



IBM Case Manager 5.2 Embedded Business Rules

Introduction

- Course Overview
 - Provide an overview of new Embedded Business Rules functionality
 - Describe how existing ICM 5.1.1 customers are impacted
- Target Audience:
 - IBM development, support, and services organizations that work with case management solutions
 - Personnel working with IBM Case Manager customers such as Sales System Engineers, Specialists and SMEs, Consulting Specialists, Field Technical Consultants and Technical Support
- Suggested Prerequisites:
 - Familiarity with IBM Case Manager
 - Rules familiarity
- Version Release Date: September, 2013

© Copyright International Business Machines Corporation 2013. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Course Objectives



After this course you will be able to:

- Describe the salient features of new Embedded Rules functionality
- Configure ICM to enable Embedded Rules features
- Author Text-based & Table-based business rules from Case Builder, use the rules in Tasks and verify rule execution results from Case Client
- Describe the steps involved in migrating Embedded Rules to full IODM (IBM Operational Decision Manager) product.
- Explain how ICM 5.1.1 customers may be impacted by this new feature
- Troubleshoot rules related issues in your ICM environment.
- Learn about Known Issues, Limitations, Best Practices in rules functionality.

Course Roadmap




- ➔ Introduction
 - Configuring Embedded Rules in ICM environment
 - Rules Authoring in Case Builder
 - Executing Rules as part of a workflow
 - Moving Rules from embedded environment to IODM
 - Impact to ICM 5.1.1 customers
 - Troubleshooting
 - Known Issues, Limitations, Best Practices
 - Course Summary


Introduction

- IBM Case Manager bundles IBM Business Rules Embedded in place of ILOG JRules
 - IBM Business Rules Embedded is an SDK provided by IODM that allows for tight integration into a product
 - Simplified installation
 - No separate installation of ILOG Rules Studio and Rule Execution Server
 - Rules are installed as part of ICM installation
 - Simplified authoring
 - Authoring of Rules from inside Case Builder
 - Simplified Rule Steps provided in Step Builder
 - No need to transfer rules to Rules Execution Server
 - No need to manually set rule WSDL in Process Designer

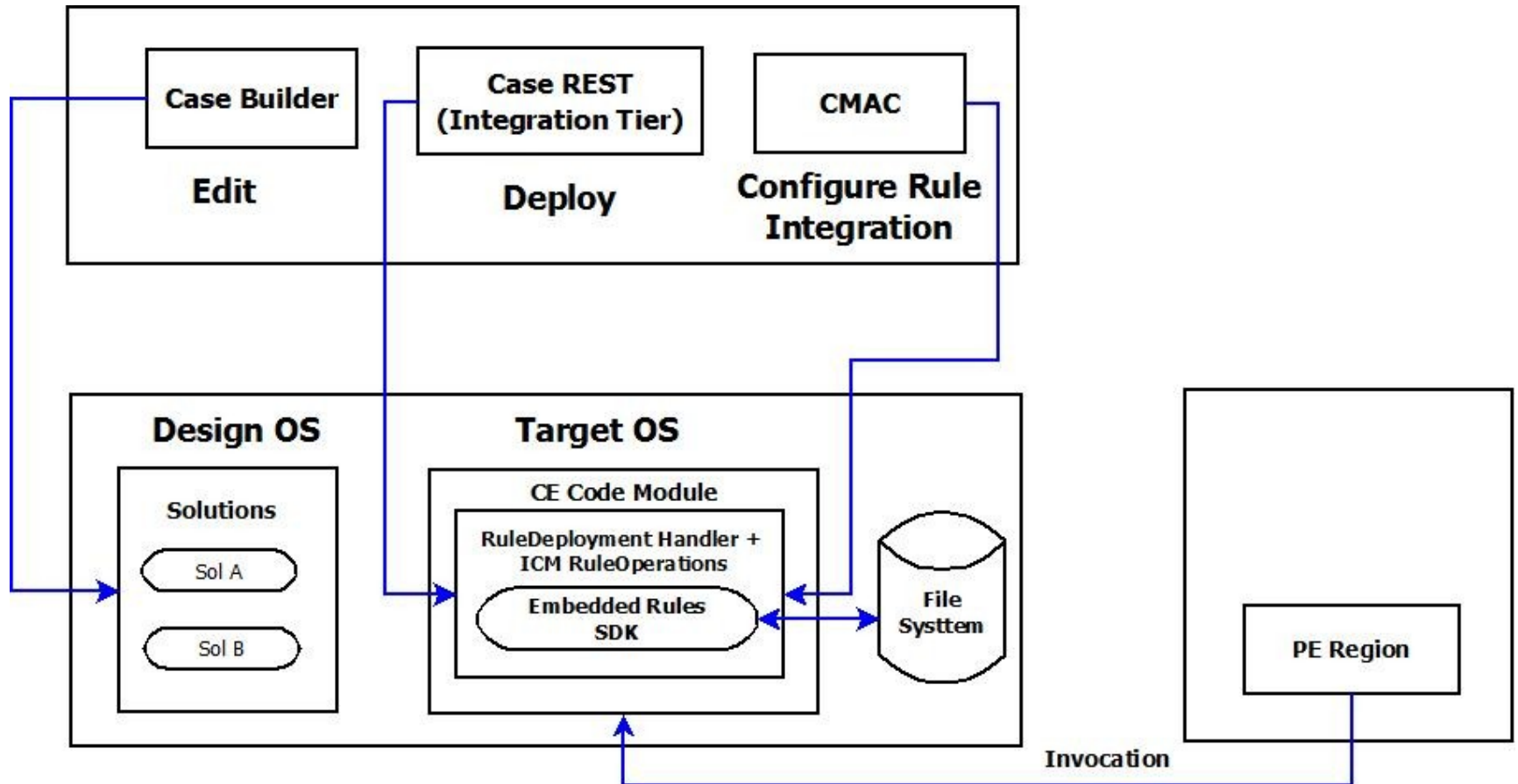
Introduction – Feature Overview

- 
- Better User Experience
 - Friendlier for the Business Analyst
 - Overall simplification for ICM solution
 - Case properties get read/updated automatically
 - Limitations in first version of ICM with Embedded Rules
 - Must redeploy entire solution to change a Business Rule
 - Business Rules only available as Task Step
 - Limited change management and governance
 - Languages supported by the IODM SDK are limited compared to what is supported in Case Builder. If the user's locale is not supported by IODM SDK, user will have to write the rule in English.

Comparison between Embedded Rules, JRules 7.1 & IODM

Capabilities	Embeddable Business Rules	Host Bundled JRules 7.1 + Rule Studio	IBM ODM	
			Decision Server	Decision Center
Rule authoring in natural language based on a business vocabulary using rules and decision tables	✓ Business users	✓ Technical users	✓ Technical users	✓ Business users
Rule authoring for business users embeddable in host application web UI allowing authoring in host solution context.	✓ Host application rule repository			✓ Decision Center repository
Business Vocabulary Generation and customization	Vocabulary generated from schema	Host defined vocabulary Customization	✓ Full vocabulary Customization	
Extended rule authoring: packages, ruleflows, decision trees, templates		✓ Technical users	✓ Technical users	✓ Business users
Support for Business event rules in natural language			✓ Technical users	✓ Business users
Rule execution as part of solution	✓ Embeddable Runtime	✓ Rule Execution server	✓ Scalable Server Decision services	
Management and Deployment of rulesets and decision services to execution runtime	✓ API for Ruleset deployment	✓ Management Console	✓ Management Console	✓ Governed Deployment
Centralized business rule repository				✓
Business rules lifecycle governance,				✓
Multiple release management for reliable change mgt				✓
Enterprise-wide collaboration on business rules, including IT/Business Alignment				

Embedded Rules Integration - Architecture



Embedded Rules Integration - Architecture



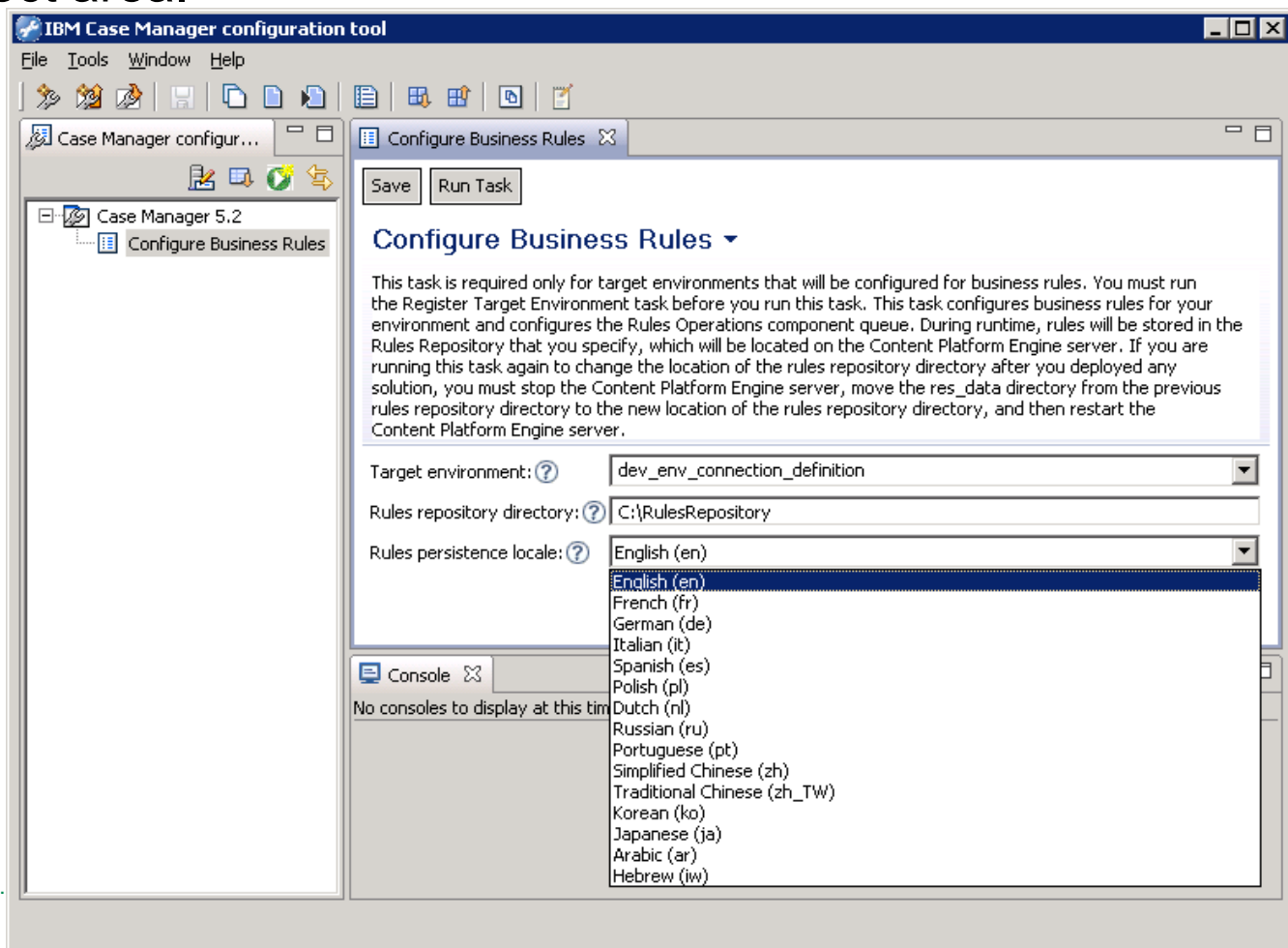
- Dojo based Rule Editor and Decision Table embedded in Case Builder
- Rule Definition artifacts are persisted in CPE (Content Platform Engine) Design Object Store and compiled ruleSet.jar is stored in file system (Rules Repository)
- Embedded SDK is hosted as code module in CPE Target Object store
- Rule runtime runs as a Java component in CPE

