you will create a Manual Process Action to start a User Task. The Action will be invoked by a Decision Tree which will be called by a Rule Task Activity in a business process. This scenario shows how to call a Decision Tree from a business process to dynamically assign User Tasks depending on the rule conditions.

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



13

Business Rules invoked from BPM User Tasks

Notes:

Objectives

At the end of this chapter you ...

- Created a User Task Assignment in the Task Editor invoking a Business Rule Decision Entity
- Tested the User Task Assignment

Notes:

Using Business Rules in the Digital Business Platform

- From a Business Process**
Put a Rule Task Activity step into a Business Process
- From an Event source (BPM or other)**
An Event (int./ext.) triggers an Event Rule and starts related Rule processing
- For a User Task Assignment**
A Decision Entity is called when a User Task Assignment is performed
- From Integration Server**
Call a Decision Entity using a public IS service

DIGITAL BUSINESS PLATFORM

Notes:

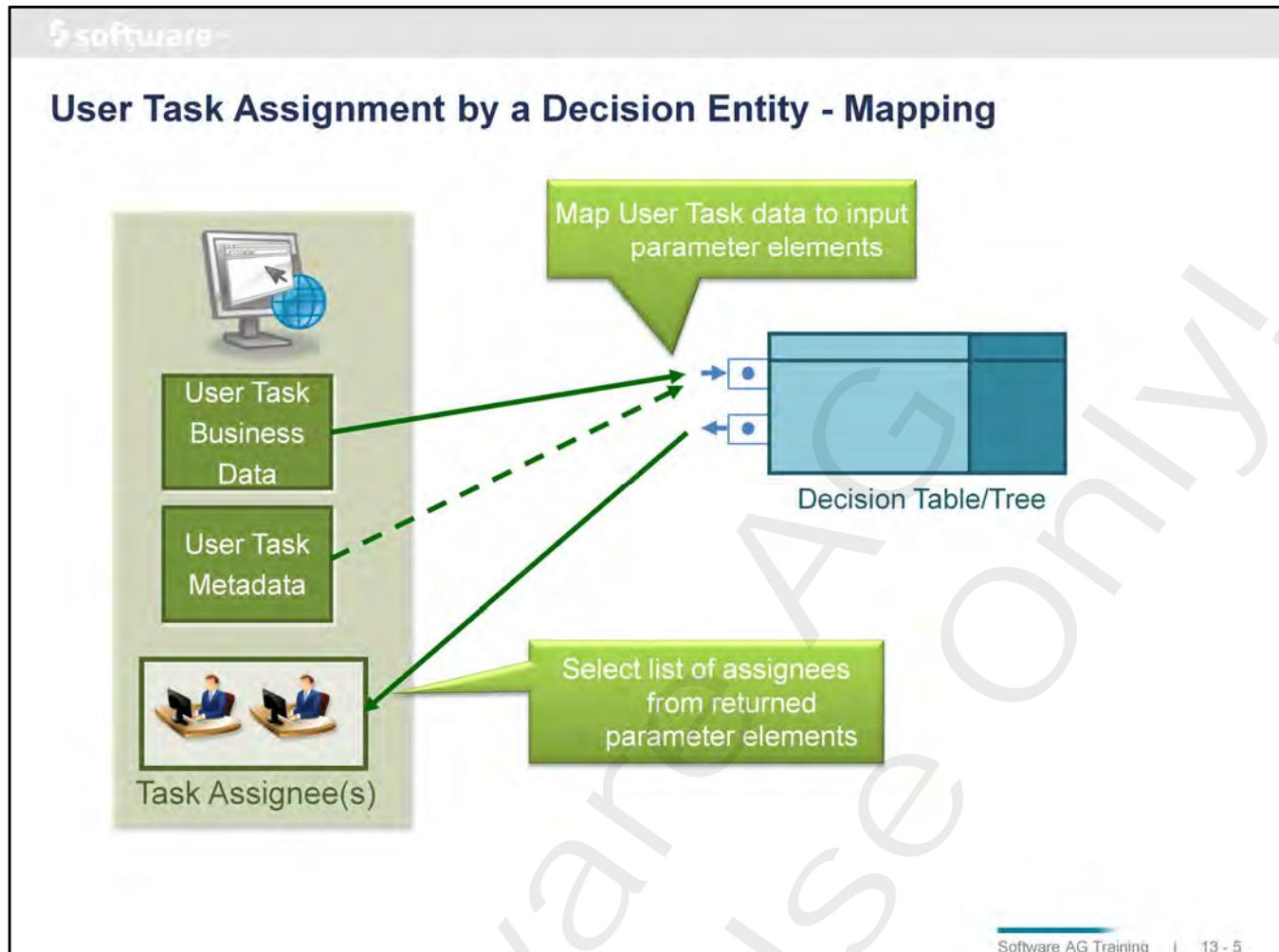
User Task Assignment

The diagram illustrates the interaction between a Task Assignee (a person at a desk) and a User Task Execution (a computer monitor displaying a web browser). Below, a process flow diagram shows a sequence of activities: Receive Order, Validate Order, Approve Order, Map to Canonical, Persist Order, and Terminate. The Approve Order activity is highlighted with a user icon, indicating it is a user task. A dashed double-headed arrow labeled 'Assignment' connects the Task Assignee and the User Task Execution.

- Processes interact with human tasks through User Task Activities
- User Task has to be assigned to an Assignee (Task Assignment)
 - ➡ List of Assignees can be returned by a Decision Entity or Rule Set

Software AG Training | 13 - 4

Notes:



Notes:

Setup User Task Assignment...

The screenshot shows the 'Assignments' screen where a new assignment is being created. The 'Assignment Name' is set to 'DynamicAssignmentByRule'. In the 'Assignee' section, a 'Business Rule' is selected from the dropdown menu. A modal dialog titled 'Edit Business Rule Assignment' is open, showing the 'Business Rule Source' configuration. The 'Project' is set to 'OrderRulesProject', and the 'Table' is set to 'DetermineOrderTaskAssignee'. The 'Use Decision Table' radio button is selected.

- Use Task Editor to create a User Task Assignment
- Select type "Business Rule"
- Browse for Decision Entity or Rule Set

Notes:

...Setup User Task Assignment...

The screenshot shows the 'Edit Business Rule Assignment' dialog. On the left, under 'Input Parameter Decision Entity', there is a tree view of data elements. In the center, under 'User Task Data', there is a text area for mapping. A green arrow labeled 'Mapping direction' points from the input tree to the text area. The text area contains two lines of mapping code:

```
#{currentTask.taskData.purchaseOrder.numberOfOrderedItems} ...  
#{currentTask.taskData.purchaseOrder.customer.rating}
```

A blue box highlights the first line of code, and a green box highlights the second line of code.

- Map User Task Business/ Meta data to Decision Entity input parameter elements

Software AG Training | 13 - 7

Notes:

...Setup User Task Assignment

The screenshot shows the 'Edit Business Rule Assignment' dialog with the title 'Select Assignment Field'. It displays a list of 'Rule Output' elements on the left and a 'Selection' grid on the right. A callout box labeled 'User Task Data' points to the 'Assignee' row in the selection grid, which has 'Yes' selected. Another callout box provides instructions: 'Select 'Yes' if parameter element returns a string value to be used as an Assignee'. The 'Mapping direction' button is highlighted.

Output Parameter Decision Entity

User Task Data

Select 'Yes' if parameter element returns a string value to be used as an Assignee

Mapping direction

- Select Assignee(s) from returned parameter elements
- Multiple Assignees can be selected
- Selected and returned string value(s) should contain a defined MWS principal or a comma-separated list of MWS principals
 - If a principal does not exist, value is ignored

Software AG Training | 13 - 8

Notes:

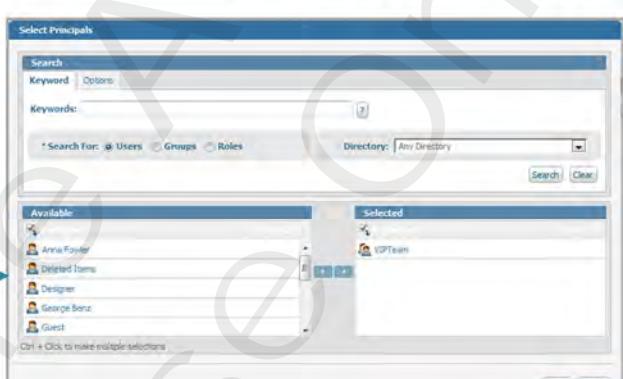
Principal Picker Feature in Designer and RMC



- Designer allows to enable an MWS Principal Picker for an Assignment column of a Decision Table*
- Principal Picker becomes visible in RMC only



- Supported Principal types can be customized in My webMethods**



- RMC offers Principal Picker for that column to pick one or multiple MWS Principals as Assignment value

	Country	CustomerStatus	Discount	HandledBy
1	= US	= Gold	= 20%	= VIPTeam 
2	= US	= Silver	= 10%	= StandardTeam
3	= Germany	= Gold	= 15%	= VIPTeam
4	= Germany	= Silver	= 5%	= StandardTeam

* not available for Decision Trees

** Supported Principal types can also be customized using Command Central

Notes:



Exercise 17

- In this exercise, you will invoke a Decision Table from a User Task. The Decision Table will return a string value that contains an assignee for the User Task Assignment. The value can contain a user, group, or role.
- User Task and user definitions are already provided in your environment.

Notes:



14

Business Rules invoked from IS Services



Notes:

Objectives

At the end of this chapter you ...

- Invoked Decision Entities deployed to an IS runtime from an IS service
- Know where to find the IS interface documentation of the Rule Engine

Notes:

Using Business Rules in the Digital Business Platform

The diagram illustrates four methods for integrating business rules into the Digital Business Platform:

- From a Business Process:** Shows a BPMN-like process flow with a "Rule Task Activity" step.
- From an Event source (BPM or other):** Shows an event source triggering a rule processing step.
- For a User Task Assignment:** Shows a decision table mapping discount levels to handling teams.
- From Integration Server:** Shows a service structure for calling decision entities.

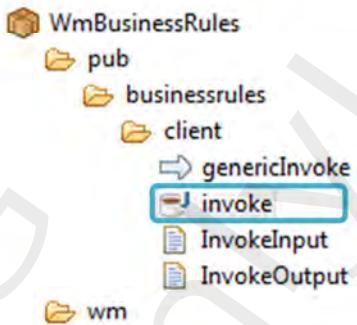
DIGITAL BUSINESS PLATFORM

Software AG Training | 18 / 3

Notes:

Invoke Decision Entities from Public IS Service

- Invocation of
 - Decision Table
 - Decision Tree
 - Rule Set
- via public IS service in package WmBusinessRules:
`pub.businessrules.client:invoke()`
- Prerequisites:
 - Business Rule project deployed (exported) to Integration Server runtime
 - Packaged within feature “Business Rules Public API”,
so “Business Rules Add-ons” license is needed (WOKAA)
- IS interface and service described in documentation
“webMethods Business Rules Reference”



Software AG Training | 14 - 4

Notes:

invoke() - Service Signature

Input parameters:

- Project Name
 - Name of Business Rule project
- Invocation Target
 - Specify "**DT/**<DecisionTableName>" to invoke a Decision Table
 - Specify "**D3/**<DecisionTreeName>" to invoke a Decision Tree
 - Specify "**RS/**<RuleSetName>" to call a Rule Set
- Create Missing Inputs (*optional*)
 - Create empty values {true | false}
- Inputs
 - Document(s) containing the required rule input parameter
 - Parameter name(s) must fit
- Desired Outputs (*optional*)
 - Rule output parameter to be returned
 - Acts like a filter
 - If empty, all parameters are returned

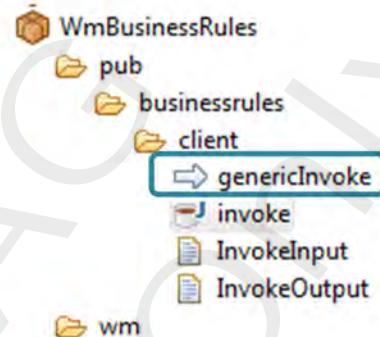
Output parameter:

- Outputs
 - Document(s) containing the returned rule output parameter

Notes:

Add Invocation to a Custom Flow Service...

- Add invocation of `pub.businessrules.client:invoke()` to custom Flow service
 - Manually assign all required inputs and outputs
 - Map to Inputs and from Outputs document
 - Flow service `genericInvoke` demonstrates how to specify Inputs



Software AG Training | 14 - 6

Notes:

...Add Invocation to a Custom Flow Service

- Much easier:
Drag & drop a Decision Table, Decision Tree or a Rule Set into Tree view of an IS Flow Service
 - Creates a pre-configured invocation of `pub.businessrules.client:invoke`
 - Assigns parameters to Project Name, Invocation Target, Desired Outputs

The screenshot shows the Software AG Studio interface. At the top, there's a toolbar with various icons. Below it is a main workspace divided into several panes:

- Top Left:** A tree view pane titled "invokeCalcaulateDiscount" containing a "pub.businessrules.client:invoke" node and a "MAP" node.
- Top Right:** A "Specification Reference" pane with input and output fields, and checkboxes for "Validate input" and "Validate output". It also shows a mapping between "Customer (rulesSupport.docs:Customer)" and "discount".
- Bottom Left:** A "Pipeline" pane showing a "Pipeline (OrderRulesProject)" with a "createOrder" step and a "DT/CalculateDiscount" step. The "DT/CalculateDiscount" step is highlighted with a blue arrow pointing to it from the text "Map to Inputs".
- Bottom Right:** A "Service Out" pane showing the "invoke" service with its "Outputs" and "ruleSupport.rules.Customer" output. A blue arrow points from this pane to the "Customer" parameter in the pipeline, labeled "Map from Outputs".
- Bottom Center:** A "Pipeline Out" pane showing the final output structure.

Annotations in the screenshot include a blue arrow labeled "Map to Inputs" pointing to the "DT/CalculateDiscount" step in the pipeline, and another blue arrow labeled "Map from Outputs" pointing to the "Customer" parameter in the pipeline output.

Notes:

The screenshot illustrates the 'Test Invocation in Designer' feature. In the Package Navigator, a flow named 'InvokeCalculateDiscount' is selected. A context menu is open, with 'Run As' being the active option. Under 'Run As', '1 Run Flow Service' is highlighted. The Pipeline Results window shows the configuration for this invocation, including inputs like 'Invocation Target' and 'Create Missing Inputs', and desired outputs like 'Desired Outputs[0]' and 'Customer' details.

Name	Value
Project Name	OrderRulesProject
Invocation Target	DT/CalculateDiscount
Create Missing Inputs	true
Inputs	
rulesSupport_docs_Customer_1	
Rating	gold
PurchaseHistoryCount	33
OpenInvoices	1
Desired Outputs	
Desired Outputs[0]	
Parameter Name	rulesSupport_docs_Customer_1
Customer	
Discount	15
OpenInvoices	1
PurchaseHistoryCount	33
Rating	gold
Ruleset Name	DT/CalculateDiscount

Notes:



15

External Event Rules and Event Models

Notes:

Objectives

At the end of this chapter you ...