Course Roadmap

- Introduction
- ➔ Configuring Embedded Rules in ICM environment
- Rules Authoring in Case Builder
- Executing Rules as part of a workflow
- Moving Rules from embedded environment to IODM
- Impact to ICM 5.1.1 customers
- Troubleshooting
- Known Issues, Limitations, Best Practices
- Course Summary

Configuring Embedded Rules in ICM environment

- Rules feature is disabled by default
- Use CMAC to configure business rules. This task needs to be run for each project area.



Configuring Embedded Rules in ICM environment

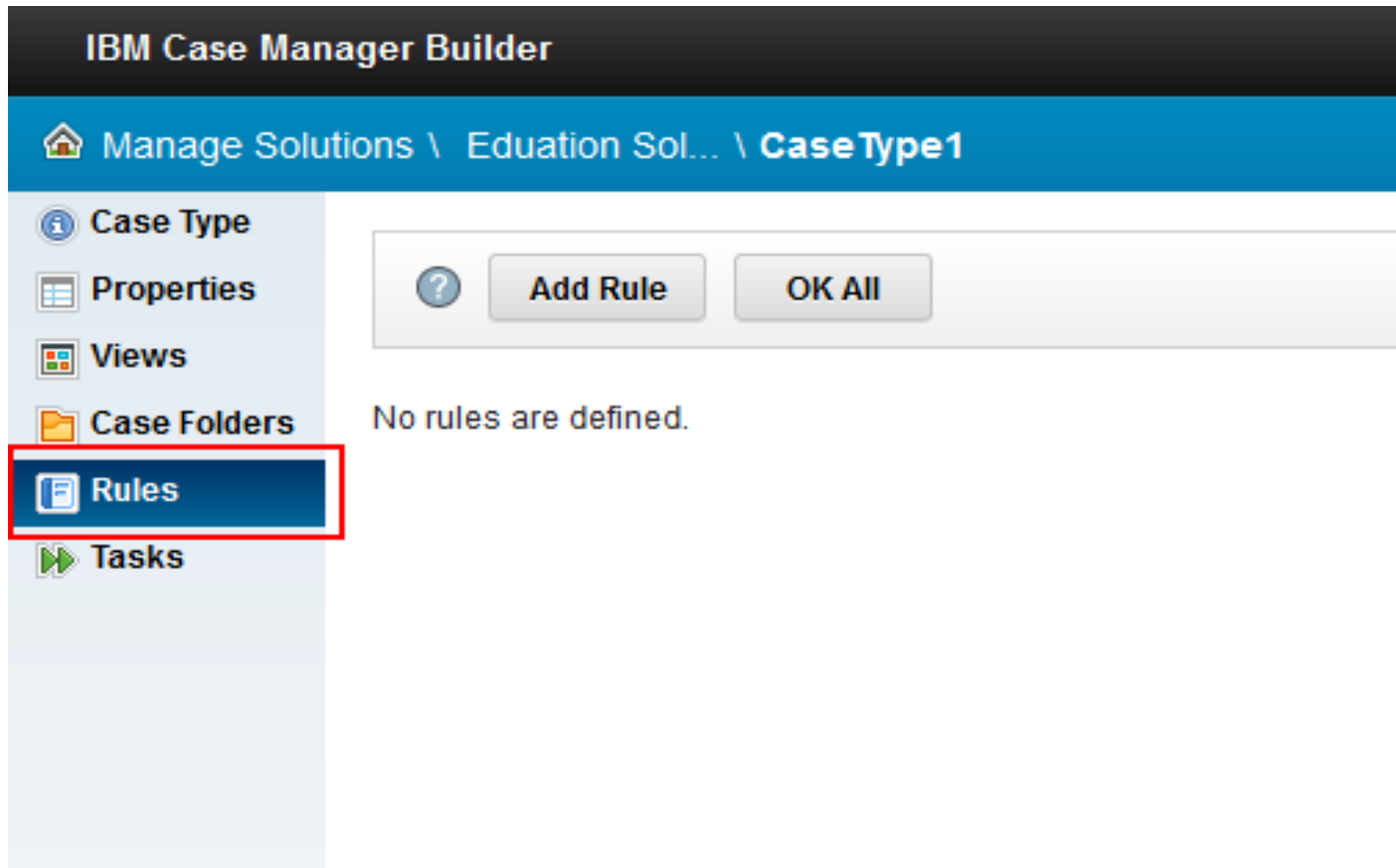
- CMAC configures ICM_RuleOperations queue and Rule Deployment code module in CPE and does additional configurations (create custom events and register event handlers)
- Rule repository path can be a local file system directory. If CPE is clustered, specify a shared directory which is accessible to all nodes.
- You can select only one locale in which to write the business rules.
 - If need to create solutions with business rules in other locales, rerun this task to change the rule persistence locale.
 - After a solution is created, its rule persistence locale cannot be changed.
 - If exporting rules to Decision Center component of IBM Operational Decision Manager, persistence locale should match the persistence locale of Decision Center.

Course Roadmap

- Introduction
- Configuring Embedded Rules in ICM environment
- ➔ Rules Authoring in Case Builder
- Executing Rules as part of a workflow
- Moving Rules from embedded environment to IODM
- Impact to ICM 5.1.1 customers
- Troubleshooting
- Known Issues, Limitations, Best Practices
- Course Summary

Authoring rules in Case Builder : CaseType

- Rules can be added from the “Rules” tab in a CaseType



Authoring rules in Case Builder : Add Rule

- Rule Name – 64 characters, also used to associate with a rule step
- Rule Unique Identifier – 64 characters (unique within CaseType)
- Rule Description – Meaningful description of the rule

?

Add Rule

OK All

Rule Name ^	Type ^	Description ^
<div><div>*Name:</div><div>CourseFee</div></div>	<div><div>Description:</div><div>Calculate Course Fee based on Selected Course</div></div>	<div>OK</div> <div>Cancel</div>
<div><div>*Unique Identifier</div><div>CourseFee</div></div>	<div>Type:</div> <div><input checked="" type="radio"/> Text-based rule <input type="radio"/> Table-based rule</div>	

<div><div>?</div><div>Add Rule</div><div>OK All</div></div>		
Rule Name ^	Type ^	Description ^
StudentGrading	Table-based rule	Student Grading based on Marks scored
CourseFee	Text-based rule	Calculate Course Fee based on Selected Course

Authoring rules in Case Builder : Text-Based Rules

- Natural language based business rule authoring
- Based on Business Action Language (BAL)
- Intuitive editing / error checking
- Text-based Rules sections
 - Local variables (Definitions) - optional
 - Conditions (if) - optional
 - Actions (then /else)
- All Case Properties available for rule authoring
- Case System Properties available for rule authoring
- Custom Parameters to reference external data sources

Authoring rules in Case Builder : Text-based Rule - An Example

Design Rule - Text

? Rule1



if

all of the following conditions are true :

- the Str1 Single Value of **CaseType1** contains **"Hello"**
- the Str3 Multiple Values of **CaseType1** contain the Str1 Single Value of **CaseType1**

then

set the Str2 Single Value of **CaseType1** to **"All Is Well"** ;
set the Str4 Multiple Values of **CaseType1** to { **"Hello World"**, **"All Is Well"** } ;
print "Strings are valid";

else

print "Incorrect String";

Authoring rules in Case Builder : Text-based Rule – Errors and Warnings

Design Rule - Text

ValidateStocks

Save Close

definitions

```
set 'variable1' to 100 ;  
if the StockPriceFromES of StocksCaseType is "IBM"  
then  
  print "Validated for IBM";
```