- Can explain the concepts and advantages of an Event-Driven Architecture (EDA)
- Know how to configure Event Messaging and Event Routing for Business Rules
- Generated Event Types in Designer
- Deployed Event Types
- Created Event Models in Designer
- Created External Event Rules
- Created JMS Trigger for External Event Rules
- Know to test and invoke External Event Rules

Notes:

Using Business Rules in the Digital Business Platform

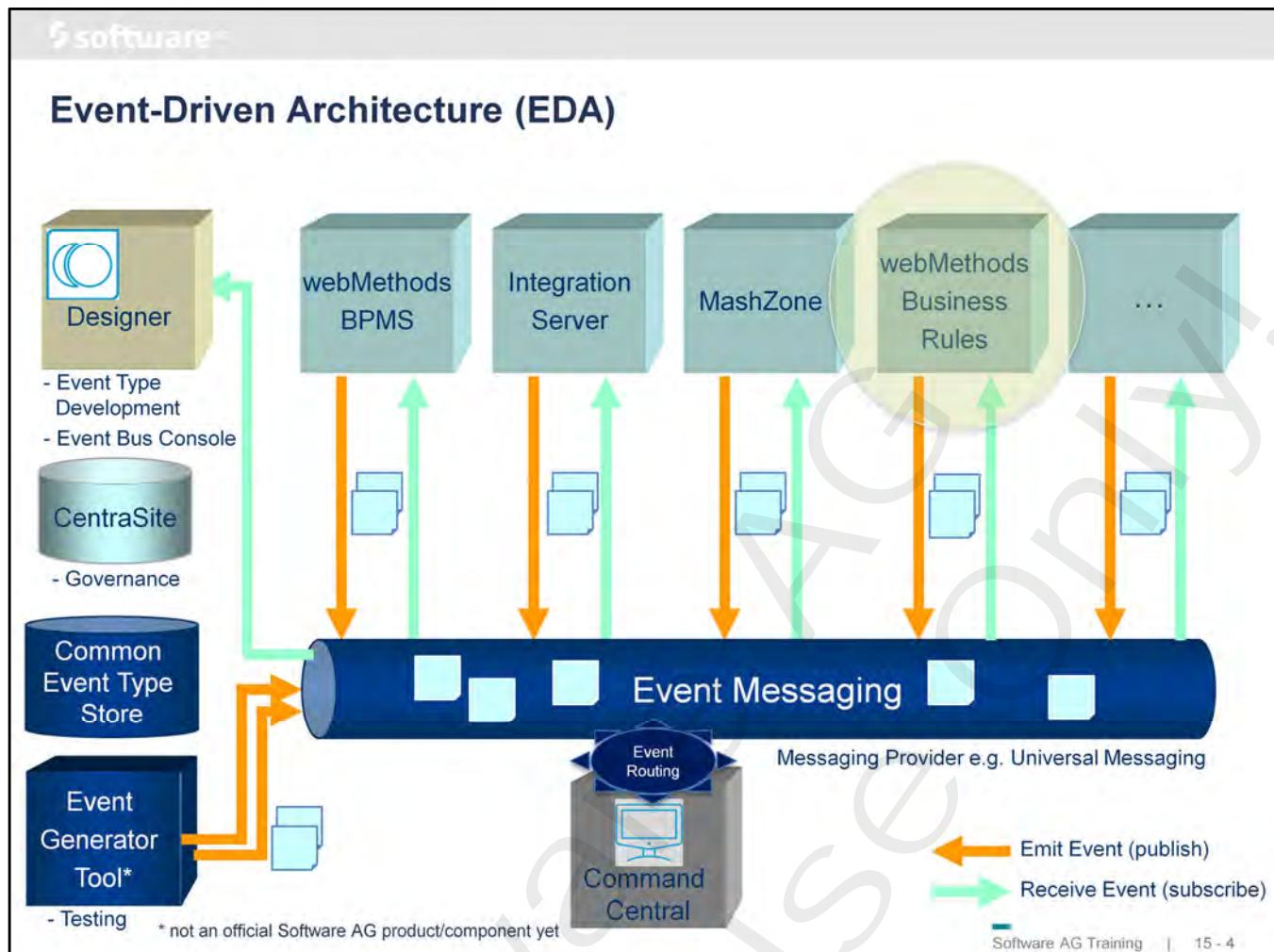
The diagram illustrates four methods for integrating Business Rules into the Digital Business Platform:

- From a Business Process:** Shows a BPMN process diagram with a Rule Task Activity step.
- From an Event source (BPM or other):** Shows an event definition for "OrderReceivedEM" with results for "CreateOrderReceivedDocument()" and "writeToLog()".
- For a User Task assignment:** Shows a decision table mapping discount levels to teams: 20% to VIPTeam, 10% to StandardTeam, 15% to IPTeam, and 5% to StandardTeam.
- From Integration Server:** Shows a directory structure for "WmBusinessRules" including "pub", "businessrules", "client", "genericinvoke", "invoke", "invokewithinput", and "InvokeOutput".

DIGITAL BUSINESS PLATFORM

Software AG Training | 15 / 3

Notes:



Notes:

 software AG

EDA Participants

- Designer
 - Event Development perspective, Event Bus Console*
- Common Event Type Store
- Command Central
 - Event Routing configuration
- Universal Messaging
 - Event Messaging using JMS
- Integration Server
 - Event Messaging and Event Routing (ER) API
- webMethods BPMS**
- [webMethods Business Rules**](#)
- Event Generator Tool (unsupported)*

*using EDA Orchestrator API ** based on Integration Server EDA API and native JMS Triggers

Software AG Training | 15 - 5

Notes:

Some Event Technology



Notes:

Software AG

EDA Event Types

- webMethods EDA uses XML Schema (.xsd) for Event Type definitions
- Event Type definition is a blueprint for each event instance
- Event Type structure similar to a SOAP request
- Acts as a contract between publishers and receivers
- Allows reuse of groups of event attributes

```
graph LR; subgraph Event_Type [EDA Event Type]; subgraph Header [Header]; H1[Type, Kind]; H2[Start Time]; H3[End Time]; end; subgraph Payload [Payload]; P1[e.g. Stock Id]; P2[Order Amount]; P3[GPS Location]; P4[Temperature]; end; Event_Type --> EnvelopeXSD[Envelope.xsd]; Event_Type --> UserDefinedXSD[User-defined xsd]; Event_Type --> ComponentXSD[Component XSD]
```

Envelope.xsd

User-defined xsd

Component XSD

Header
Type, Kind
Start Time
[End Time]
...

Payload
e.g. Stock Id
Order Amount
GPS Location
Temperature

Software AG Training | 15 - 7

Notes:

Exchanging EDA Events

```

<xsd:element name="Event">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Header" type="HeaderType"/>
      <xsd:element name="Body" minOccurs="0">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="Payload" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