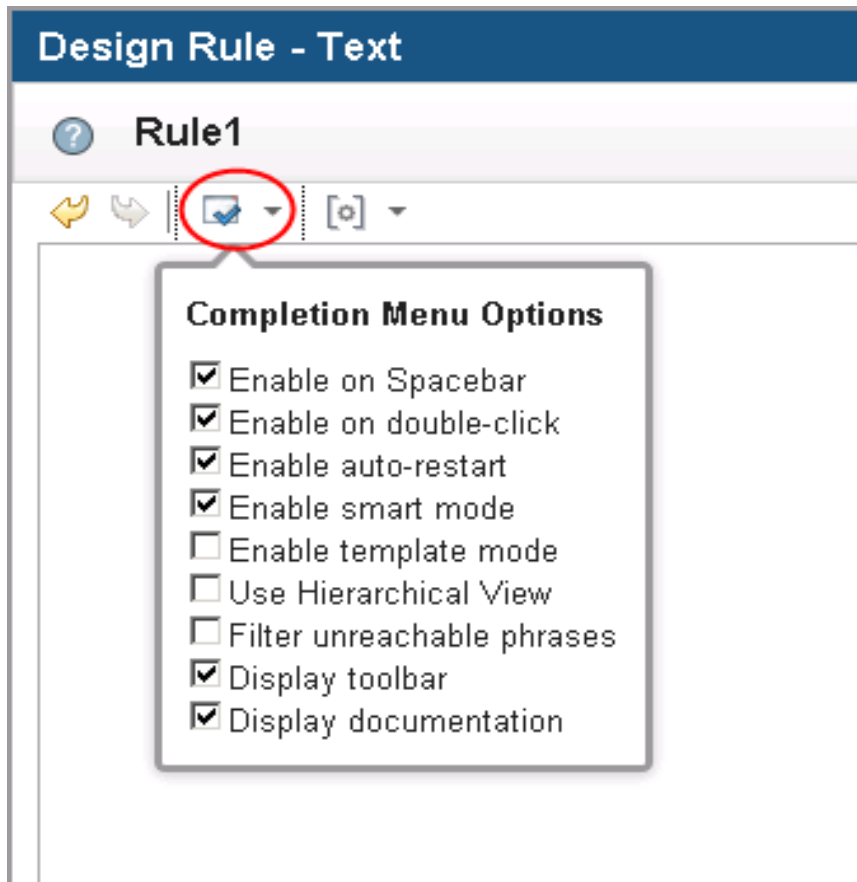
Warnings

Severity	Line	Message
Warning	2	Variable variable1 is not used.
Error	3	Invalid type 'string', it is not assignable from type 'number'

Errors

Authoring rules in Case Builder : Completion Menu options

- Assistance by providing all legal options while rule authoring



Authoring rules in Case Builder : Completion Menu options - Example

Design Rule - Text

Rule1



```
if
  all of the following conditions are true :
    - the Str1 Single Value of CaseType1 contains "Hello"
    - the S
then
  set the S
  set the S
  print "St
else
  print "In
```

CaseType1

Phrase:
the Case Identifier of <a caseType> : string

Hierarchical View

Documentation

Authoring rules in Case Builder : Choice List in Completion Menu

- Choice list values appears as options in completion menu

The screenshot displays the 'Design Rule - Text' window in IBM Case Builder. The rule is named 'Rule1' and contains the text: 'if the Str1 Single Value of **CaseType1** is <an object>'. A completion menu is open, showing a tree structure of values and phrases. The 'Values' section is expanded, showing 'Customer_Rating' with sub-items 'Bronze', 'Gold', and 'Silver'. The 'Phrases' section is also expanded, showing 'caseType' with sub-items 'the Str1 Single Value of <a caseType>' and 'the Str2 Single Value of <a caseType>', and 'object' with sub-items 'not <an object>', 'not null', 'not one of <objects>', 'null', and 'one of <objects>'. A red arrow points from the 'Choice List Name' label to the 'Customer_Rating' item. Another red arrow points from the 'Choice List Options' label to the 'Bronze' item. On the right side of the completion menu, the 'Domain value:' is listed as 'Bronze : Customer_Rating'.

Design Rule - Text

Rule1

if the Str1 Single Value of **CaseType1** is <an object>

Choice List Name

Choice List Options

Domain value:
Bronze : Customer_Rating

Authoring rules in Case Builder : Text-based rules – Use Case

- Get the **StocksPrice** from an external source, declared as a custom parameter in the rule.
- Declare a threshold as a definition, **StockThreshold**, in the rule.
- If the **StocksPrice** is more than the **StockThreshold**, then set the **StockRating** as Excellent, else as Satisfactory.
- Set the boolean case property **Validated**, to true if the **StocksPrice** is more than 0 (verifying that the external source has worked)
- For debugging purposes, we will print **StocksPrice** from external source

Authoring rules in Case Builder : Text-based rules – Use Case Contd.

- Solution Artifacts:
- Case Properties:
 - StockRating (String)
 - StockSymbol (String)
 - StockPriceFromES (Float)
 - Validated (Boolean)
- Choice List: RatingOptions
 - Excellent
 - Good
 - Satisfactory
- Role: Agent
- CaseType: StocksCaseType

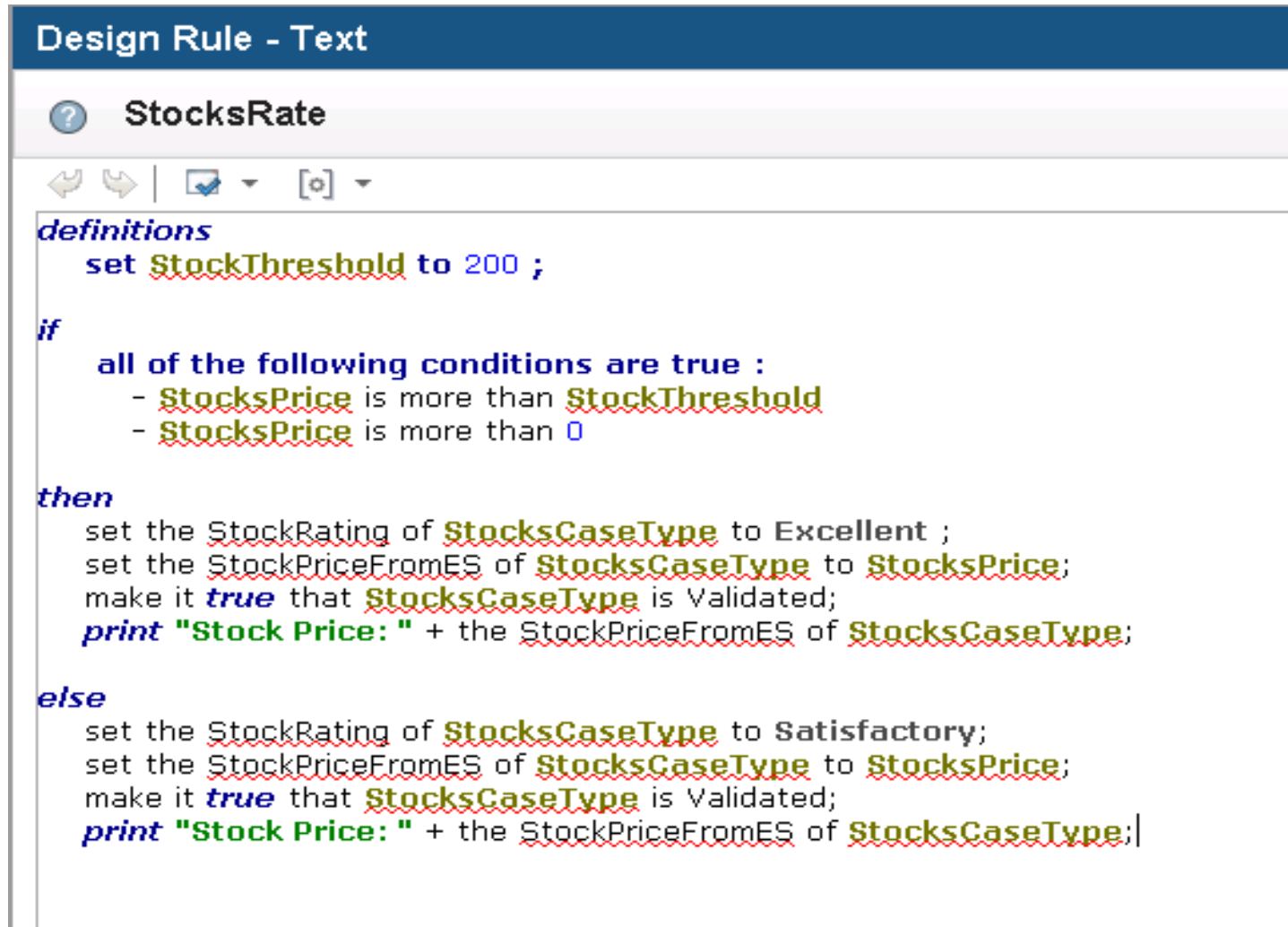
Authoring rules in Case Builder : Text-based rules – Custom Parameters

- Custom Parameters: Variables local to the rule, holding values from external sources, which can be referenced in the rule

The screenshot shows the 'Design Rule - Text' interface for a rule named 'StocksRate'. A modal dialog is open to define a custom parameter. The dialog contains the following fields and options:

- Add Parameter** button
- *Name:** StocksPrice
- Type:** Float (dropdown menu)
- *Unique identifier:** StocksPrice
- This parameter can have:**
 - ☒ A single value
 - ☐ Multiple values
- OK** and **Cancel** buttons
- Close** button (at the bottom of the main dialog)

Authoring rules in Case Builder : Text-based rules – Use Case – Rule Definition



The screenshot shows the 'Design Rule - Text' editor in IBM Case Builder. The rule is titled 'StocksRate'. The rule text is as follows:

```
definitions
  set StockThreshold to 200 ;

if
  all of the following conditions are true :
    - StocksPrice is more than StockThreshold
    - StocksPrice is more than 0

then
  set the StockRating of StocksCaseType to Excellent ;
  set the StockPriceFromES of StocksCaseType to StocksPrice;
  make it true that StocksCaseType is Validated;
  print "Stock Price: " + the StockPriceFromES of StocksCaseType;

else
  set the StockRating of StocksCaseType to Satisfactory;
  set the StockPriceFromES of StocksCaseType to StocksPrice;
  make it true that StocksCaseType is Validated;
  print "Stock Price: " + the StockPriceFromES of StocksCaseType;
```

The rule uses underlined text for variables and constants, and bold text for boolean values. The rule is structured with 'definitions', 'if' conditions, 'then' actions, and an 'else' branch.