```

<xsd:schema ... xmlns:evt="http://namespaces.softwareag.com/EDA/Event"
  xmlns="http://namespaces.softwareag.com/EDA/WebM/Sample/InventoryMgmt"
  targetNamespace="http://.../EDA/WebM/Sample/InventoryMgmt">
  <xsd:import namespace="..." schemaLocation=".../Envelope.xsd"/>
  <xsd:element name="PartInventoryLow" substitutionGroup="evt:Payload"
    type="PartInventoryLowType" />
  <xsd:complexType name="PartInventoryLowType">
    <xsd:annotation>
      <xsd:documentation>Report inventory low for a part</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="Part" type="PartType"></xsd:element>
      <xsd:element name="InventoryLevel" type="xsd:integer"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>

```

Envelope.xsd

User-Defined xsd

Emitter

Receiver

Event : JMS Topic

Event Messaging via JMS Message Provider

Software AG Training | 15 - 8

Notes:

Event Messaging and Event Routing Requirements

EDA Event Messaging and Event Routing requires for:

1. Configured UM realm supporting JMS incl. configured JNDI Connection Factory* in UM realm
2. Common Event Type Store
3. Configured (Default) Event Routing
4. Configured JMS connectivity to UM realm in IS*
 - a) JNDI Connection Alias
 - b) JMS Connection Alias using JNDI Provider Alias
5. Event Messaging and Event Routing API (in IS)



Software AG Installer creates 1.5 during installation

* only required for EDA participants currently not supporting/bypassing Event Routing

Software AG Training | 15 - 9

Notes:

1. Configured Universal Messaging Realm Supporting JMS

- Configured UM realm will be used as default JMS provider for Event Messaging
- Connectivity to UM configured in Event Routing and Integration Server*

* only required for EDA participants currently not supporting/bypassing Event Routing

Software AG Training | 15 - 10

Notes:

2. Common Event Type Store...

- Common location containing
 - Event Types as XML schemas
 - Component schemas
- Used by EDA participants within the same installation
 - Integration Server incl. BPM and Business Rules
 - Designer
 - ...
- Initial Content
 - *Envelope.xsd*
 - Predefined Events

eventTypeStore

File Edit View Tools Help

Organize Include in library Share with New folder

Name

SoftwareAG

_documentation

Apama

ApamaCapitalMarketsFound

API_Portal

ApplnX

Broker

CentraSite

CloudStreamsAnalytics

common

ADAPA

AssetBuildEnvironment

bin

conf

db

DigitalEventServices

EventTypeStore

_mappings

Common

Event

META-INF

WebM

5 items

Common Event Type Store

Software AG Training | 15 - 11

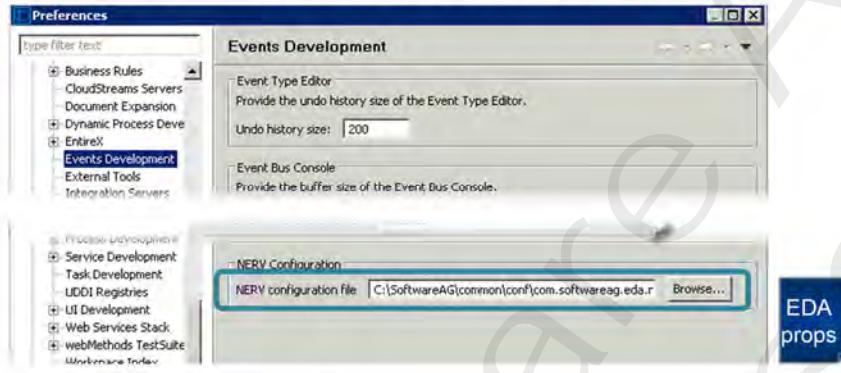
Common Event Type Store

The event type store provides a central location where predefined and user-defined event types are stored. This shared location is used by all EDA participants at run-time to retrieve deployed custom event types.

... 2. Common Event Type Store

Common Event Type Store Location

- For each OSGI-based, Event Routing-enabled product (e.g. Integration Server, Command Central), the location of the Common Event Type Store in the current installation is preset:
`<Inst-Dir>\common\EventTypeStore`
- Designer*: Location of Common Event Type Store contained in the parameter file:
`<Inst-Dir\common\conf\com.softwareag.eda.nerv.properties`



* Designer doesn't support Event Routing yet.

Common Event Type Store

Common Event Type Store and Designer:

At design time, a local copy of the predefined event types of the event type store is available for reference. User-defined event types can be created using the Event Type Editor and stored in the local copy (workspace or VCS).

Event types in the local copy must be deployed to the run-time store, so that EDA participants that process an event stream can retrieve the schema definition of the event.

Location of a sample Event Type schema in the Designer project (Workspace or VCS):

`...\.metadata\.plugins\com.softwareag.eda.eventtypeeditor\EventTypeStore\
Event Types\WebM\Sample\WebM\Sample\Geofencing\VehiclePosition.xsd`

3. Configured (Default) Event Routing

Event Routing defines and configures for an EDA participant:

- **Available Services**
 - Messaging Service (JMS Provider), by default: UniversalMessaging (UM realm)
 - Event Persistence
 - In-process
- **Available Service Groups**
 - Service Group contains one or multiple Services
 - Default Service Group contains default UniversalMessaging Service
 - Is used for all Events which haven't been associated to a custom Service Group
- **Available Event Type Associations**
 - Associate Event Types to custom Service Groups
 - By default, every unassociated Event Type is associated with the Default Service Group

Notes:

3. Configured (Default) Event Routing...

- **Event Routing** has to be configured for each EDA participant (e.g. IS instance) using Command Central
- Default UniversalMessaging Service and optional custom Messaging Service configuration:

The screenshot shows two side-by-side configurations of the Software AG Command Central interface. On the left, the 'Event Routing' section is selected, showing a 'Universal Messaging Settings' panel with a service named 'UniversalMessaging' and a provider URL of 'http://localhost:9000'. On the right, a separate configuration window titled 'Universal Messaging Settings' is open, showing a new service named 'Training-UM' with a description 'Messaging provider for training events' and a provider URL of 'http://segbase.eur.ad.sag:9000'. Both windows have 'Save' and 'Cancel' buttons at the bottom.

Notes:

...3. Configured (Default) Event Routing...

- Default Service Group and optional custom Service Groups configuration:

Software AG Command Central

Home > Instances > All > IS_default

IS_default

Application Platform

Digital Event Services

Event Routing

Integration Server

NERV

Web Bundle Runtime

Service Groups

Group Name: Default

Group Description: The default event group includes all event types that are not added to other defined groups

Services:

Service Name	Reliable
UniversalMessaging	true

Service Groups

Group Name*: Training

Group Description*: Training Service Group

Services:

Service Name	Reliable	Usage
Training-UM	true	Destination Only
In-Process		
Training-UM		
UniversalMessaging		

Test Save Cancel

Software AG Training | 15 - 15

Notes:

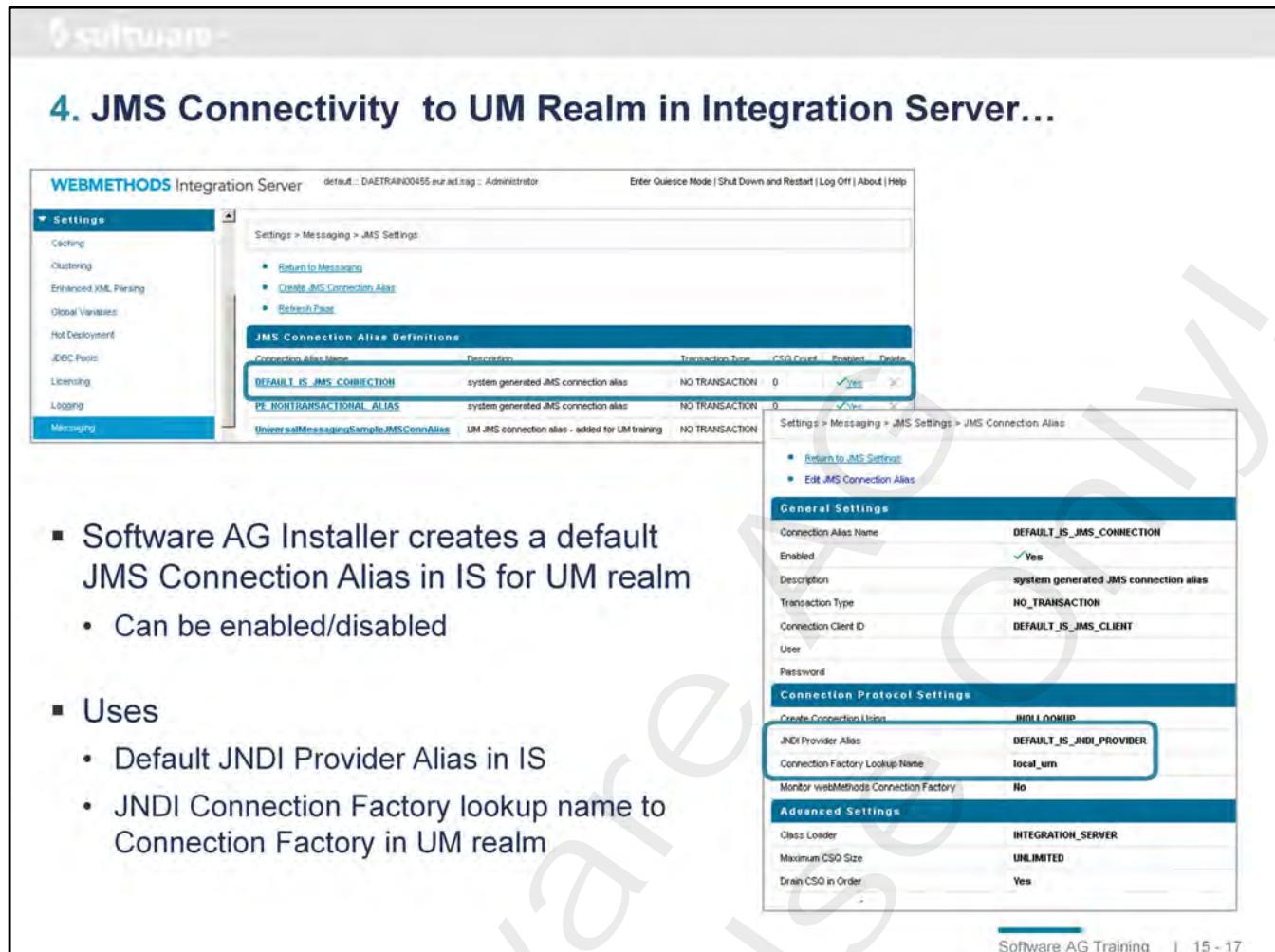
...3. Configured (Default) Event Routing

- *Optionally:* Add custom Event Type Associations if Event Type should use a different Messaging Service as defined in the Default Service Group
- Event Types being listed are retrieved from the Common Event Type Store of the (remote) instance to be configured, e.g. IS instance

Event Type	Namespace	Service Group	In-Memory Capacity	On-Disk Capacity
FraudEvent	.../Edu/Credit	Training	Default	Default
MeasurementResult	.../EDA/WebM/Analysis/1.0	Default	Default	Default
MeasurementResultStatistics	.../EDA/WebM/Analysis/1.0	Training	Default	Default

Notes:

4. JMS Connectivity to UM Realm in Integration Server...



The screenshot shows the Software AG Integration Server web interface. On the left, a sidebar menu under 'Settings' includes options like Caching, Clustering, Enhanced XML Parsing, Global Variables, Hot Deployment, JDBC Pools, Licensing, Logging, and Messaging. The main content area displays 'JMS Connection Alias Definitions' with two entries:

Connection Alias Name	Description	Transaction Type	CSQ Count	Enabled	Date
DEFAULT_IS_JMS_CONNECTION	system generated JMS connection alias	NO TRANSACTION	0	<input checked="" type="checkbox"/> Yes	
PE_NONTRANSACTIONAL_ALIAS	system generated JMS connection alias	NO TRANSACTION	0	<input checked="" type="checkbox"/> Yes	

Below this, a detailed view of the 'DEFAULT_IS_JMS_CONNECTION' alias shows its configuration:

General Settings	
Connection Alias Name	DEFAULT_IS_JMS_CONNECTION
Enabled	<input checked="" type="checkbox"/> Yes
Description	system generated JMS connection alias
Transaction Type	NO TRANSACTION
Connection Client ID	DEFAULT_IS_JMS_CLIENT
User	
Password	

Connection Protocol Settings	
Create Connection Uri	JNDI LOOKUP
JNDI Provider Alias	DEFAULT_IS_JNDI_PROVIDER
Connection Factory Lookup Name	local_um
Monitor webMethods Connection Factory	No

Advanced Settings	
Class Loader	INTEGRATION_SERVER
Maximum CSQ Size	UNLIMITED
Drain CSQ in Order	Yes

Software AG Training | 15 - 17

Notes:

...4. JMS Connectivity to UM Realm in Integration Server

JNDI Provider Alias Definitions

JNDI Alias Name	Description	Test Lookup	Delete
DEFAULT IS JNDI PROVIDER	system generated JNDI provider alias		
UniversalMessagingJNDIAlias	UM JNDI alias - added for UM training		

Test lookup successful

• [Return to Messaging](#)
• [Create JNDI Provider Alias](#)

JNDI Provider Alias Definitions

JNDI Alias Name	Description	Test Lookup	Delete
DEFAULT IS JNDI PROVIDER	system generated JNDI provider alias		
ClosedLoopAnalyticsProcessProject_EscalationProcess_SUBQUEUE	javax.jms.Queue		
ClosedLoopAnalyticsProcessProject_EscalationProcess_TRANSQUEUE	javax.jms.Queue		
EDAProcessProject_TestEDAProcess_TRANSQUEUE	javax.jms.Queue		
Event/WebMProcess/V2_0ProcessInstanceChange	javax.jms.Topic		
Event/WebMProcess/V2_0ProcessStepInstanceError	javax.jms.Topic		
Event/WebMProcess/V2_0ProcessStepInstanceChange	javax.jms.Topic		
Event/WebMProcess/V2_0ProcessStepInstanceError	javax.jms.Topic		
Event:CreditFraudEvent	javax.jms.Topic		
Event:WebM_SampleInventoryMount_1_0_PartInInventoryShortage	javax.jms.Topic		

Notes:

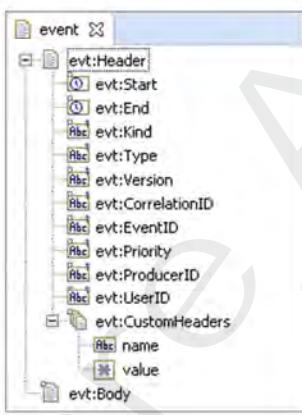
Software AG

5. Event Messaging and Event Routing API in Integration Server

- Integration Server offers built-in assets and service API in pub.event.* folder for EDA calls and Event Routing
- Internally used by other webMethods products participating to the EDA like BPM* and Business Rules

Integration Server

- WmPublic
- pub
 - assets
 - cache
 - client
 - date
 - document
 - event
 - eda
 - eventToDocument
 - event
 - schema_event
 - routing
 - send
 - subscribe
 - unsubscribe
 - eventAcknowledgement
 - addSubscriber
 - alarm
 - audit
 - auditError
 - deleteSubscriber



* with BPM version 10.0 and above

Software AG Training | 15 - 19

Notes:

How to Create Custom EDA Event Types using Designer...

- Use provided Events Development perspective in Designer
- Events can be created as part of an "Event Type Project" in the Project Explorer view

The screenshot shows the Software AG Designer interface. A 'New' dialog box is open, titled 'Select a wizard'. Under 'Wizards:', 'Event Type Project' is selected and highlighted with a blue border. In the center, there is a 'New Event Type Project' dialog with a 'Project name:' field containing 'DemoEvents!'. To the right, the 'Project Explorer' view shows a tree structure with various projects like 'Backup', 'BPMDevSupport[localhost_5555]', and 'DemoEvents'. Inside 'DemoEvents', there is a 'Event Types' folder containing 'Credit'. A context menu is open over this folder, with 'Event Type' highlighted in blue. Other options in the menu include 'New', 'Go Into', 'Copy', 'Delete', 'Move...', 'Rename...', 'Project...', 'File', 'Folder', 'SQL File', 'Event Type Project', 'Example...', and 'Other...'. At the bottom of the dialog, there are 'Back', 'Next >', 'Finish', and 'Cancel' buttons.

Notes:

... How to Create Custom EDA Event Types using Designer

▪ EDA Event Type can be created

- from scratch
- derived from an IS Document type

As stated by R&D, b) is not recommended for Event Type creation because the created xsd is not in the shape that will be expected by Business Rules.

Always start with the creation of the Event Type from scratch, then create a corresponding IS Document type by importing the structure from the XSD.

For some reasons, in Designer, a local copy of the predefined event types of the event type store is available for reference. This allows to create an ID Document type based on an (XSD) Event Type by importing a new Document from an xsd. All available xsd show up when you click the browse button.

Location in the workspace is:

<workspace>\metadata\plugins\com.softwareag.eda.eventtypeeditor\EventTypeStore\Event Types\

Create Custom EDA Event Types in Designer from Scratch

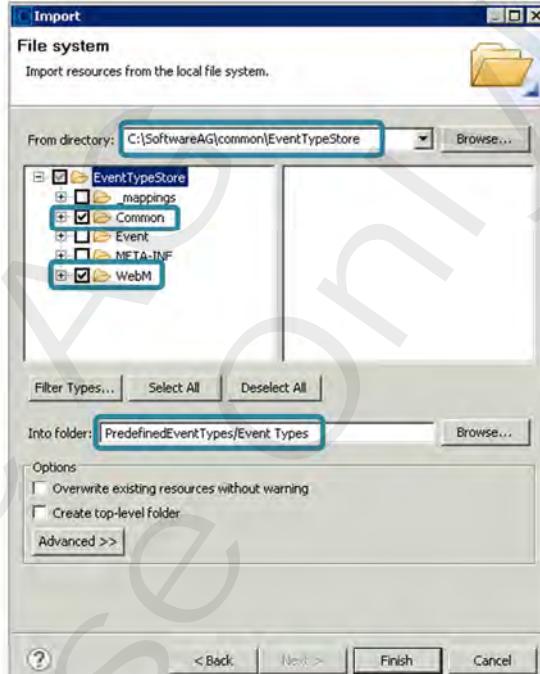
- Event Type Editor can be used for Event Type creation from scratch or later modification
- Offers Palette to define/modify the Event Type structure
- Properties view to customize field properties (type cardinality,...)
- Save persists Event Type to Workspace or VCS – not to Event Type Store!

Software AG Training | 15 - 22

Notes:

Optional: Importing EDA Event Types into Designer...

- Many Predefined EDA Events are shipped with Software AG Common Event Type Store
- To import them into Designer:
 1. Create Event Type project, e.g. PredefinedEventTypes
 2. At generated Event Type project root folder Event Types: Choose "Import > File System"
 3. Browse for existing Event Type Store root folder
 4. Select folders Common and WebM

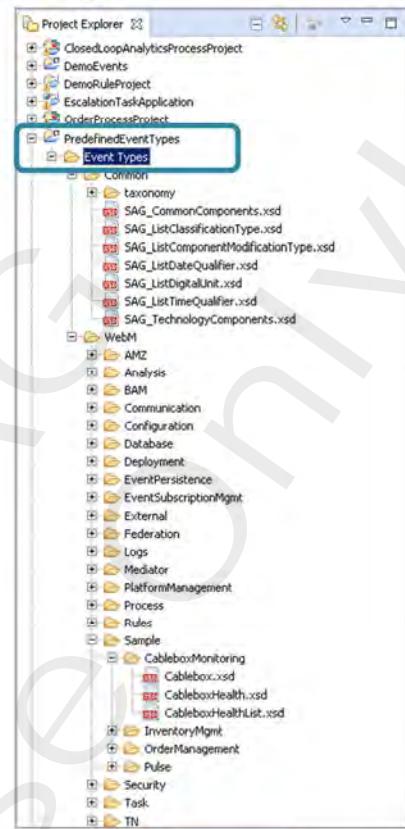


Software AG Training | 15 - 23

Notes:

...Optional: Importing EDA Event Types into Designer

- Imported predefined Software AG EDA Event Types become available in chosen Event Type project
 - Some EDA Event Types are nested
 - Can be used for EDA samples and demos



Software AG Training | 15 / 24

Notes:

Deploy Custom EDA Event Types from Designer

- Before you can use Custom Event Types created in Designer, they...
 - have to be deployed to the **Common Event Type Store**
 - may get a custom Event Type Association in **Event Routing** (*optional*)
 - require for JMS Destinations (topics) in the associated **Event Messaging** provider

- **Deployment possibilities:**

1. Deployment using
webMethods Deployer
(not covered here)
2. Manual Deployment



Notes:

Manual Deployment of Custom EDA Event Types – Deploy to the Common Event Type Store

- Copy Event Type schemas from Designer workspace or VCS into Common Event Type Store location of desired installation(s)

The screenshot illustrates the manual deployment process. On the left, a 'Project Explorer' window shows a file named 'FraudEvent.xsd' selected within a 'Credit' folder under the 'Event Types' section of the 'DemoEvents' project. A blue arrow labeled 'copy' points from this file to the right side of the screen. On the right, a 'EventTypeStore' window displays the target directory structure. The 'Credit' folder is highlighted with a blue border, indicating it is the destination for the copied schema. Below the window, a large blue cylinder icon represents the 'Common Event Type Store'. At the bottom right of the window, the text 'Software AG Training | 15 - 26' is visible.

Notes:

Manual Deployment of Custom EDA Event Types – Create Destinations at the Messaging Provider...

Add JMS topics and JNDI entries for new EDA Event Type to the Event Messaging Provider (UM) by performing one of the following choices:

- **Easy way:** Lacking JMS destinations (topics) are automatically generated in the Event Messaging Provider (UM) whenever you...
 - Emit an EDA Event of that type using IS service `pub.event.routing.send()`
 - Subscribe to receive EDA Events of that type using IS service `pub.event.routing.subscribe()`
 - Start an Event Bus Console to listen for corresponding EDA Event Type
 - Build & Upload a process model that uses corresponding EDA Event Type
 - ...
- **Otherwise:** Add destinations manually to the JMS Provider that implements the Event Messaging layer:
 - Enterprise Manager or utility `jmsadmin` (Universal Messaging)
 - My webMethods or utility `jmsadmin` (webMethods Broker, deprecated)

Notes:

...Manual Deployment of Custom EDA Event Types - Create Destinations at the Messaging Provider

- Example: Add JMS destination (topic) and JNDI entries for new EDA Event Type to UM as Event Messaging Provider using Enterprise Mngr.

The screenshot shows the Software AG Enterprise Manager interface. On the left, the Project Explorer displays a 'FraudEvent.xsd' file under a 'ClosedLoopAnalyticsProcessProject'. The 'FraudEvent.xsd' file contains XML code defining an EDA event type named 'FraudEvent'. On the right, the Universal Messaging Enterprise (UM) interface shows the creation of a channel. A tooltip indicates: "Destination (topic=UM Channel) name MUST match EDA Event type name and target namespace suffix as provided in the .xsd file". The channel attributes being set are:

Channel Name:	FraudEvent
Channel Type:	Transient
Channel TTL:	
Channel Capacity:	
Parent Realm:	umserver
Dead Event Store:	

Three boxes highlight key components: 'Event' (red), 'Credit' (blue), and 'FraudEvent' (cyan).

Notes:

Optional: Event Type Governance using CentraSite

- Publish Event metadata to CentraSite
 - Stores VCS location (CVS, SVN: Subclipse, Subversive) of Event Type schema
- Search for Event Types
 - Get VCS location
- Graphical impact analysis
- Un-publish Event metadata from CentraSite

```
graph TD; UD[XSD] --> E[XSD]; UD --> C1[XSD]; C1 --> C2[XSD]; C1 --> C3[XSD]
```

Event Type Governance

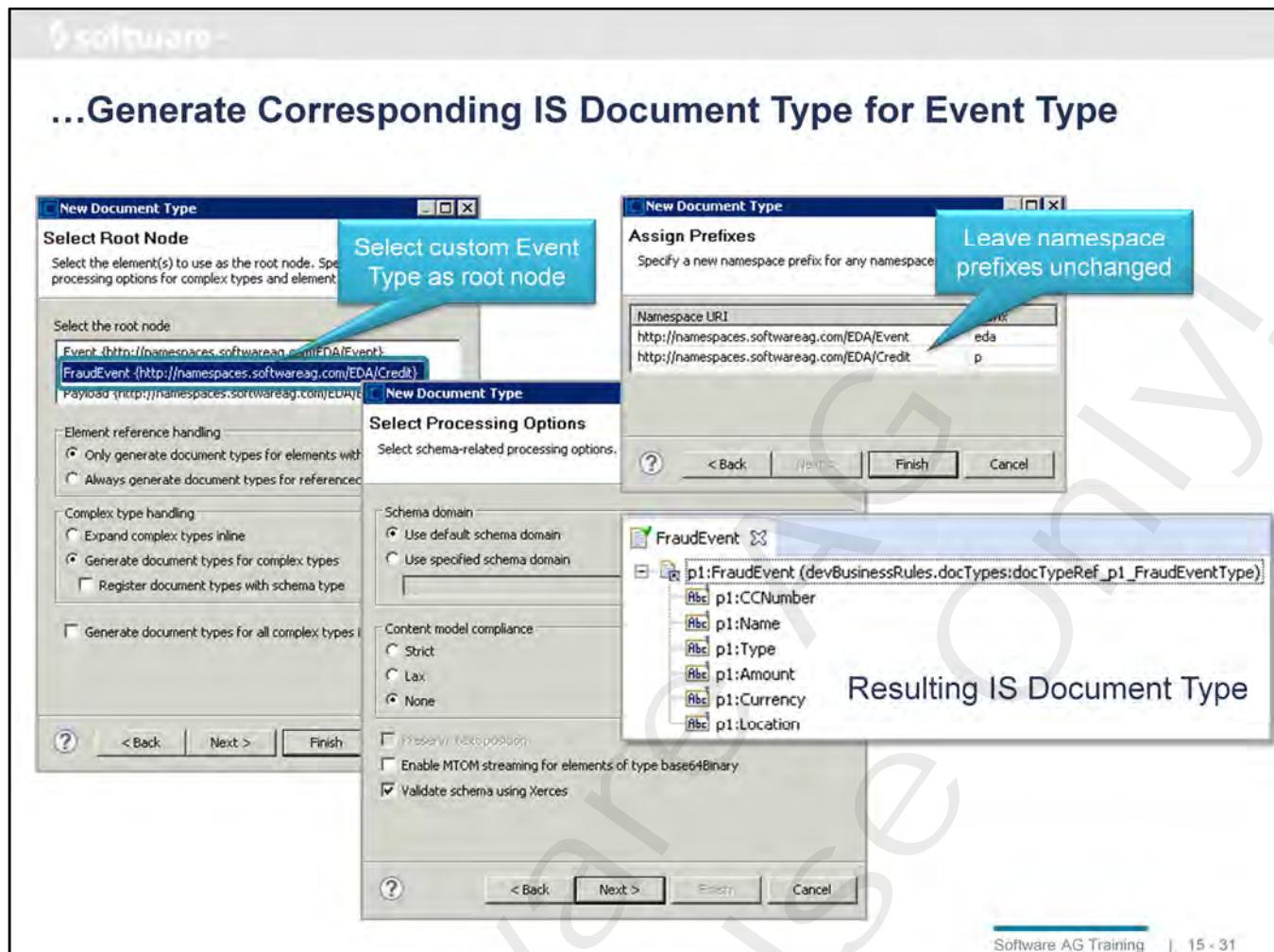
Use CentraSite to archive, categorize and govern event definitions. The Event Type Editor offers import and export functions for the transfer of event types to and from CentraSite.

Generate Corresponding IS Document Type for Event Type...

- To work on EDA Event in IS Services, BPM processes... a corresponding IS document type is required
- For a custom Event Type, always generate Event Type from scratch in Designer and deploy it to the Common Event Store first

Import Event Type schema from Common Event Type Store

Notes:



Notes:

Exchanging EDA Event using Event Messaging - Requirements

- IS Document type imported from the Event Type (.xsd)
- Running Event Messaging Provider implemented by JMS Provider
 - Default: Universal Messaging realm
- Access to the Event Messaging Provider:
 - Common Event Type Store
 - Configured Event Routing at IS, Command Central,...
 - IS assets in pub.event.routing
 - JMS access infrastructure at IS
 - JMS Trigger and Trigger handling services

The diagram illustrates the architecture for exchanging EDA events using event messaging. It shows the Integration Server (IS) interacting with the Event Messaging layer (Universal Messaging) and the Common Event Type Store.

The IS contains the following components:

- Designer (with EDA props)
- Rule Engine
- Process Engine
- JMS

The Event Messaging layer consists of:

- JMS
- Event Routing
- Event Messaging (Universal Messaging)
- Common Event Type Store

Interactions are indicated by arrows:

- Orange arrow pointing down from the IS JMS to the Event Messaging layer, labeled "Emit Event (publish)".
- Green arrow pointing up from the Event Messaging layer to the IS JMS, labeled "Receive Event (subscribe)".

Event Messaging

The event messaging functionality for webMethods EDA components is implemented by a JMS-based message provider like Universal Messaging (default) or webMethods Broker (deprecated).

In addition third party JMS providers can be utilized.

To access the Event Messaging layer a product, component, client or server can use:

- Event Routing
- IS JMS capabilities

Software AG

Let's get back to webMethods Business Rules

The illustration features a man with glasses and a maroon shirt, sitting cross-legged on a white floor with his hands raised in excitement. To his left is a stack of three books: a blue one at the bottom, a red one in the middle, and an orange one on top, all with the word 'Technology' visible. To his right is a green rounded rectangle containing the text 'webMethods Business Rules'. A large, faint watermark reading 'Internal Use Only!' is diagonally across the background.

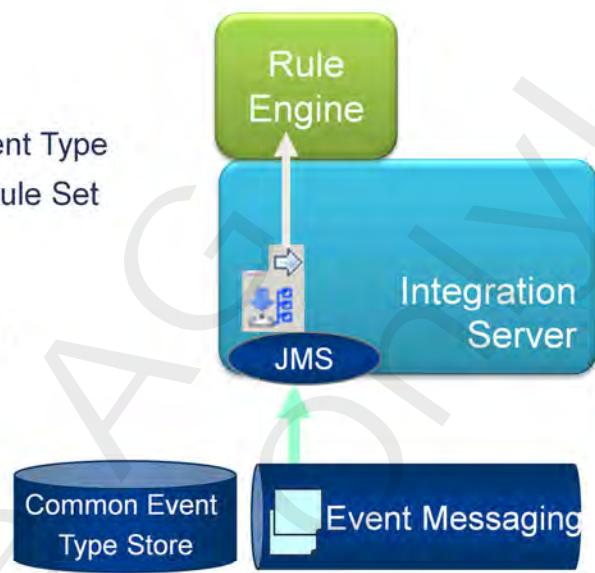
Software AG Training | 15 - 33

Notes:

Receiving EDA Events by webMethods Business Rules

Design Time:

- Rules Development perspective:
 1. Generate Event Model for existing EDA Event Type
 2. Create external Event Rule assigned to a Rule Set other than the Default Rule Set
- Service Development perspective:
 3. Create JMS Trigger for external Event Type
- IS Administration Console
 4. Check JMS Trigger



Run Time:

5. Emit EDA Events by any Event Provider to be received by Business Rules via IS Trigger

Software AG Training | 15 - 34

Notes:

1. Generate Event Model

- Add Event Model to Rules Project:

The screenshot shows two windows side-by-side. On the left is the 'New Event Model' dialog, which has a title bar 'New Event Model' and a main area titled 'Event Model' with the sub-instruction 'Create a new event model.' It contains fields for 'Event model name:' (set to 'FraudEvent'), 'Rule project:' (set to 'DemoRuleProject'), and 'Description:' (containing the text 'External Event model indicating a detected credit card fraud.'). At the bottom are buttons for '?', 'Back...', 'Next >', 'Finish', and 'Cancel'. On the right is the 'Select Event Type' dialog, which has a title bar 'Select Event Type' and a main area titled 'Select an event type to be used to create the new event model.' It features a scrollable list of 'Matching items:' including various event types like 'EnqueuedDocument', 'EventTypeDeploymentChange', 'Feedback', and 'FraudeEvent - [Credit/FraudEvent]'. Below the list is a text input field containing 'Credit/FraudEvent'. At the bottom are 'OK' and 'Cancel' buttons. A watermark 'Internal Use Only' is diagonally across the background.

- Browse shows all available Event Types in Workspace or VCS
- Import Event Type into Designer beforehand if lacking
- No connection required

Notes:

Resulting Event Model

- Event Model added to Rules Explorer view (NOT Solutions view)
- Embedded structures generated as additional Event Models on the fly
- Origin (Event Type) is kept in Event Model
 - Allows later sync
- Rename, if desired

Notes:

Software AG

2. Create External Event Rule

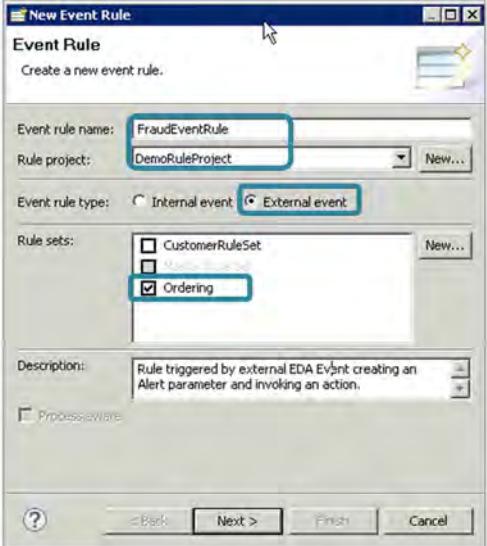
- Event Rules are Decision Entities
 - Belong to a Rule Project
 - Belong to a user-defined Rule Set (mandatory for external Event Rules)
 - Can be internal or external
- External Event Rules get triggered by an external Event emitted via Event Messaging
 - Corresponding external Event Type described by an Event Model in the Rule Project
- Create External Event Rule:
 - a) Name and description
 - b) Parameters with type and direction
 - c) Output parameter elements for results
 - d) *Optional:* Result actions
 - e) Add Rule content (assignments, actions)

The diagram illustrates the 'Create External Event Rule' interface. It features a light blue header bar with the title. Below it is a large light blue rectangular area containing a list of steps. To the right of this area is a vertical blue bracket labeled 'structure' at the top and 'content' at the bottom. Inside this bracket is a small icon of a document with a yellow starburst. The entire interface is set against a background watermark that reads 'Internal Use Only' diagonally.

Software AG Training | 15 - 37

Notes:

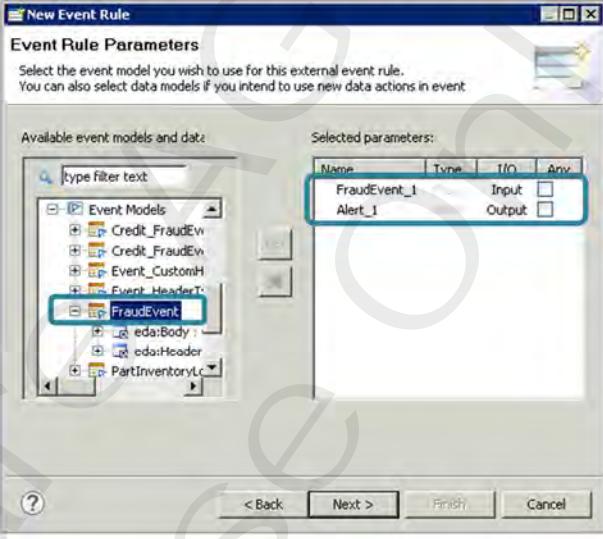
2a..d: Create External Event Rule Structure...



The screenshot shows the 'New Event Rule' dialog with the following settings:

- Event rule name: FraudEventRule
- Rule project: DemoRuleProject
- Event rule type: External event
- Rule sets: CustomerRuleSet, Ordering (selected)
- Description: Rule triggered by external EDA Event creating an Alert parameter and invoking an action.

Event Rule Parameters



The screenshot shows the 'Event Rule Parameters' dialog with the following configuration:

- Available event models and data: Event Models > FraudEvent (selected)
- Selected parameters:

Name	Type	I/O	Any
FraudEvent_1	Input	<input type="checkbox"/>	<input type="checkbox"/>
Alert_1	Output	<input type="checkbox"/>	<input type="checkbox"/>

Software AG Training | 15 - 38

Notes:

... 2a..d: Create External Event Rule Structure

The screenshot shows the 'New Event Rule' dialog box. The title bar says 'New Event Rule'. The main area is titled 'Event Rule Assignment Results' with the sub-instruction 'Select parameter elements for assignment results.' Below this, there are two sections: 'Available parameter elements:' and 'Selected parameter element:'.

In the 'Available parameter elements:' section, there is a tree view under 'Alert_1 : Alert'. The expanded node '_env : pub_publish_envelope' contains three items: 'AlertReason : String', 'AlertType : String', and 'OrderNo : String'. There are also '<' and '>' buttons for navigating through the list.

In the 'Selected parameter element:' section, there is a table with columns 'Label' and 'Parameter El...'. It lists three items: 'AlertReason', 'AlertType', and 'OrderNo'. The 'AlertReason' row is highlighted with a blue border.

At the bottom of the dialog are buttons: '?', '< Back', 'Next >', 'Finish', and 'Cancel'.

To the right of the dialog, there is a list of notes:

- Select Parameter Elements used as Rule results
- Rename label, if desired
- Subsequent panel (*not displayed*): Optionally preselect Action to be invoked

Software AG Training | 15 - 39

Notes:

2e: Adding Content to Event Rule – Event Rule Editor

Add new Event Rule Assignment to the end of the Results list

Add a new action to the end

Clear selected elements

Delete selected elements

Collapse/Show description

Assignment – Add a new Assignment

Action – Add a new Action

Cut, copy and paste

ONE rule

Description
Rule triggered by external EDA Event creating an Alert parameter
Invoking an action

FraudEvent occurred

Results

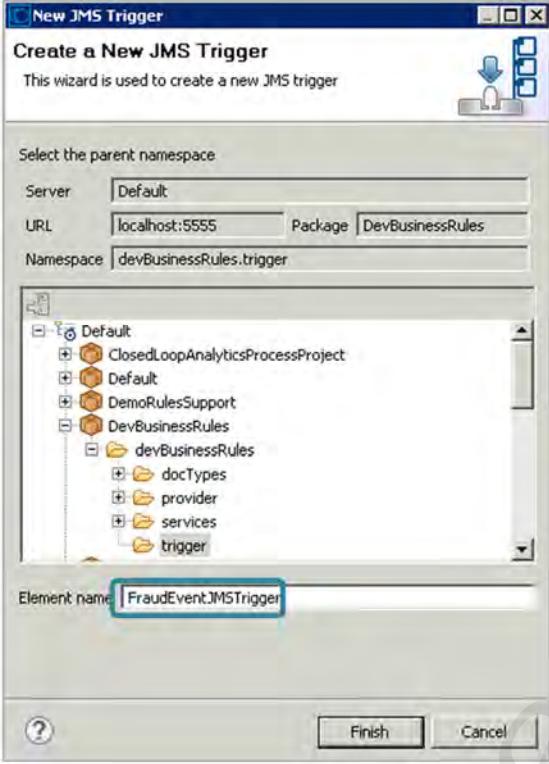
1	AlertReason	= Fraud detected
2	AlertType	= CRITICAL
3	OrderNo	= p:CCNumber
4	MailAlert()	✓

Software AG Training | 15 - 40

- Use appropriate icons in Event Rule Editor panel or use icons from Palette to customize Event Rule:
 - Add/modify Rule
 - Add/remove/rename results (assignment results, action results)

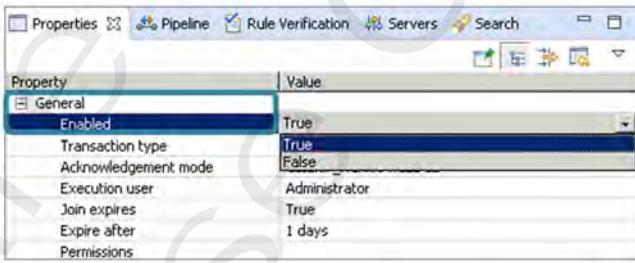
Notes:

3. Create JMS Trigger for External Event Type...



The screenshot shows the 'New JMS Trigger' wizard interface. It includes fields for Server (Default), URL (localhost:5555), Package (DevBusinessRules), and Namespace (devBusinessRules.trigger). A tree view shows the project structure under 'Default'. The 'Element name' field contains 'FraudEventJMSTrigger', which is highlighted with a blue border.

- Create JMS Trigger in an IS Package
- Trigger is used to subscribe to an JMS Topic (= external Event Type) and to pass any received Event to the Rule Engine
- Trigger can be activated/deactivated:



Property	Value
General	
Enabled	True
Transaction type	True
Acknowledgement mode	False
Execution user	Administrator
Join expires	True
Expire after	1 days
Permissions	

Software AG Training | 15 / 41

Notes:

...3. Create JMS Trigger for External Event Type

- Add lacking JMS Connection Alias, Event Destination (Topic), and IS Message Routing service to JMS Trigger

JMS connection alias name: **DEFAULT_IS_JMS_CONNECTION**

JMS trigger type: Standard

JMS destinations and message selectors

Destination Name	Destination Type	JMS Message Selector
Event::Credit::FraudEvent	Topic	

Join type: All(AND) Any(OR) Only one(XOR)

Message routing

Name	Service	Local Filter
toRuleEngine	wm.businessrules.eda:jmsTriggerService	

Select a JMS connection alias for devBusinessRules.trigger:FraudEventJMSTrigger

OK Cancel

Destination List

Destination Name	Destination Type
ClosedLoopAnalyticsProcessProject_EscalationProcess_SUBQUEUE...	Queue
ClosedLoopAnalyticsProcessProject_EscalationProcess_TRANSQUEU...	Queue
Event::Credit::FraudEvent (Event::Credit::FraudEvent)	Topic
Event::WebM::Sample::InventoryMgmt::1.0::PartInventoryShortag...	Topic
Event::WebM::Sample::OrderManagement::1.0::PurchaseOrderCre...	Topic
Event/WebM/Process/V2_0/ProcessInstanceChange (Event/WebM/...	Topic

OK Cancel Create New Destination

jmsTriggerService provided by
webMethods Business Rules; converts and
passes Event to running Rule Engine.

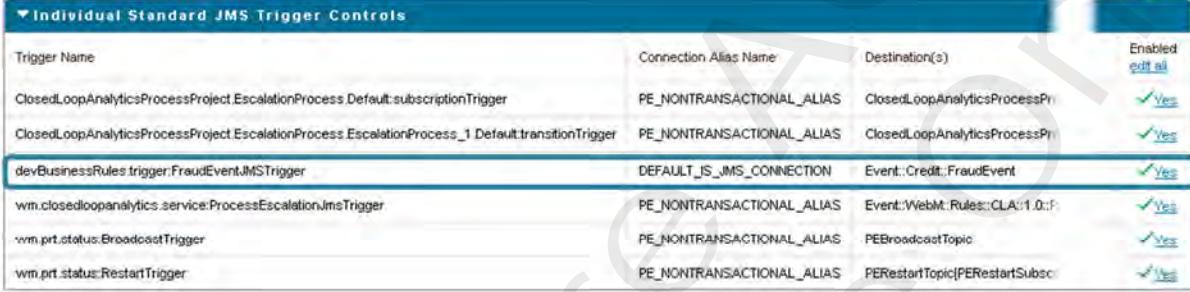
Software AG Training | 15 - 42

Using a JMS Trigger bypasses Event Routing and uses direct JMS connectivity as defined in IS JMS Connection Alias.

Allows to use predefined Trigger Handling Service to pass data into Rule Engine.

4. Check JMS Trigger in IS Administration Console

- **Optional:** Use IS Administration Console to
 - Verify JMS Trigger is enabled after saving the Trigger definition in Designer
 - Activate/deactivate the JMS Trigger
- Settings > Messaging > JMS Trigger Management



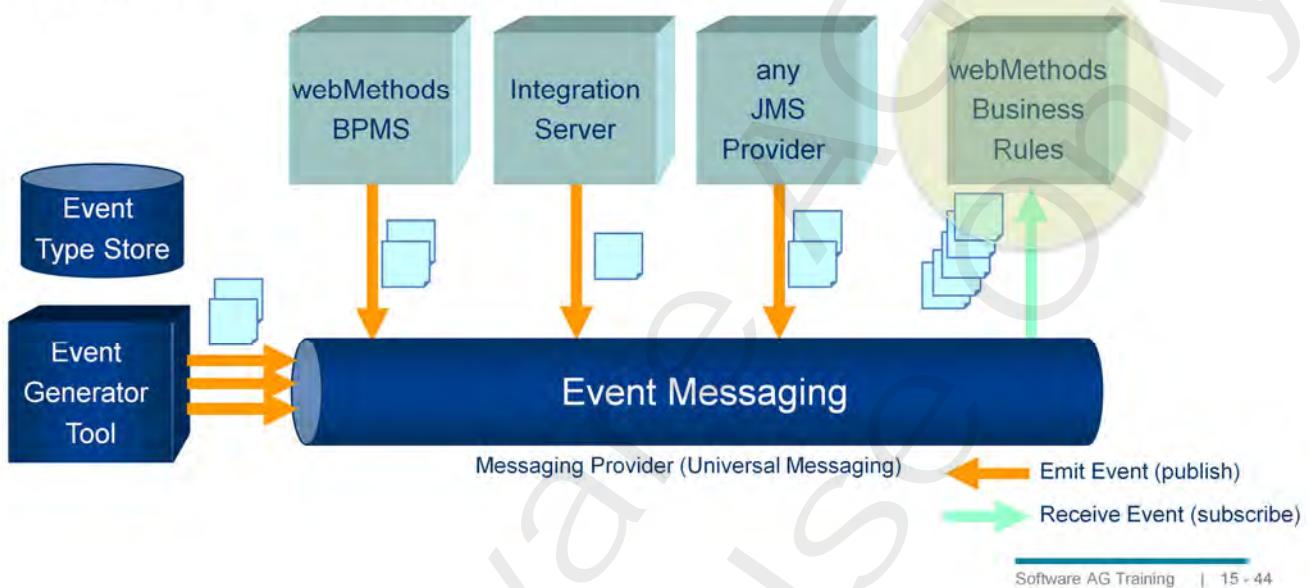
Individual Standard JMS Trigger Controls			
Trigger Name	Connection Alias Name	Destination(s)	Enabled
ClosedLoopAnalyticsProcessProject.EscalationProcess.Default:subscriptionTrigger	PE_NONTRANSACTIONAL_ALIAS	ClosedLoopAnalyticsProcessPr...	<input checked="" type="checkbox"/> Yes
ClosedLoopAnalyticsProcessProject.EscalationProcess_EscalationProcess_1 Default:transitionTrigger	PE_NONTRANSACTIONAL_ALIAS	ClosedLoopAnalyticsProcessPr...	<input checked="" type="checkbox"/> Yes
devBusinessRules.trigger.FraudEventJMSTrigger	DEFAULT_IS_JMS_CONNECTION	Event::Credit;FraudEvent	<input checked="" type="checkbox"/> Yes
wm.closedloopanalytics.service.ProcessEscalationJmsTrigger	PE_NONTRANSACTIONAL_ALIAS	Event::WebM:Rules::CLA::1.0::F...	<input checked="" type="checkbox"/> Yes
wm.prt.status.BroadcastTrigger	PE_NONTRANSACTIONAL_ALIAS	PEBroadcastTopic	<input checked="" type="checkbox"/> Yes
wm.prt.status.RestartTrigger	PE_NONTRANSACTIONAL_ALIAS	PERestartTopic PERestartSubsc...	<input checked="" type="checkbox"/> Yes

Software AG Training | 15 / 43

Notes:

Testing an External Event Rule

- Like internal Event Rules, external Event Rules can't be started/tested standalone in Designer
- To test an external Event Rule, EDA Events have to be emitted
- Requires for an EDA Event Provider



Notes:

Event Providers to Test an External Event Rule

- EDA-enabled webMethods products can act as Event provider:
 - IS application using built-in public service in pub.event.routing.send
 - webMethods BPMS processes
 - ...
- Software AG Event Generator Tool (not supported)

Notes:

Example: Emit Events using IS Service

- Public service `pub.event.routing:send` can be used to emit an Event via Event Messaging and Event Routing

Integration Server
JMS
EDA props

Corresponding IS doctype namespace

Event Type Namespace

Assign Event payload here

Software AG Training | 15 - 46

Notes:

Example: Emit Events using the Event Generator Tool*

The screenshot shows the Software AG Event Generator Administration interface with three main panels:

- Event Generator:** Shows a message "Generator FraudEventATMtoBus stopped successfully".
- Connection:** Shows a connection named "EventBus" of type "EDAConnection" with the following settings:


```
InitialContextFactory=com.webmethods.jms.naming.WmJmsNamingCtxFactory
ProviderURL=wjmnsnaming://Broker #1@localhost:6849
SecurityPrincipal=OtherProperties=ConnectionFactory=EventFactory
EDAJMSComponentDelimiter=:;
EventTypeStore=C:/SoftwareAG/common/EventTypeStore_EventBus/libraryFolder=
```
- Event Types:** Shows an event type named "FraudDemoATM" of type "EDAEEvent" with the following settings:


```
EDAEEventType=
{http://namespaces.softwareag.com/EDA}FraudDemoATM
TopicName=Event::FraudDemoATM
```

* not an official Software AG product/component yet

Software AG Training | 15 - 47

Notes:

Send Events from webMethods Business Rules

Currently no direct path – but you can leverage the existing Events support of an IS:

- First create an IS Service that invokes pub.event.routing.send
- Add a Service Action to your Rule Project that invokes the IS service from above
- Invoke the Service Action from any Decision Entity in your Rule project

The diagram illustrates the workflow for sending events. At the top, a blue cube labeled "webMethods Business Rules" contains a "Service Action" icon. A vertical line connects this to a tree view of rule projects under "Default". The tree includes "ClosedLoopAnalyticsProcessProject", "Default", "DemoRulesSupport", and "DevBusinessRules". Under "DevBusinessRules", there is a folder "devBusiness_Rules" containing "docTypes", "provide", and "zynapse". Within "provide", a file named "emitFraudEvent" is highlighted with a green checkmark. A blue callout box at the bottom left contains the text "pub.event.routing.send" with a right-pointing arrow icon. An arrow points from this callout box to the "emitFraudEvent" file in the tree view.

Software AG Training | 15 - 48

Notes:



16

Business Rule Metadata



Notes:

Objectives

At the end of this chapter you ...

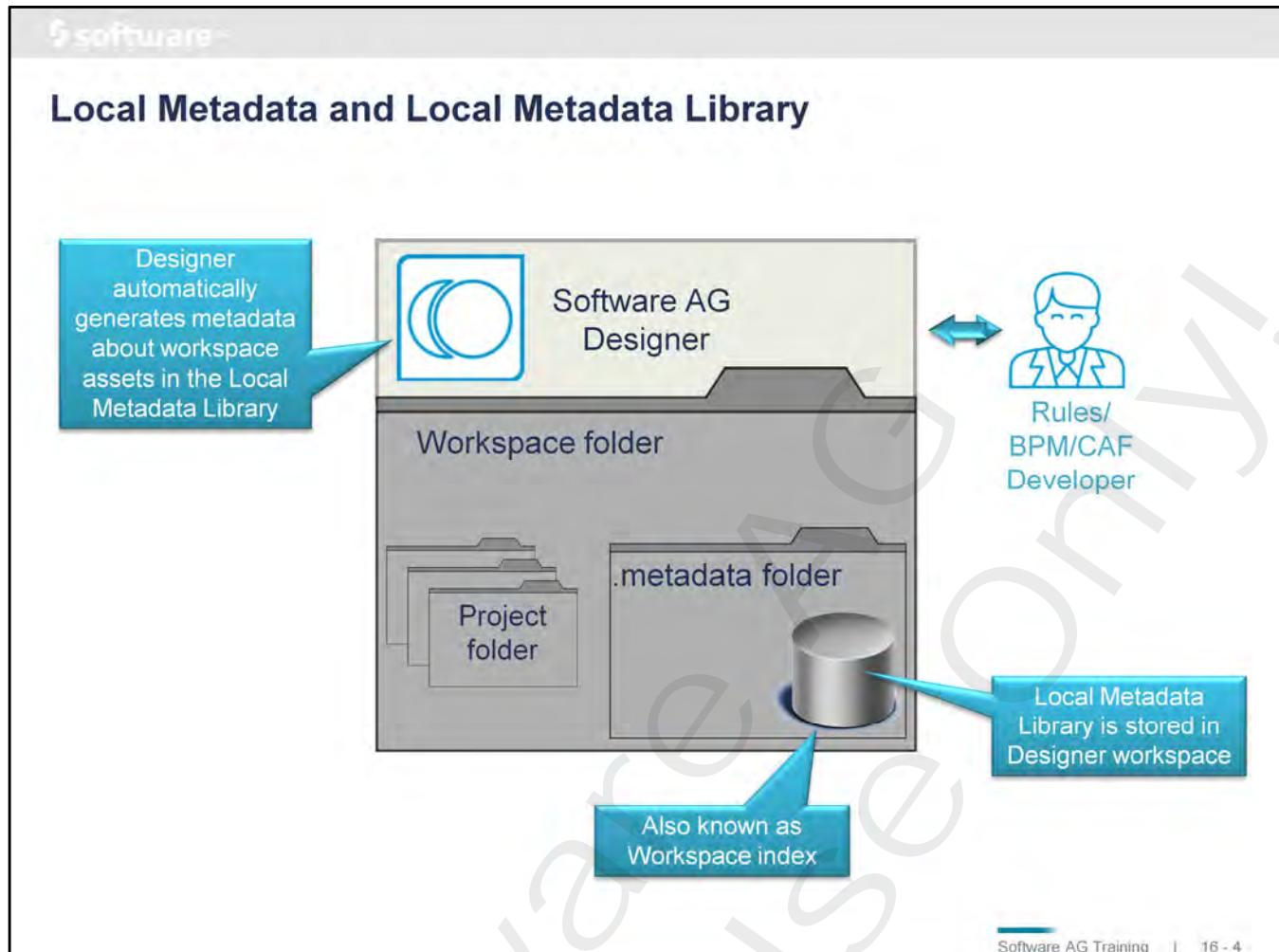
- Can explain the benefits of Business Rules metadata
- Know the difference between local and shared metadata
- Created and executed Saved Metadata Searches
- Published and retracted Business Rules Metadata to/from CentraSite
- Performed an impact analysis

Notes:

Designer Metadata: Assets

- Designer can be configured to generate metadata about workspace assets
- Regarding Business Rules and BPM, Designer tracks the following assets:
 - **Business Rule projects**
 - Process projects
 - Processes and process references
 - Composite application framework (CAF) projects
 - Portlets and portlet views
 - User Tasks and User Task references
 - IS and TN Document Type references
 - IS Service references
 - ...

Notes:



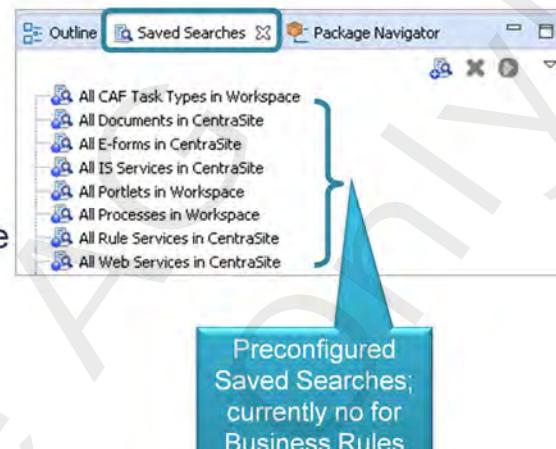
Notes:

Metadata and Saved Searches

- (Local) Metadata is leveraged for Saved Searches
- Designer provides multiple preconfigured Saved Searches
 - Preconfigured Saved Searches cannot be deleted or changed
- Custom Saved Searches may be added
- Custom/preconfigured Saved Searches are
 - created
 - viewed
 - edited
 - executed

in Saved Searches view

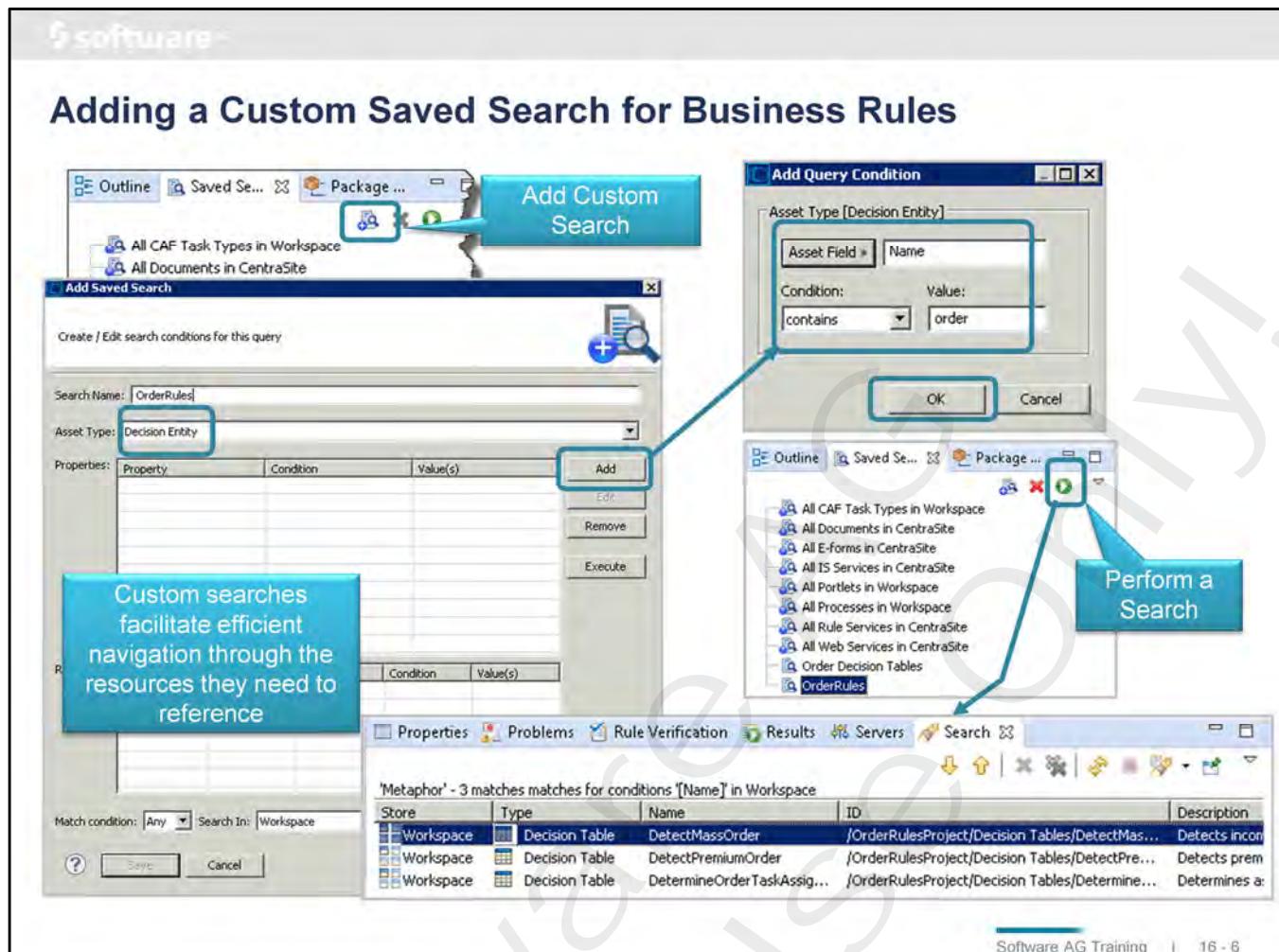
- Rule Developers and Process Developer can leverage this information to determine where assets are used



Preconfigured Saved Searches; currently no for Business Rules

Software AG Training | 16 - 5

Notes:



Notes:

If I change this Decision Table, what effect will it have on other assets?

Store	Type	Name	ID	Description
Workspace	Decision Table	DetectMassOrder	/OrderRulesProject/Decision Tables/DetectMas...	Detects incoming mass orders by che...
Workspace	Decision Table	DetectPremiumOrder	/OrderRulesProject/Decision Tables/DetectPremium...	Detects premium order based on num...
Workspace	Decision Table	AssigneeTaskAss	/OrderRulesProject/Decision Tables/AssigneeTaskAss...	Determines assignee for an Order U...

- Results of a Saved Search can be leveraged for:
 - showing Dependencies
 - showing References

Software AG Training | 16 - 7

Notes:

Software AG Training | 16 - 8

Show Dependents and References

- Developers leverage dependents and references to gain a deeper understanding of asset relationships
- Sample: Dependencies of *DetectMassOrder* Decision Table:

Run the current Search again (F5)

Show previous Searches

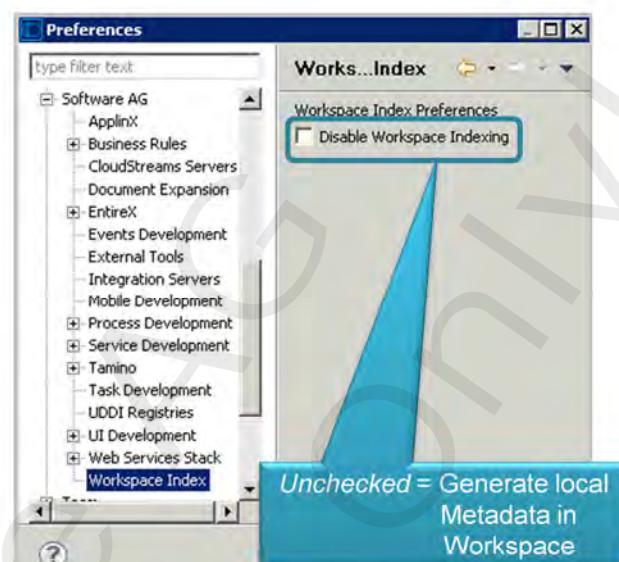
Direct relationships are relationships that are one link apart,
e.g. DT *DetectMassOrder* directly uses Rule Parameter *PurchaseData*

Direct	Type	Name	Full Name	Description
RuleParameter	ProcessData	urn:///OrderRulesProject/...	urn:///OrderRulesProject/...	
RuleParameter	PurchaseOrder_1	urn:///OrderRulesProject/...	urn:///OrderRulesProject/...	
RuleAction	startApproveMassOrderTask	urn:///OrderRulesProject/...	urn:///OrderRulesProject/...	Starts an instance of User Task t...
RuleParameter	PurchaseOrderApproval_1	urn:///OrderRulesProject/...	urn:///OrderRulesProject/...	

Notes:

Generate Local Metadata - Enablement

- Generation of local metadata is enabled by default
- Use Designer preferences to disable metadata generation in Local Metadata Library (Workspace Index), if it is not being used
- This may marginally increase the performance of Designer



The screenshot shows the 'Preferences' dialog box. In the left pane, under 'Software AG', 'Business Rules' is expanded, and 'Workspace Index' is selected. In the right pane, under 'Workspace Index Preferences', there is a checkbox labeled 'Disable Workspace Indexing'. A blue callout box points to this checkbox with the text 'Unchecked = Generate local Metadata in Workspace'.

Notes:

Software AG Training | 16 - 10

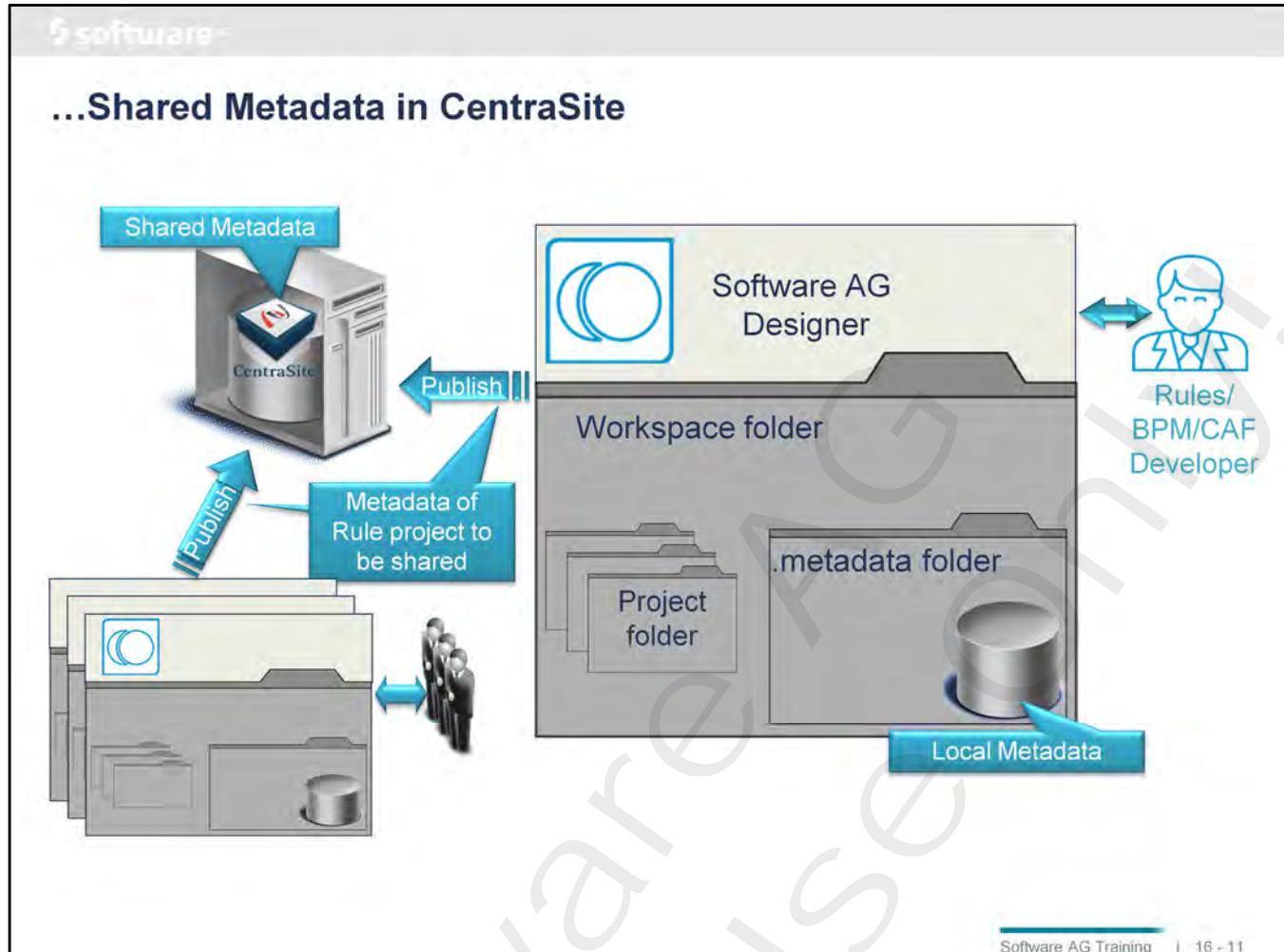
Shared Metadata in CentraSite...

- Shared Metadata can be created in a common CentraSite registry/repository
 - CentraSite is typically launched on a shared network node
 - Access to CentraSite configured using Designer preferences:

The screenshot shows the 'Preferences' dialog for CentraSite. On the left, under 'CentraSite' in the tree view, 'Connections' is selected. A sub-dialog titled 'Edit Connection Configuration' is open in the foreground, showing fields for Name (Local CentraSite), Host (http://localhost:53305/INMConfiguration/INMConfiguration.xml), User (Administrator), and Password (*****). The 'Test' button is highlighted with a blue box. In the background, the main 'Connections' dialog lists a single configuration: Local CentraSite, URL localhost, User Administrator. The 'Edit...' button for this entry is also highlighted with a blue box. A smaller 'Edit Connection Configuration' dialog is visible in the bottom right, stating 'Connection configuration data successfully validated.' with an 'OK' button.