Authoring rules in Case Builder : Table - based Rules

- Table based business rule authoring
- Used to express sets of similar conditions and actions in rules
- Intuitive editing / error checking
- Each row in a decision table rule can be considered as a text-based rule
- Multiple condition columns and multiple action columns
- Gap checks and overlap checks done by default
- Supports custom parameters

Authoring rules in Case Builder – Table-based rules Editor

Design Rule - Table

StocksRate by Company

Condition Columns: Company Symbol, Threshold

Action Columns: Rating, Validated

Refresh Button

Context Menu:

- Define Column...
- ☒ Check Gap
- ☒ Check Overlap
- Cut
- Copy
- Paste**
- Insert Column ▶
- Delete
- Clear

	Company Symbol	Threshold	Rating	Validated
1			-	-
2			-	-
4			-	-
5			-	-
6			-	-
7			-	-
8			-	-
9			-	-
10			-	-

Authoring rules in Case Builder : Defining Condition Column

Define Condition Column

Use this editor to write a condition for the column.

Completion Menu Options

☒ Enable on Spacebar

☒ Enable on double-click

☐ Enable auto-restart

☒ Enable smart mode

☒ Enable template mode

☐ Use Hierarchical View

☒ Filter unreachable phrases

☒ Display toolbar

☒ Display documentation

Severity	Line	Message
----------	------	---------

Define Cancel

Define Condition Column

Use this editor to write a condition for the column.

the StockSymbol of StocksCaseType contains <a string>

Placeholder for values to be filled in the rows




Severity	Line	Message
----------	------	---------

Define Cancel

Authoring rules in Case Builder – Defining Action Column

Define Action Column

Use this editor to write an action for the column.



set the StockRating of StocksCaseType to <a RatingOptions>

ChoiceList


Severity	Line	Message
----------	------	---------

DefineCancel

Authoring rules in Case Builder : Table-based rules

– Example Contd.

- Condition Columns:
 - **Company Symbol:** the StockSymbol of StocksCaseType contains <a string>
 - **Threshold:** StocksPrice is between <min> and <max>
- Action Columns:
 - **Rating:** set the StockRating of StocksCaseType to <a RatingOptions>
 - **Validated:** make it <a boolean> that StocksCaseType is Validated

Design Rule - Table				
StocksRate By Company				
[?] [x] ▼				
	Company Symbol	Threshold	Rating	Validated
1				

Authoring rules in Case Builder : Table-based rules

– Example – Populating Rows

The screenshot displays the Case Builder interface. On the left, a table with columns 'Company Symbol', 'Threshold', and 'Rating' is shown. Row 1 contains the text 'contains IBM' in the 'Company Symbol' column. A context menu is open over the table, with the 'Edit Custom Value...' option highlighted. A red arrow points from this option to the 'Edit Custom Value' dialog box on the right. A yellow callout box with the text 'Entering values into cells' and a red arrow points to the 'Company Symbol' column of row 2.

	Company Symbol	Threshold	Rating
1	contains IBM		
2			
3			
4			
5			
6			
7			
8			

Edit Custom Value

Use this editor to write a custom value for the cell.

the StockThresholdForDT of 'StocksCaseType' is between <sa number> and <sa number>

Severity Line Message

Update Cancel


Authoring rules in Case Builder :Table-based rules

– Example - Populating Rows

- Drop down menu for choice list values

Design Rule - Table				
StocksRate By Company				
[x] ▾				
	Company Symbol	Threshold	Rating	Validated
1	contains IBM	< 180	<div>Excellent</div> <div>Good</div> <div>Satisfactory</div>	
2				
3				
4				

- Drop down menu for Boolean values

Design Rule - Table				
StocksRate By Company				
[x] ▾				
	Company Symbol	Threshold	Rating	Validated
1	contains IBM	< 180	Satisfactory	<div>false</div> <div>true</div>
2				
3				

Authoring rules in Case Builder – Table-based rules

– Refresh Button

- Refresh button – removes unused rows, sorts rows in ascending order, validates for overlaps and gaps

The screenshot shows the Case Builder interface with a table of rules. A red arrow points to a 'Refresh Button' (a circular arrow icon) in the top left corner of the table. The table has five columns: 'Company Symbol', 'Threshold', 'Rating', and 'Validated'. The first four rows contain rules for 'contains IBM' with various thresholds. The fifth row is empty, and the 'Validated' column has a dropdown menu. Below the table, an error dialog box is displayed, showing the following text:

'StocksPrice' is between <min> and <max>

Errors:

- Lines 1 and 2 overlap
- Line 4: Value 'IBM' is invalid

The table below the dialog shows the same rules as the top table, but with the 'Validated' column set to 'true' for all rows.

	Company Symbol	Threshold	Rating	Validated
1	contains IBM	[180; 200]	Satisfactory	true
2	contains IBM	[200; 220]	Good	true
3	contains IBM	> 220	Excellent	true
4	contains IBM	> IBM	Excellent	true
5				

	Company Symbol	Threshold	Rating	Validated
1	contains IBM			true
2	contains IBM			true
3	contains IBM			true
4	contains IBM			true

Authoring rules in Case Builder : Table-based rules

– Otherwise and Disable

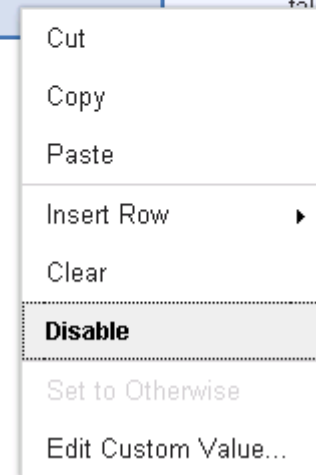
- Otherwise – Condition Column option – “catch all” for values not satisfying other rows, thereby resolving gaps in values

	Company Symbol	Threshold	Rating
1	contains IBM	[180; 200]	Satisfactory
2	contains IBM	[201; 220]	Good
3	contains IBM	> 220	Excellent
4	contains IBM		





- Disable – Action Column option – for disabling a particular action

Rating	Validated
Satisfactory	true
Good	true
Excellent	true
	false



Authoring rules in Case Builder : Table-based rules

– Example – Completed Rule

	Company Symbol	Threshold	Rating	Validated
1	contains AMNZ	[200; 250]	Satisfactory	true
2	contains AMNZ	[251; 300]	Good	true
3	contains AMNZ	> 300	Excellent	true
4	contains AMNZ	Otherwise		false
5	contains IBM	[180; 200]	Satisfactory	true
6	contains IBM	[201; 220]	Good	true
7	contains IBM	> 220	Excellent	true
8	contains IBM	Otherwise		false
9	Otherwise	Otherwise		false

The screenshot displays the IBM Case Manager interface. On the left is a 'Favorites' sidebar with a tree view of folders including 'All Searches', 'Checkout List', 'CodeModules', 'IBM Case Manager', 'Audit Configurations', 'Connection Definitions', 'Datasets', 'Page Templates', 'Rule Packages', 'Security Configurations', 'Solution Templates', and 'Solutions'. Under 'Solutions', the 'Stocks' folder is expanded, showing 'Pages', 'Rules', and 'NGSTO_StocksCaseType'. On the right is a table listing rules:

Name	Size	Modified By	Modified On	Major Version
StocksRate	1 KB	Intgpeadmin	8/19/13 9:02 PM	6
StocksRate By Company	4 KB	Intgpeadmin	8/20/13 11:16 PM	2

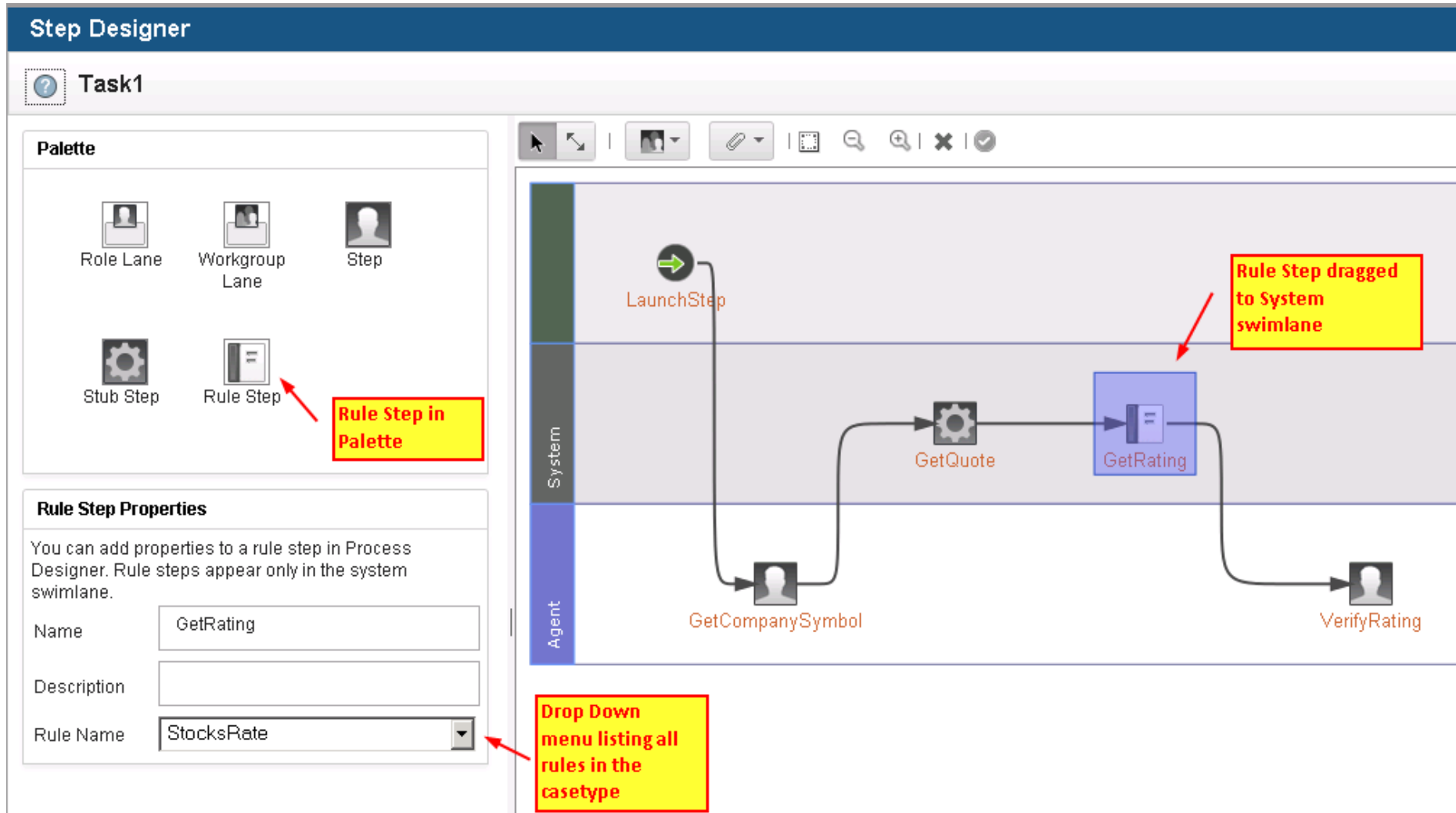
Annotations with red arrows point to the following elements:

- Text-based Rule stored in text format**: Points to the 'StocksRate' rule in the table.
- Table-based Rule stored in XML format**: Points to the 'StocksRate By Company' rule in the table.
- Rules Folder and CaseType Folder under the Solution folder**: Points to the 'Rules' folder under 'Stocks' and the 'NGSTO_StocksCaseType' folder.


Course Roadmap

- Introduction
- Configuring Embedded Rules in ICM environment
- Rules Authoring in Case Builder
- ➔ Executing Rules as part of a workflow
- Moving Rules from embedded environment to IODM
- Impact to ICM 5.1.1 customers
- Troubleshooting
- Known Issues, Limitations, Best Practices
- Course Summary

Executing Rules as part of a workflow : Step Editor



Executing Rules as part of a workflow :Mapping Custom Parameters to external data sources



- If Custom Parameters are used in the rules, the following have to be edited in Process Designer:
 - **ruleName** – The name of the rule to be executed
 - **customRuleParameterNames** – String Array with all custom parameters defined (added when rule step is added to the workflow)
 - **customRuleParameterValues** – String Array defining values to custom parameters in the order specified in customRuleParameterNames
 - **return_value** – ruleReturnValue, holds the value of the “print” statements in the rule

Executing Rules as part of a workflow :Mapping Custom Parameters to external data sources

Operation Parameters			
Name	Type		Expression
caseId	String		F_CaseFolder
ruleName	String		"StocksRate"
customRuleParameterNames	String[]		{"StocksPrice"}
customRuleParameterValues	String[]		{substr(DF_StockString, (strloc(DF_StockStri...
return_value	String		ruleReturnValue

DF_StockString defined as DataField for the workflow

Expression Builder (customRuleParameterValues: String Array)

Data Fields	Name
DF_StockString	DF_StockString
ruleReturnValue	
SolutionIdentifier	
	Type
	String
	Description

Expression defined for customParameterValues

Operators AND OR () Insert Clear

{substr(DF_StockString, (strloc(DF_StockString, "<Last>") + 6), ((strloc(DF_StockString, "</Last>") - (strloc(DF_StockString, "<Last>") + 6))))}

StocksPrice value extracted from External Sources Output

OK Cancel Help

Defining custom rule parameter feeds in Process Designer

- CustomRuleParameterValues is a string array and all values have to be converted to strings

DataType	DataField (example)	Converting to string
Boolean	DF_Bool1	convert(DF_Bool1, string)
DateTime	DF_DateTime1	timetostring (DF_DateTime1,"yyyy-mm-dd hh:tt:ss")
Float	DF_Float1	convert(DF_Float1, string)
Integer	DF_Int1	convert(DF_Int1,string)
String	DF_Str1	No need to convert as string is expected
Multi-value properties	DF_StrArray	arraytostring(DF_StrArray,"{","}","","")

CaseType Validation - Rules

The screenshot shows the IBM Case Manager interface. The top navigation bar includes 'Manage Solutions \ Nag JRules S... \ CaseType1', 'Show Locked Items', 'Back', 'Validate', and 'Save'. The left sidebar has 'Case Type', 'Properties', 'Views', 'Case Folders', 'Rules', and 'Tasks'. The main area displays a table with columns 'Rule Name', 'Type', and 'Description'. A red arrow points from the 'Validate' button in the top bar to the 'Validate' button in the table's toolbar. A modal dialog box is open in the foreground, titled 'Errors occurred when the case type was validated:'. It contains the following text: 'Rule1', 'There are validation errors in the rule. Open Rule Designer and fix the errors that are listed.', 'There are validation errors in the case type.', and 'Correct the errors that are described in the error message.' The dialog box also shows the error code 'FNRPB6111E' and a 'Close' button.

Rule Name	Type	Description
Rule1	Text-based rule	

Errors occurred when the case type was validated:

Rule1
There are validation errors in the rule. Open Rule Designer and fix the errors that are listed.

There are validation errors in the case type.

Correct the errors that are described in the error message.


FNRPB6111E

Close

Course Roadmap

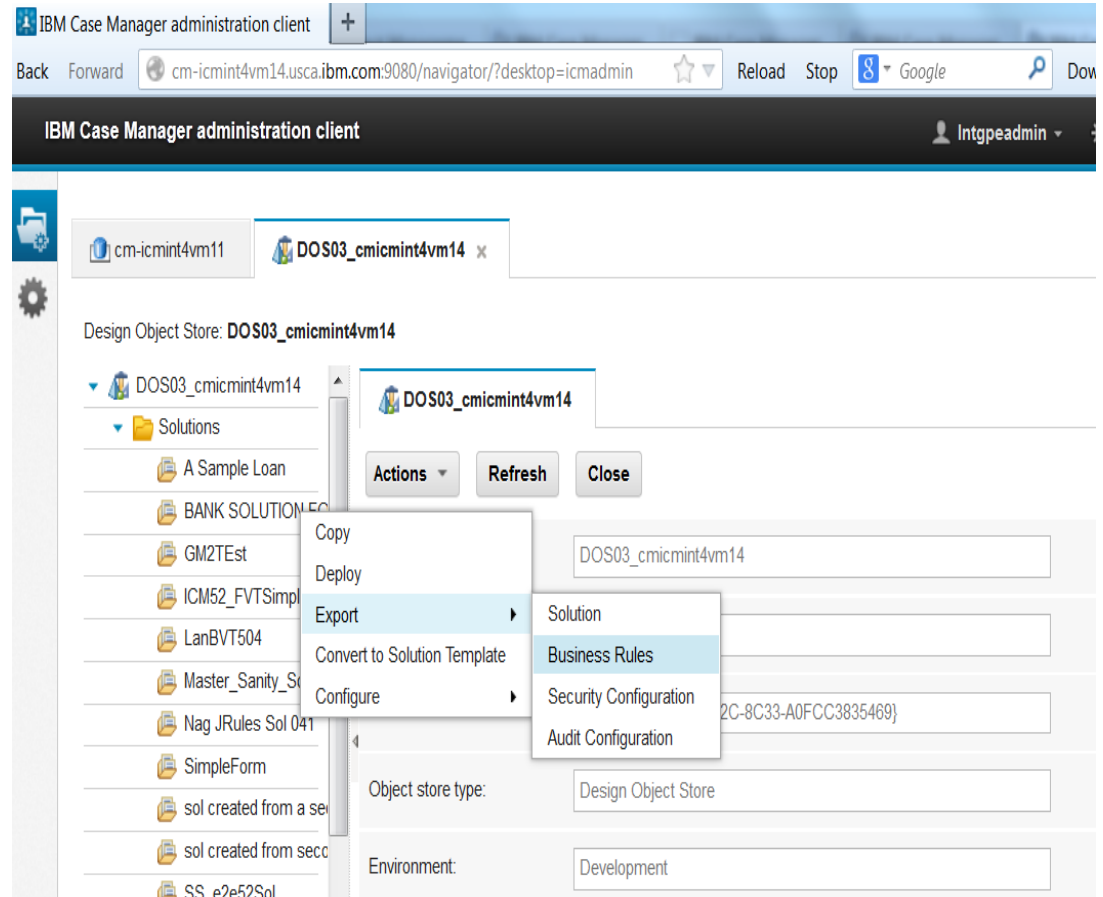
- Introduction
- Configuring Embedded Rules in ICM environment
- Rules Authoring in Case Builder
- Executing Rules as part of a workflow
- ➔ Moving Rules from embedded environment to IODM
- Impact to ICM 5.1.1 customers
- Troubleshooting
- Known Issues, Limitations, Best Practices
- Course Summary

Motivations for moving embedded rule to IODM

- 
- Centralized governance requirements
 - Change in organizational policies around Change management, version control requirements for rule artifacts
 - New requirements arise that need more advanced authoring features like Decision Trees, Rule Flows, Templates
 - Can be a stage by stage process, some rules remain in embedded and some moved to full IODM

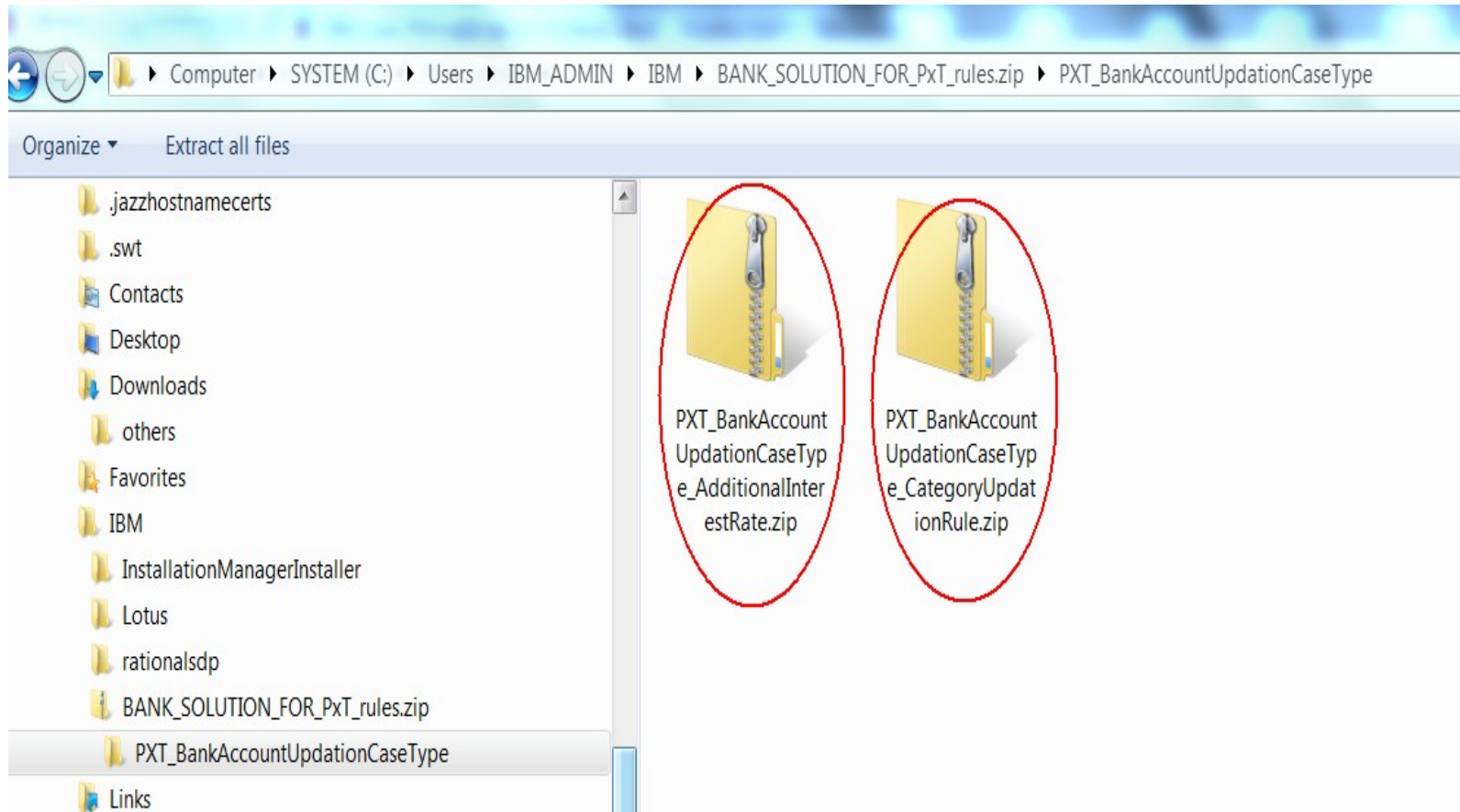
Exporting embedded rules as rule projects using ICM Administration Client

- All rules in a solution are exported as a zip file from ICM Administration Client
- Pre-requisite : Solution should have been successfully deployed before export is invoked



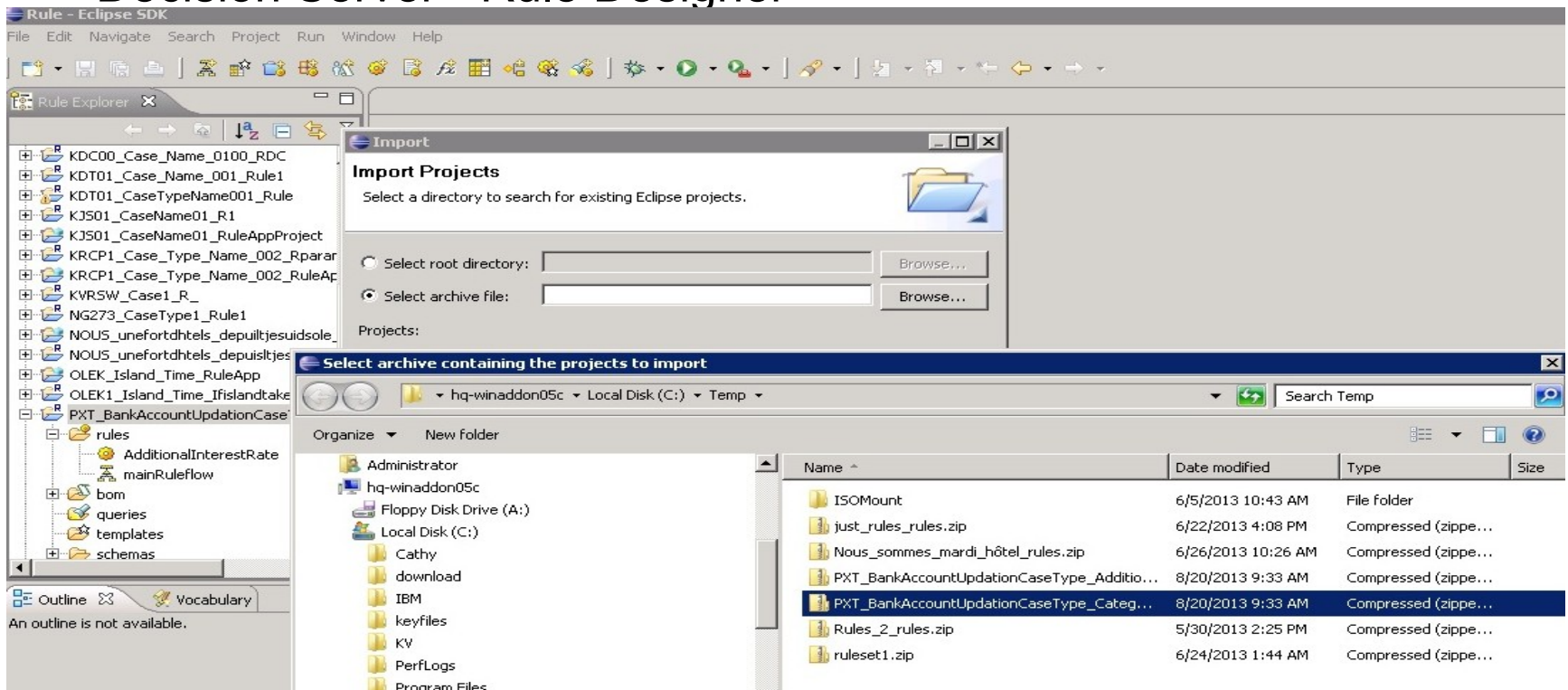
Structure of rule export output

- Output zip contains multiple rule project zip files, one per rule



Importing to IODM

- Individual rule project zip files can be imported to IODM products
 - Decision Center
 - Decision Server - Rule Designer



Changes required in solution for migrated rules

- Remove rule steps, bring in web services invocation methods
- You can choose to move only few of the rules to web services invocation
- Use outgoing type as “XML message” to pass info to IODM web service

The screenshot displays the IBM Business Process Manager (BPM) interface, illustrating the configuration for migrating rules to web services invocation.

Workflow (Main Map): The main map shows a workflow starting with a **LaunchStep** (green icon) leading to an **ILOG DS** step (gear icon). Below the main map, the **Workflow Properties** tab is active, showing the **Step Name** as **ILOG DS**.

Invoke Message: The **Invoke** dialog box is open, showing the configuration for the **ILOG DS** step. The **Message** tab is selected, showing the **Partner Link** as **ILOGLink** and the **Operation** as **PXT_BankAccountUpdationCaseType_CategoryUpdationRule**. The **Message Type** is set to **XML** (selected).

Outgoing Message: The **Outgoing Message** section shows the XML message structure for the **PXT_BankAccountUpdationCaseType_CategoryUpdationRuleRequest**. The message is an XML document with a root element **<caseObject>** containing various fields and their values, including **<DecisionID>**, **<CreditRatingFromExternalAgency>**, **<CmAcCaseState>**, **<DateCreated>**, **<CmAcCaseIdentifier>**, **<Creator>**, **<LastModifier>**, **<DateLastModified>**, **<PXT_customerCategory>**, **<PXT_relationshipStartDate>**, **<PXT_AccountBalance>**, **<PXT_AccountNumber>**, **<PXT_isSalaryAccount>**, and **<PXT_age>**.

Selected Functions: The **Selected Functions** list at the bottom right shows **Invoke** and **Assian** functions.

Sample outgoing XML message



```
"<PXT_BankAccountUpdationCaseType_CategoryUpdationRuleRequest
xmlns=""http://www.ibm.com/rules/decisionservice/PxTRuleApp/PXT_BankAccountUpdationCaseType_CategoryUpdationRule"">"+
  "<DecisionID>"+ ILOGDecisionID +"</DecisionID>"+
  "<CreditRatingFromExternalAgency>"+ convert(CreditRatingDataField,string) +"</CreditRatingFromExternalAgency>"+
  "<caseObject xmlns=""http://www.ibm.com/rules/decisionservice/PxTRuleApp/PXT_BankAccountUpdationCaseType_CategoryUpdationRule/param"">"+
    "<caseObject xmlns="">"+
      "<CmAcmCaseState>"+ "Working" +"</CmAcmCaseState>"+
      "<DateCreated>"+ timetostring(F_CaseFolder.DateCreated,"xs:datetime") +"</DateCreated>"+
      "<CmAcmCaseIdentifier>"+ F_CaseFolder.CmAcmCaseIdentifier +"</CmAcmCaseIdentifier>"+
      "<Creator>"+ F_CaseFolder.Creator +"</Creator>"+
      "<LastModifier>"+ F_CaseFolder.LastModifier +"</LastModifier>"+
      "<DateLastModified>"+ timetostring(F_CaseFolder.DateLastModified,"xs:datetime") +"</DateLastModified>"+
      "<PXT_customerCategory>"+ F_CaseFolder.PXT_customerCategory +"</PXT_customerCategory>"+
      "<PXT_relationshipStartDate>"+ timetostring(F_CaseFolder.PXT_relationshipStartDate,"xs:datetime") +"</PXT_relationshipStartDate>"+
      "<PXT_AccountBalance>"+ convert(F_CaseFolder.PXT_AccountBalance,string) +"</PXT_AccountBalance>"+
      "<PXT_AccountNumber>"+ F_CaseFolder.PXT_AccountNumber +"</PXT_AccountNumber>"+
      "<PXT_isSalaryAccount>"+ convert(F_CaseFolder.PXT_isSalaryAccount,string) +"</PXT_isSalaryAccount>"+
      "<PXT_age>"+ convert(F_CaseFolder.PXT_age,string) +"</PXT_age>"+
    "</caseObject>"+
  "</caseObject>"+
"</PXT_BankAccountUpdationCaseType_CategoryUpdationRuleRequest>"
```

Using XPath expressions in Assign function

The screenshot displays the IBM Business Process Manager interface. The main map shows a workflow with a 'LaunchStep' icon pointing to an 'ILOG DS' step. The 'Assign' function configuration window is open, showing the 'Assignment Parameters' table. The 'Expression' column contains the XPath expression: `substitute (ilog_output_xml, " xsi:type=""ns0:caseType"", "")`. The 'Expression Builder' window is also open, showing the 'Data Fields' list with 'CreditRatingDataField' selected. The 'Operators' section shows the expression: `xmlstringexpr(ilog_output_xml,"/","/*[local-name()='PXT_BankAccountUpdationCaseType_Cat
egoryUpdationRuleResponse']/*[local-name()='caseObject']/*[local-name()='caseObject']/*
[local-name()='PXT_customerCategory']")`. The 'Workflow Properties' panel at the bottom left shows the 'Step Name' as 'ILOG DS' and the 'Available Functions' as 'Assign', 'Call', and 'Checkpoint - BeginCheckpoint'.

Assignment Parameters

Name	Expression
ilog_output_xml	substitute (ilog_output_xml, " xsi:type=""ns0:caseType"", "")
F_CaseFolder.PXT_customerCategory	Editing...

Expression Builder

Data Fields

- CreditRatingDataField
- ILOGDecisionID
- SolutionIdentifier

Name: CreditRatingDataField

Type: Integer

Description

Operators: AND OR () Insert Clear

Expression: `xmlstringexpr(ilog_output_xml,"/","/*[local-name()='PXT_BankAccountUpdationCaseType_Cat
egoryUpdationRuleResponse']/*[local-name()='caseObject']/*[local-name()='caseObject']/*
[local-name()='PXT_customerCategory']")`

Workflow Properties

Step Name: ILOG DS

Available Functions: Assign, Call, Checkpoint - BeginCheckpoint

Course Roadmap

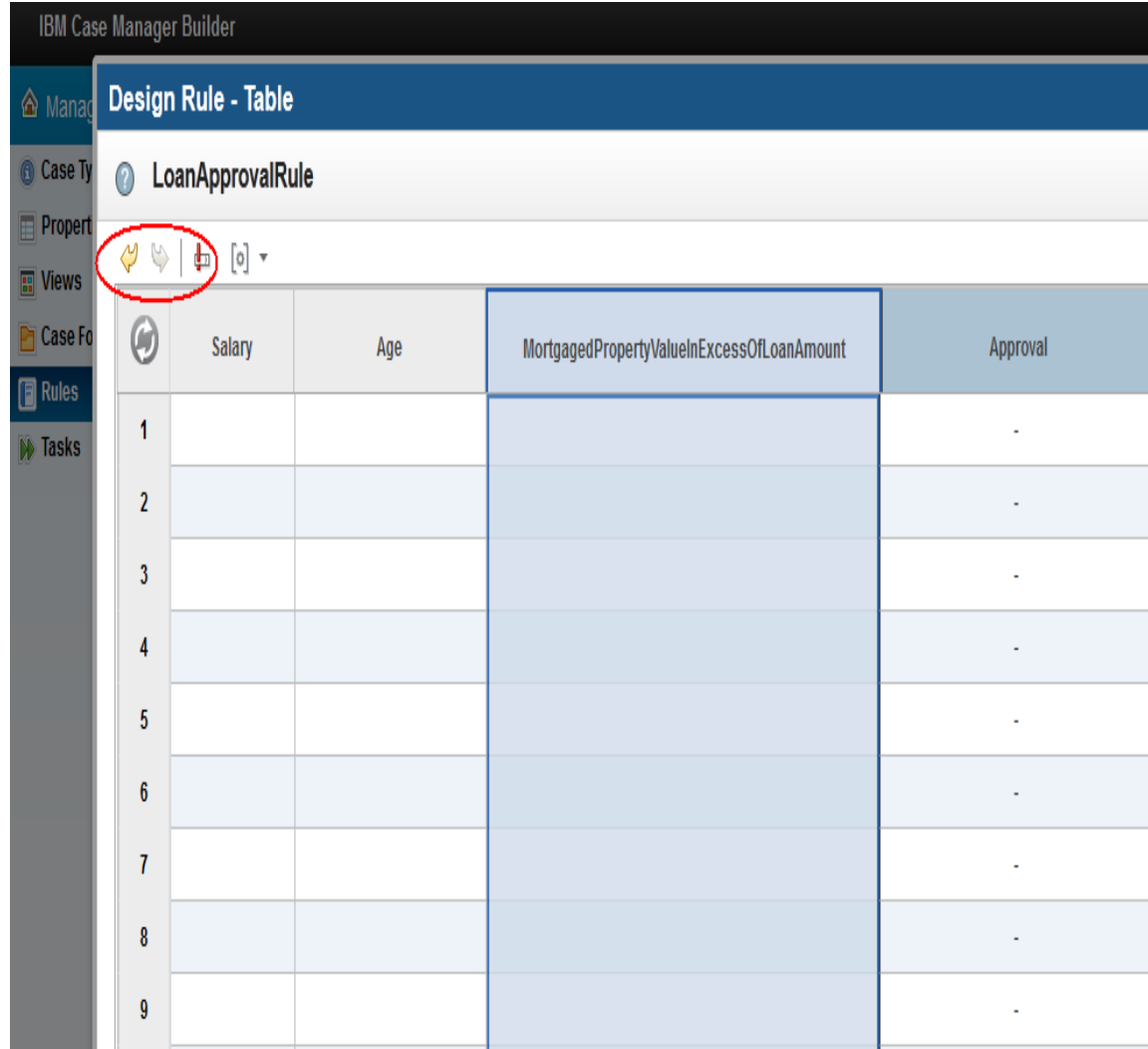
- Introduction
- Configuring Embedded Rules in ICM environment
- Rules Authoring in Case Builder
- Executing Rules as part of a workflow
- Moving Rules from embedded environment to IODM
- ➔ Impact to ICM 5.1.1 customers
- Troubleshooting
- Known Issues, Limitations, Best Practices
- Course Summary

Impact to ICM 5.1.1 customers

- ILOG 7.1 (or IODM) no longer bundled with ICM
- Web services invocation methodology still exists in CPE.
- Customer has purchased ILOG 7.x separately
 - Continue to use ILOG 7.x as is (or upgrade to latest IODM version), without any change to task processes
- Customer has been using bundled ILOG 7.x
 - Procure a license of IODM 8.5, continue to use web services invocation methodology
 - Move to Embedded Rules
 - Manual effort involved.
 - Make an assessment of impact due to functional differences, per client basis.

Embedded Rules Improvements – ICM 5.2.1

- Table-based rules have been improved
 - Undo
 - Redo
 - Preconditions



IBM Case Manager Builder

Design Rule - Table

LoanApprovalRule

Undo Redo [0]

	Salary	Age	MortgagedPropertyValueInExcessOfLoanAmount	Approval
1				-
2				-
3				-
4				-
5				-
6				-
7				-
8				-
9				-

Embedded Rules Improvements – ICM 5.2.1

- Preconditions enable you to test data before executing rules inside a decision table
- If the preconditions are not satisfied, the rules in the table are not executed
- Use preconditions to declare variables that can be used in the decision table rules

Course Roadmap

- Introduction
- Configuring Embedded Rules in ICM environment
- Rules Authoring in Case Builder
- Executing Rules as part of a workflow
- Moving Rules from embedded environment to IODM
- Impact to ICM 5.1.1 customers
- ➔ Troubleshooting
- Known issues, Limitations, Best Practices
- Course Summary

Troubleshooting - Log Locations

- Rule Deployment errors
 - Deployment log in CPE
 - p8_server_error.log
 - System.out on the AppServer profile where CPE is running
- Rule Runtime errors
 - pesvr_system.log
 - System.out on the AppServer profile where CPE is running
- Web Services invocation details
 - ws*.log file in the CPE logs location

Troubleshooting - Tracing

- Tracing can be enabled for the ICM_RuleOperations Java component
 - Trace entries are written to pesvr_trace.log
 - Enable TRACE_CM flag (component manager tracing flag)
 - Details about properties that are updated by rule execution
 - Details about properties that had a null value before execution
 - Output of all print statements in the rule
 - Results in tracing of rule execution APIs of IODM SDK as well.

Troubleshooting - Case properties not visible in Rule Editor



- **Exception in logs**

Exception created : java.lang.NoSuchMethodError: ilog/rules/xml/schema/IlrXsdSchemaReader.<init>(Z)V
at com.ibm.rules.sdk.builder.XmlSchema.computeSchemaInfo(XmlSchema.java:304)
at com.ibm.rules.sdk.builder.XmlSchema.getNamespaceResolver(XmlSchema.java:230)
at com.ibm.rules.sdk.builder.internal.ombuilders.XsdBuilder.createModels(XsdBuilder.java:97)

- **Reason : ILOG 71 and ICM 52 or CPE are co-existing in same WebSphere profile**
- **Suggestion : Not supported configuration, remove ILOG 71 from the profile**

Troubleshooting - Boolean case properties not visible



- Symptom : String/Integer/float/Date properties are visible, but boolean case properties are not
- Reason : The way IODM handles boolean properties during authoring is quite different.
- Suggestion : Review the following example
 - If the value of a Boolean property is true, set the value of another Boolean property to false:

if CaseTypeName is BooleanProperty1

then make it false that CaseTypeName is BooleanProperty2

Troubleshooting - Ambiguous Sentence in Rules

- Symptom : Rule Editors show 'ambiguous sentence' errors
- Reason : Parser is not able to figure out condition after **or** is a separate 3rd condition or it is part of earlier section.
- Suggestion : Add extra parenthesis or comma to resolve ambiguity.

if all of the following conditions are true :

- CreditRatingFromExternalAgency is more than 75
- the AccountBalance of BankAccountUpdateCaseType is more than 1000000
- or the Age of BankAccountUpdateCaseType is more than 40

then

set the CustomerCategory of BankAccountUpdateCaseType to GOLD ;

Severity	Line	Message
3	2	Ambiguous sentence.

if all of the following conditions are true :

- CreditRatingFromExternalAgency is more than 75
- the AccountBalance of BankAccountUpdateCaseType is more than 1000000
- or the Age of BankAccountUpdateCaseType is more than 40

then

set the CustomerCategory of BankAccountUpdateCaseType to GOLD ;

Severity	Line	Message
----------	------	---------

if all of the following conditions are true :

- CreditRatingFromExternalAgency is more than 75
- the AccountBalance of BankAccountUpdateCaseType is more than 1000000
- or the Age of BankAccountUpdateCaseType is more than 40

then

set the CustomerCategory of BankAccountUpdateCaseType to GOLD ;

Severity	Line	Message
----------	------	---------

Troubleshooting - Null Pointer exceptions during rule execution

■ Exception in logs

Caused by: ilog.rules.engine.IlrUserRuntimeException: null
at call to 'mainRuleflow flow task body'
at call to 'execute'
at ilog.rules.engine.sequential.IlrWorkingMemoryTupleIterator.iterate(IlrWorkingMemoryTupleIterator.java:124)
at
ilog.rules.engine.sequential.IlrSEQBaseExecTask.executeBodyOnWorkingMemory(IlrSEQBaseExecTask.java:161)
at ilog.rules.engine.sequential.IlrSEQBaseExecTask.executeBody(IlrSEQBaseExecTask.java:123)
at ilog.rules.engine.sequential.IlrSEQBaseExecTask.runBody(IlrSEQBaseExecTask.java:80)

- Reason : One or more case properties that are retrieved in the rule does not have a value set
- Suggestions
 - Make sure that all case properties retrieved in rule have a value set
 - Make use of defensive rules programming techniques to avoid Null Pointer Exception

Troubleshooting - Rules Pool Size exceeded

- Sample Exception in logs

[NG101_Task1:4F4F3DF1C7FA514B83C2260F889B892F:Workflow:executeRule] FAILED.; Exception:
ilog.rules.res.xu.pool.IlrPoolException: The pool is full.

at com.ibm.rules.res.xu.pool.internal.PoolImpl.waitNotFull(PoolImpl.java:179)

at com.ibm.rules.res.xu.spi.internal.XUConnectionManager.allocateConnection(XUConnectionManager.java:84)

at ilog.rules.res.xu.cci.IlrXUConnectionFactory.getConnection(IlrXUConnectionFactory.java:95)

at com.ibm.rules.res.xu.client.internal.XUClient.createRuleEngineSession(XUClient.java:111)

at ilog.rules.res.session.impl.IlrStatefulSessionBase.<init>(IlrStatefulSessionBase.java:105)

- Tune the rules max pool size (default is 100) by adding **-DrulesMaxPoolSize=<maxSize>** JVM argument to CPE AppServer profile and restart CPE AppServer.
- Tuning Guideline : Number of expected parallel rule executions in a CPE server

Course Roadmap

- Introduction
- Configuring Embedded Rules in ICM environment
- Rules Authoring in Case Builder
- Executing Rules as part of a workflow
- Moving Rules from embedded environment to IODM
- Impact to ICM 5.1.1 customers
- Troubleshooting
- ➔ Known Issues, Limitations, Best Practices
- Course Summary

Known Issues

- Limited support for multi valued case properties & multi valued custom parameters in rule authoring
- Limited Arabic & Hebrew rules support – Rule authoring(syntax) supported in Hebrew/Arabic, but recommended to do authoring from an English (or other supported rule persistence locale) environment.
- Traditional Chinese (zh_TW) is not supported – the option is present in the drop-down menu of “Configure Business Rules” task in CMAC, but the rule defaults to English
- There are known issues with embedded Decision Tables in non-English rule persistence locales.

Known Issues – using different locales for browser & rule persistence

- Mix of both locales in Rule editor
 - Syntax & errors in rule persistence locale
 - UI options in browser locale

ルール設計 - テキスト

CustomerCategoryUpdate

保存 閉じる

入力メニュー・オプション

- ☒ スペース・バーで有効化
- ☒ ダブルクリックで有効化
- ☒ 自動再始動の有効化
- ☒ スマート・モードの有効化
- ☐ テンプレート・モードの有効化
- ☐ 階層ビューの使用
- ☐ 到達不可能な語句のフィルター処理
- ☒ ツールバーの表示
- ☒ 注釈の表示

definit...
set ...day' to the AccountBalance of BankAccountUpdateCaseType ;
if cust...s more than 100 and
the F...ntUpdateCaseType is not null and
the A...tUpdateCaseType is not null and
the F...ntUpdateCaseType is after 7/31/2011 and
the A...tUpdateCaseType is not one of Defaulters
then p...s programming techniques"

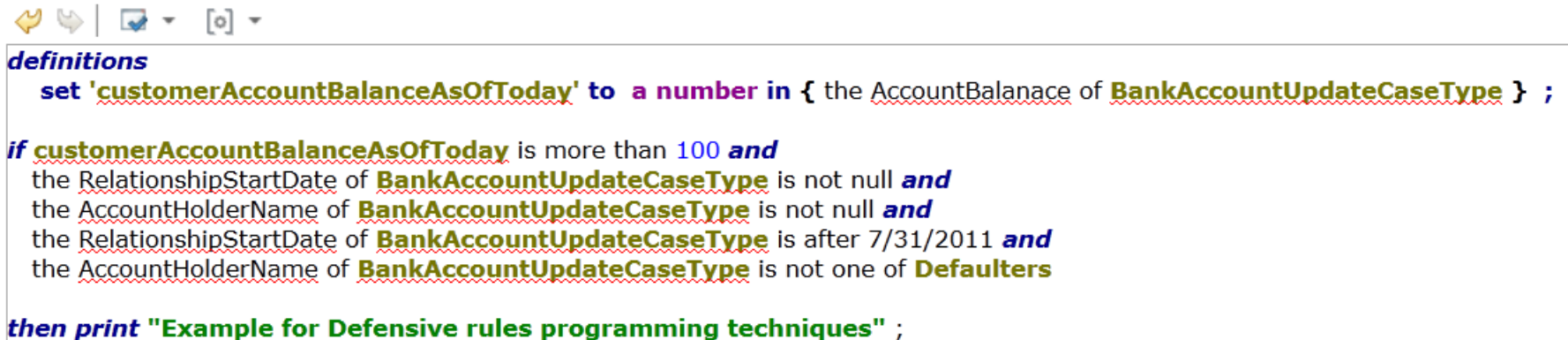
重大度	行	メッセージ
✖	10	The word ',' is missing.

Limitations

- Embedded Rules won't be able to handle certain characters in case property display names. Examples : <, >, " , (,) etc are not supported in case property display names for rules authoring
- Integer ChoiceLists are not supported in rule authoring
- There is no back-referencing in CaseBuilder from Rules to referred Case Properties or ChoiceLists
- Rule Editors are not fully accessible.

Best Practices

- Use completion menus to avoid errors/issues
- Make sure that case properties that are retrieved in a rule have values set before rule executes
- Use defensive rules programming techniques
 - Null value checks for Date/String case properties
 - Make use of restrictions in **Definitions** section to enforce implicit null value checks



The screenshot shows a rule editor interface with a toolbar at the top containing icons for undo, redo, save, and a search icon. Below the toolbar, the rule is defined in a structured format. The **definitions** section sets the **customerAccountBalanceAsOfToday** property to a number based on the **AccountBalance** of **BankAccountUpdateCaseType**. The **if** condition checks four criteria: **customerAccountBalanceAsOfToday** is more than 100, the **RelationshipStartDate** of **BankAccountUpdateCaseType** is not null, the **AccountHolderName** of **BankAccountUpdateCaseType** is not null, and the **RelationshipStartDate** of **BankAccountUpdateCaseType** is after 7/31/2011. The **then** clause prints the message "Example for Defensive rules programming techniques".

```
definitions
set 'customerAccountBalanceAsOfToday' to a number in { the AccountBalance of BankAccountUpdateCaseType } ;

if customerAccountBalanceAsOfToday is more than 100 and
the RelationshipStartDate of BankAccountUpdateCaseType is not null and
the AccountHolderName of BankAccountUpdateCaseType is not null and
the RelationshipStartDate of BankAccountUpdateCaseType is after 7/31/2011 and
the AccountHolderName of BankAccountUpdateCaseType is not one of Defaulters
then print "Example for Defensive rules programming techniques" ;
```

Course Roadmap

- Introduction
- Configuring Embedded Rules in ICM environment
- Rules Authoring in Case Builder
- Executing Rules as part of a workflow
- Moving Rules from embedded environment to IODM
- Impact to ICM 5.1.1 customers
- Troubleshooting & Logging
- Known Issues, Limitations, Best Practices
- ➔ Course Summary

Course Summary

You have completed this course and can:

- Describe the salient features of new Embedded Rules functionality
- Configure ICM to enable Embedded Rules features
- Author Text-based & Table-based business rules from Case Builder, use the rules in Tasks and verify rule execution results from Case Client
- Describe the steps involved in migrating Embedded Rules to full IODM product.
- Explain how ICM 5.1.1 customers may be impacted by this new feature
- Troubleshoot rules related issues in your ICM environment.
- Describe Known Issues, Limitations, Best Practices in rules functionality.

Product Help/Documentation/Resources



- Case Manager 5.2 Information Center
 - <http://pic.dhe.ibm.com/infocenter/casemgmt/v5r2m0/index.jsp>
- P8 5.2 Information Center
 - <http://pic.dhe.ibm.com/infocenter/p8docs/v5r2m0/index.jsp>
- IODM 8.5 Information Center
 - <http://pic.dhe.ibm.com/infocenter/dmanager/v8r5/index.jsp>

Questions