- Access from multiple Designer instances supports team development

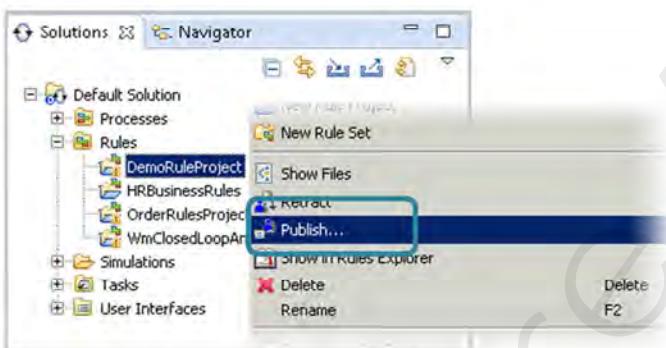
Notes:



Notes:

Publishing Business Rules Metadata to CentraSite...

- Creation of shared metadata is on demand
 - Nothing gets published to CentraSite automatically
- Use Publish to share metadata about entire Rule project:



The screenshot shows the 'Navigator' view of the webMethods Business Rules application. On the left, there's a tree view of 'Solutions' containing 'Default Solution' with various projects like 'DemoRuleProject', 'HRBusinessRules', 'OrderRulesProject', and 'WmClosedLoopA'. A context menu is open over the 'DemoRuleProject', with the 'Publish...' option highlighted by a blue rectangle. Other options in the menu include 'New Rule Set', 'Show Files', 'Rebuild', 'Delete', and 'Rename'. At the bottom right of the interface, it says 'Software AG Training | 16 < 12'.

Notes:

- CentraSite Registry is a catalog of reusable assets within a shared development environment
- Registry can be browsed via Registry Explorer view:
 - Each entry represents a single asset
 - Context menus contain a set of actions available for the selected asset, e.g.
 - Impact Analysis
 - Delete
 - ...

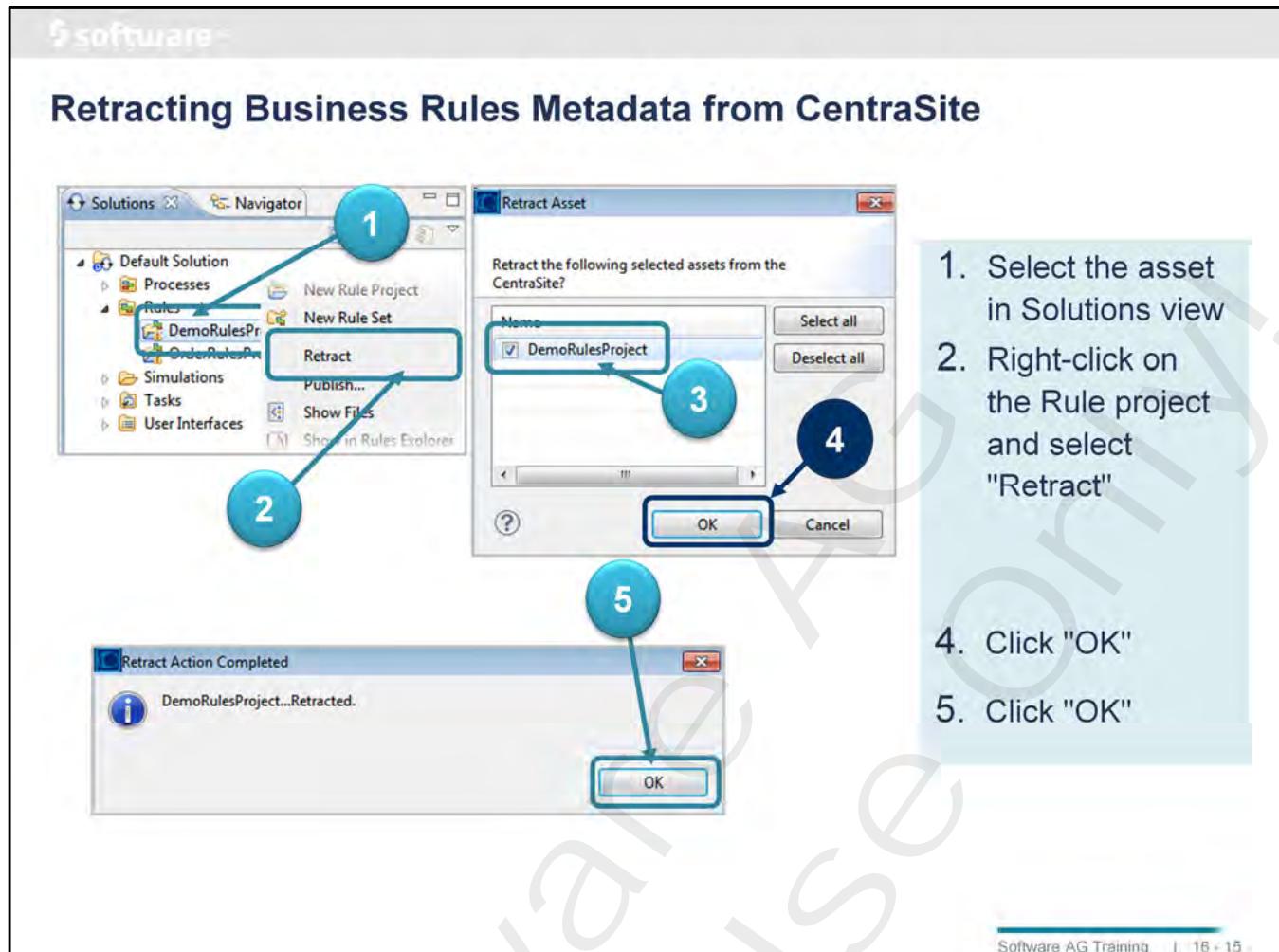
Software AG Training | 16 - 13

Notes:

Impact Analyzer View

- Impact Analyzer view provides a textual and graphical view of other assets related to the Rule project
- Impact Analyzer view can be used to navigate dependency trees quickly and efficiently to determine a deeper understanding of impact analysis
 - Assets can be collapsed
 - Assets can be expanded
 - Assets to be displayed can be filtered
- Only available for shared metadata published to CentraSite

Notes:



Notes:



Exercise 18

- In this exercise, you will use Software AG Designer to create and run a saved search against the local metadata.

Notes:



17

Logging, Auditing and Administration

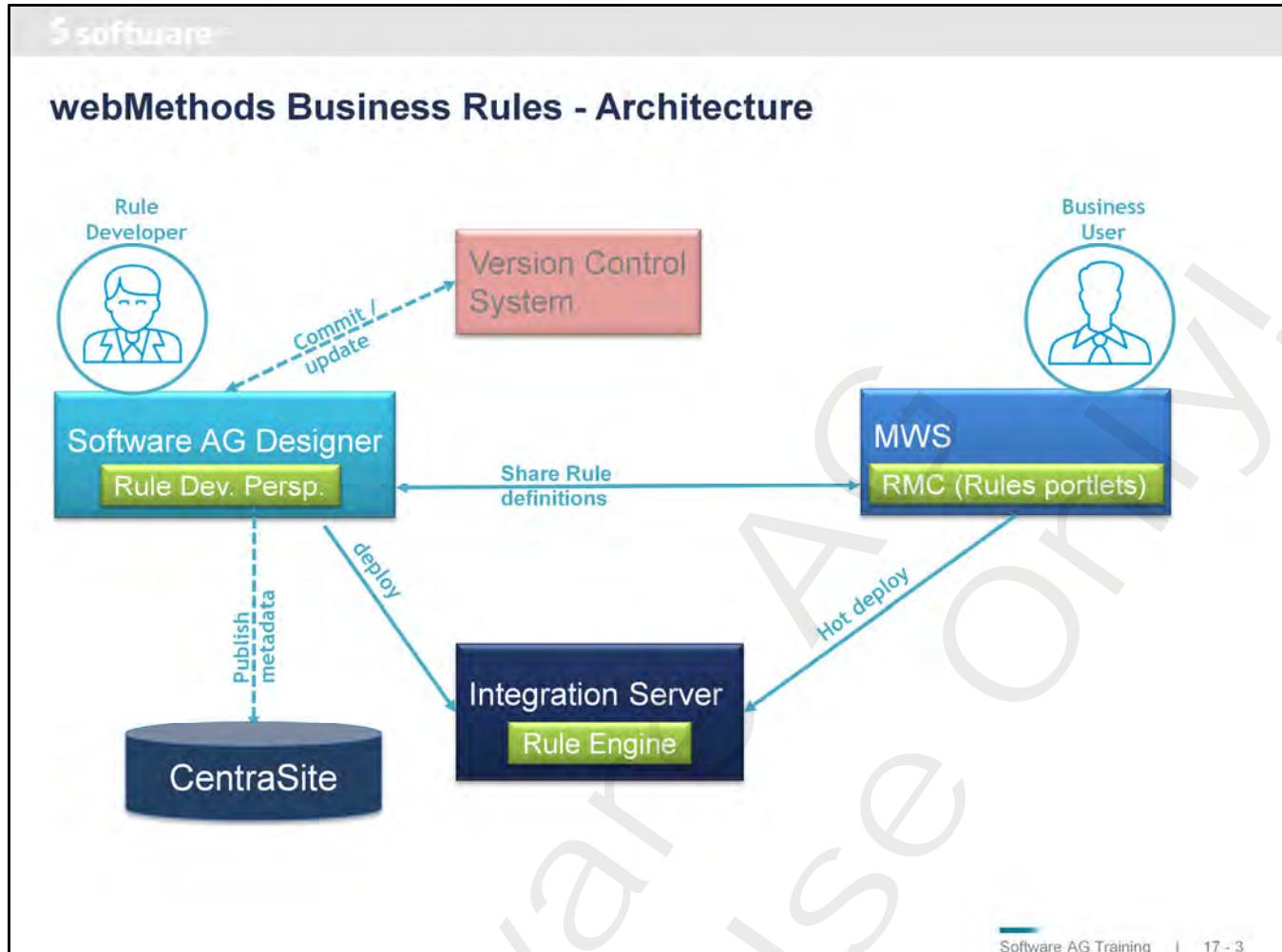
Notes:

Objectives

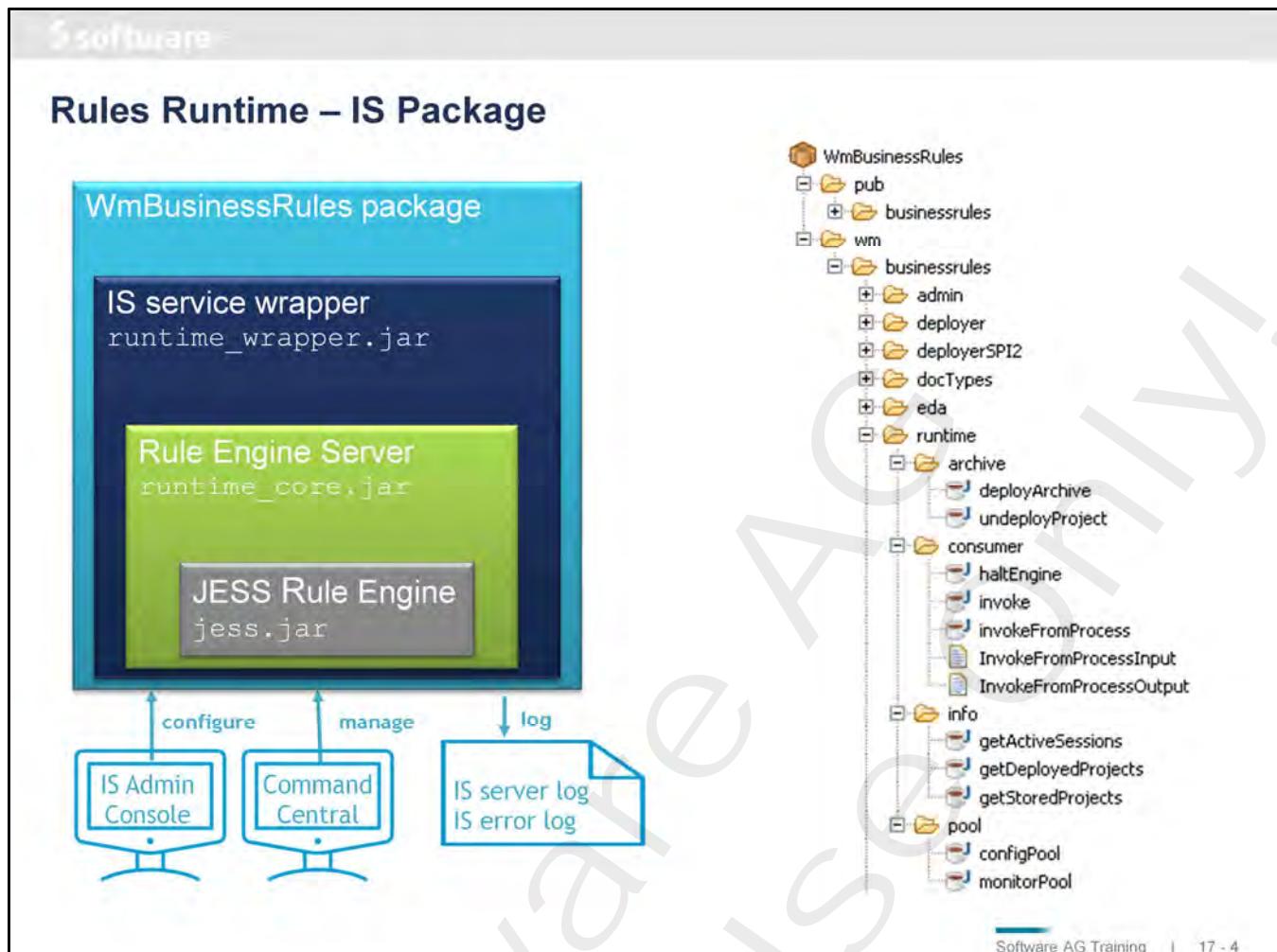
At the end of this chapter you ...

- Understand the Business Rules licensing concept
- Can locate the Business Rules-related logging data
- Set the Rule Engine logging level in an IS and Designer
- Know how to perform custom design-time auditing
- Are able to administer and monitor Business Rules components using Command Central
- Know about the Business Rules Support in Deployer

Notes:



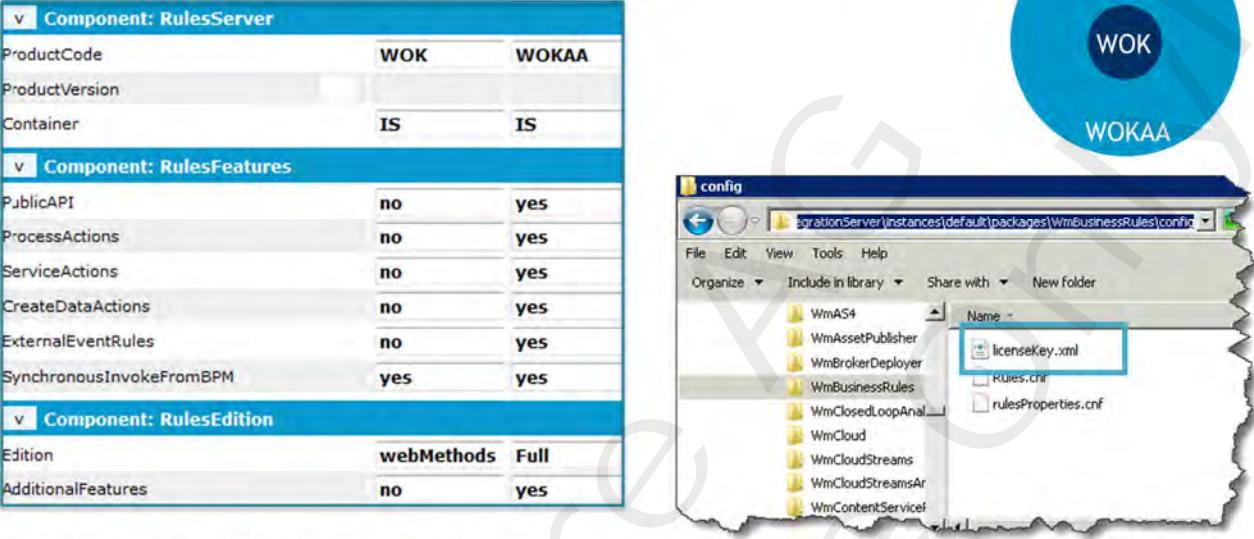
Notes:



Notes:

License Keys

- In V9.x, two license key types are supported: WOK, WOKAA



Component: RulesServer	
ProductCode	WOK
ProductVersion	WOKAA
Container	IS

Component: RulesFeatures	
PublicAPI	no
ProcessActions	yes
ServiceActions	no
CreateDataActions	no
ExternalEventRules	no
SynchronousInvokeFromBPM	yes

Component: RulesEdition	
Edition	webMethods
AdditionalFeatures	Full

- Exchange license key in folder ...\\config of WmBusinessRules package to alter supported features

Software AG Training | 17 - 5

Business Rules checks if unauthorized functionality is attempted to be executed. Error message is raised in the IS log.

Logging in webMethods Business Rules

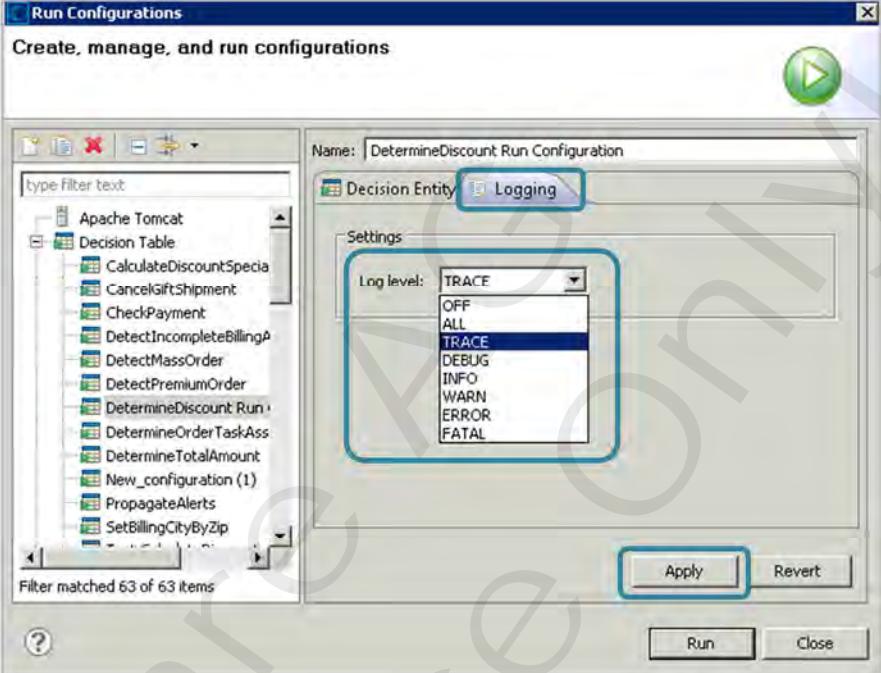
- The Rules runtime produces log output
- Can be seen in
 - Eclipse Console view
 - IS server log
- Different log levels produce different verbosity
 - FATAL, ERROR, WARN, INFO, DEBUG, TRACE, NONE



Notes:

Setting the Log Level in Designer

- Run Configurations allow for setting the log level
- Affects rule execution from Designer

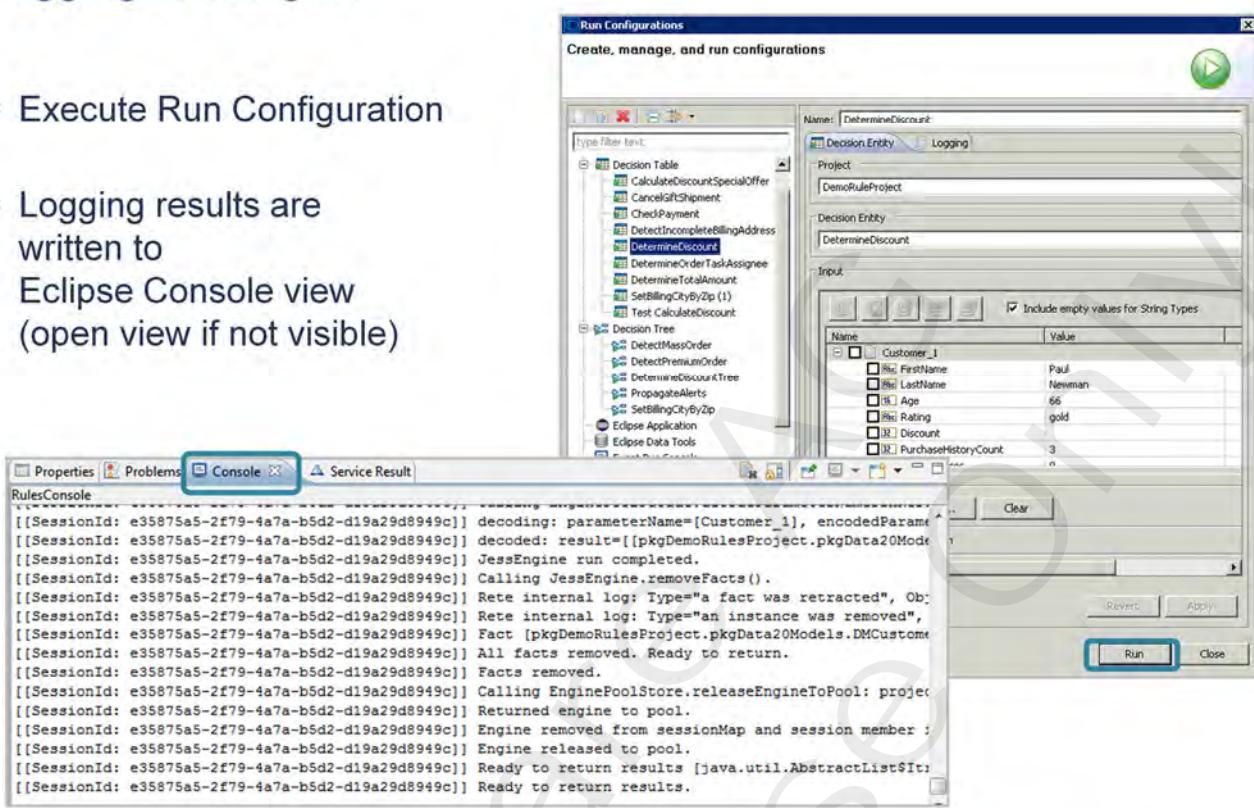


Software AG Training | 17 - 7

Notes:

Logging in Designer

- Execute Run Configuration
 - Logging results are written to Eclipse Console view (open view if not visible)



Software AG Training | 17 - 8

Notes:

Setting the Log Level in Integration Server

The screenshot shows the 'Server Logger Configuration' page in the webMethods Integration Server Administration Console. The 'Business Rules' section is selected, displaying the following logging levels:

Facility	Logging Level
Default	Info
Integration Server	Info
Optimize Packages	Info
WmPRT Package	Info
Business Rules	Debug
0005 Common	Debug
0010 Core	Debug
0050 Package	Debug
Adapters	Info

NOTE: GRAY text signifies inherited from parent.

Log Timestamp Format:
Format: yyyy-MM-dd HH:mm:ss z
Example Format: yyyy-MM-dd HH:mm:ss
Save Changes

Notes:

Custom Design-Time Auditing

- Business Rules emits additional EDA Events to allow a custom design-time auditing of rule changes
- Following EDA Events are emitted by RMC and IS hosting a Rule runtime:

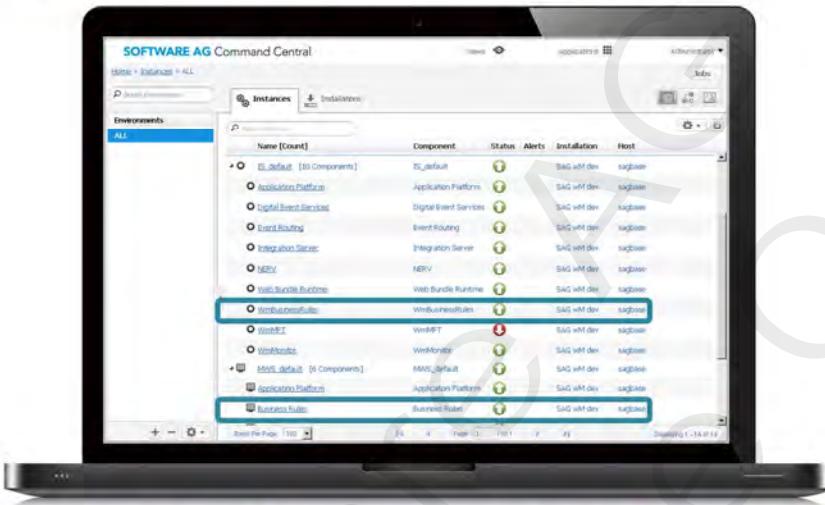
EDA Event type	Emitted by	Description
Decision Entity Changed	MWS (RMC)	Is triggered when a difference is detected between two equally named Decision Entities
Hot Deployment Started	MWS (RMC)	Is triggered when you hot deploy a Rule Project from a RMC to Integration Server runtime(s)
Project Deployed	Integration Server	Is triggered when a Rule Project is deployed on the Integration Server runtime
Project Undeployed	Integration Server	Is triggered when a rule project is undeployed from the Integration Server.

- Custom auditing application has to subscribe to these EDA Events

Notes:

Command Central Integration

- Command Central can be used to monitor and administer
 - Rule Engine in an IS instance component
 - Rules Management Console in an MWS instance component



See also documentation "*webMethods Business Rules Reference*"

Software AG Training | 17 - 11

Notes:

Command Central – Rule Engine in IS

The screenshot illustrates the Software AG Command Central interface, specifically the 'WebBusinessRules' component. The left sidebar lists various components: WebBusinessRules, WebMFT, WebMonitor, and WebClient. The main area shows the 'Overview' tab selected, displaying monitoring status and actions like Stop, Pause, and Restart. The 'Configuration' tab is also visible, showing license key details for the WebBusinessRules license.

License Key Details:	
SerialNo:	0000028902
SerialKey:	E2A752987407C500F88026667F5
CustomerID:	SAG Training
CustomerName:	SAG Training
ContractDetails:	1
ContractProductNumber:	1
LicenseTypeDetails:	Internal
Autoclose:	No
Product Info:	
ExpirationDate:	2021/06/30
OS:	win
ProductCode:	WOKAA
ProductID:	
ProductName:	
ProductVersion:	Business Rules - Addon
UserCount:	9.12
RenewalDate:	
HigherLevelRole:	
HigherLevelRoleCode:	
HigherLevelModule:	
RuleServer:	
ProductCode:	WOKAA
ProductVersion:	9.12
Container:	IS
RuleFeatures:	

Command Central allows to

- Start/stop a Rule Engine in an IS instance
- View/modify the assigned license key

Notes:

The screenshot shows the Software AG Command Central interface for managing Business Rules in an MWS instance. The main window displays the 'Business Rules' component overview, including status, alerts, and monitoring information. A modal window titled 'Lifecycle Actions' is open, showing options like 'Stop' and 'Pause'. Another window titled 'Configuration' shows a confirmation message: 'The Integration Server is online.' Below these, a 'Endpoints' configuration dialog is open, displaying 'Master IS Settings' with fields for Logical Name (local), Host (localhost), Port (5555), User Name (Administrator), and Password (redacted). The bottom right corner of the slide indicates 'Software AG Training | 17 - 13'.

Notes:

...Command Central – Rules Management Console in MWS

The image displays three separate windows of the Software AG Command Central interface, each showing a 'Configuration' step with a 'Confirmation' message:

- Business verification:** Shows configuration for 'MWS_default' with host 'localhost', port '5555', protocol 'HTTP', and base path 'rest'. Authentication method is 'Basic'.
- Data provider:** Shows configuration for 'MWS_default' with host 'localhost', port '5555', protocol 'HTTP', and base path 'rest'.
- Principal types:** Shows configuration for 'MWS_default' with 'Principal types' selected. It lists 'Users', 'Groups', and 'Roles' with checkboxes checked.

Command Central allows to

- Configure REST base URLs for Verification and Data Provider servers
- Configure principal types for principal picker

Software AG Training | 17 - 14

Notes:

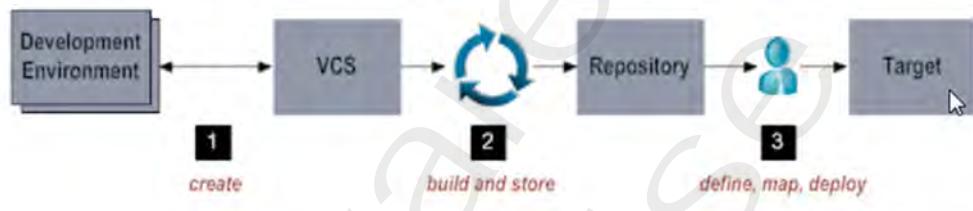
Business Rules and Deployer

- webMethods Deployer supports:
 - Runtime-based deployment - deploys assets from webMethods runtimes to which Deployer is connected
 - Repository-based deployment deploys assets built from sources in a development environment or VCS and stored on a repository
- Deployer offers Repository-based deployment for Business Rules projects to target
 - IS runtimes (Rule Engines)
 - MWS runtimes (content repositories)
- Use Asset Build Environment (ABE) to create Rule composites (.jars) in a Deployer repository

Notes:

Repository-based Deployment of Business Rules

1. Create assets in the development environment and save them on a server, or check them into a version control system (VCS).
2. Use the master build script of the Asset Build Environment (ABE) to build the composite and descriptor files from the rule projects. You store these files in the repository.
3. Use Deployer to do the following:
 - a. Define a Deployment Project and Deployment sets. Select user-created assets from the composites stored on the repository to identify the assets to include in the project.
 - b. Map the contents of the Deployment Project to IS and MWS servers
 - c. Deploy the assets in the project build to the target servers

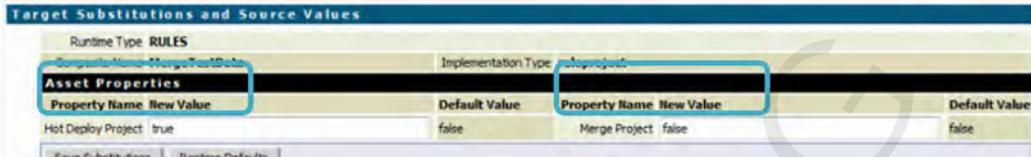


Software AG Training | 17 - 18

Notes:

MWS Deployment – Optional Hot Deploy and Merge

- Deploying a Rule project to an MWS target server using Deployer supports optional Hot Deploy and Merge



Asset Properties		Implementation Type: Deployment			
Property Name	New Value	Default Value	Property Name	New Value	Default Value
Hot Deploy Project	true	false	Merge Project	false	false

- Hot Deploy**
 - If “Hot Deploy Project” is set to “true”, the project will additionally be “hot deployed” to (Master) Integration Server(s) as defined in My webMethods
- Merge**
 - If “Merge Project” is set to true, a Decision Entity* to be deployed will be merged with the current Decision Entity in the RMC
 - Structure of Decision Entity will be taken from the Decision Entity to be deployed
 - Cell values of Decision Entity will be taken from Decision Entity in RMC

* Feature currently only available for Decision Entity of type Decision Table

Before you can hot deploy and merge a rule project using webMethods Deployer, you must configure the file system location used by webMethods Deployer for Business Rules asset deployment. To configure the file system location:

- On your file system, open:

<Installation_Folder>\profiles\<MWS_profile>\configuration\custom_wrapper.conf

- Add the path to the file system location:

wrapper.java.additional.110=-DBRMS_SPI2_PERSISTENCE_PATH=<Your_Path>

- Save the file

- Restart My webMethods Server

This page intentionally left blank.

Software AG
Internal Use Only!



18

Wrap Up

Notes:



What Should I Take Next?

Now that you have completed the “webMethods Business Rules” course, there are other courses that may be interesting for you:

- Interested in Integration Service development?
 - *webMethods Integration Workshop*
- Interested in Business Process development?
 - *webMethods BPM for Developers*

<http://www.softwareag.com> ⇒ Services ⇒ Global Education Services

Software AG Training | 18 - 2

Notes:



Certification

Our certification programs establish standards for knowledge and skills necessary to successfully implement mission-critical IT systems using Software AG technology.

- webMethods Certified ESB Developer
- webMethods Certified BPM Developer
- webMethods Certified CAF Developer

<http://www.softwareag.com> ⇒ Services ⇒ Global Education Services

Software AG Training | 18 - 3

Notes:



Further Information

- Find our Developer Communities at
<http://techcommunity.softwareag.com>
- Contact our Support web site at
<http://empower.softwareag.com>
- Submit your product and feature ideas at Brainstorm
(available within Empower)
<http://empower.softwareag.com>

Software AG Training | 18 - 4

Notes:

Software AG Training

Feedback

- Questions and Comments
- Please complete a course evaluation, ...
 - to support us matching your needs
 - **to get your certificate**



Notes:



Thank You!

Notes:
