BUSINESS RULES

Business Rules

- Business Rules "the language of business"
- Business logic, policies can be expressed as Business Rules
- Rules are stored in a repository and shared between activities, processes or even other applications.
- Business users can modify rules without reprogramming
  - Reduced costs and provides faster turnaround time
  - Increased visibility
  - Business experts have the control instead of technical experts
- Example

```
If
    credit rating is between 700-750
and
    age between 30-35
then
    APR is 5.6%
```

Business Rules Support

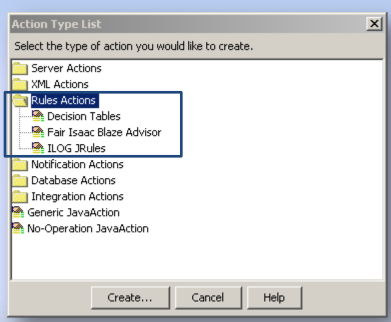
FUJITSU

- Interstage BPM supports integration with third party Business Rule Engines (BRMS)
- Out-of-the-box Actions to integrate with
  - ILog JRules
  - Fair Isaac Blaze Advisor
- Other third party engines can be integrated using Web Service or Java Interface.
- In-built support for defining and executing Business Rules

Business Rule Actions

FUJITSU

- Business Rule actions can be used to easily integrate with Rule Engine
- Sent Input from UDAs
- Execute rules at any point during process execution
- Store result in UDA for further processing.

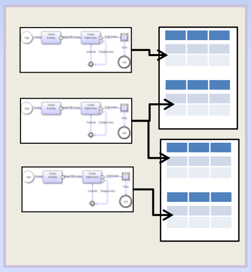

Action Type List

Select the type of action you would like to create.

Decision Tables

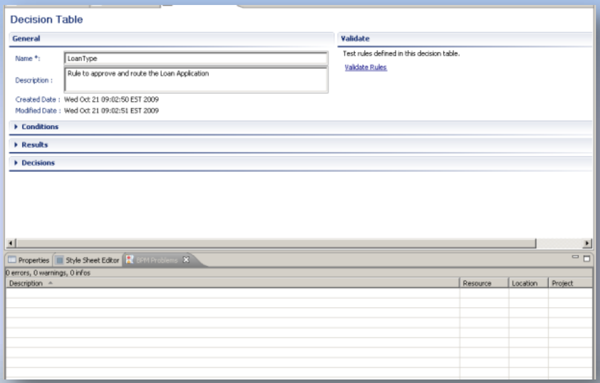
- Decision Tables allow users to create business rules in Studio.
- Create and package rules with process applications
- Rules can be shared among all process definitions within the application.
- Rules are stored in XML format.

- Organization
  - Application contains Process Definition and *"Rulesets"*
  - *Rulesets* contains *Decision Tables*
  - *Ruleset is a logical grouping of DTs*
  - Any process definition can access any DT



Creating a Decision Table

- Decision Table Editor in Studio provides easy GUI interface to define rules
  - create and manage Decision Tables; and
  - Validate business rules.



Decision Table

General

Name *: LoanType

Description : Rule to approve and route the Loan Application

Created Date : Wed Oct 21 09:02:50 EST 2009

Modified Date : Wed Oct 21 09:02:51 EST 2009

Validate

Test rules defined in this decision table.

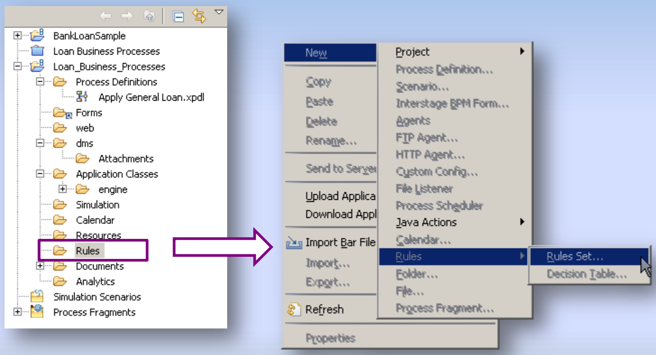
Validate Rules

0 errors, 0 warnings, 0 infos

Description	Resource	Location	Project

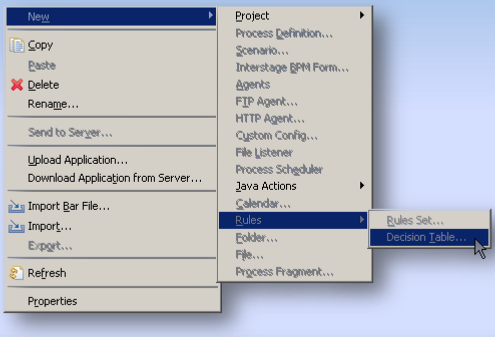
Create Rule Set

- One or more Rule Sets may be created in Application
- Rule Set stored in Rules Folder



Create Decision Table

- Create Decision Table – Example “DT 1”
- Select Rule Set to save Decision Table

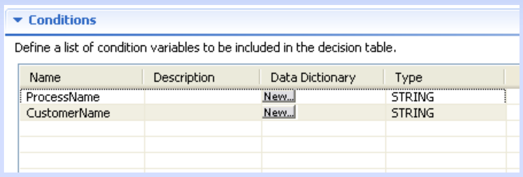


Decision Table Elements

- Decision Table Elements
  - Name
  - Description
  - Conditions
    - Conditions are “if” part of business rules e.g. *“if Customer 'age is between 25-35”*
    - Here *Customer’s Age* will be mapped to a UDA for input
  - Results
    - Are “then” part of business rules e.g. “*then APR is 5.6%*”
    - *APR* will be mapped to a UDA for output
  - Decisions
    - Each combination of condition and result form one decision, OR each rule in decision table is considered a decision.

Defining Conditions

- Name
  - Conditions can be defined by using arbitrary names of condition attribute e.g. “Customer Age”
  - A Rule may have multiple condition attributes
  - Each condition attribute is evaluated with “AND” operator in decision execution.
- Description
  - Used for documentation
- Type
  - Data type of attribute, this should match the UDA to map

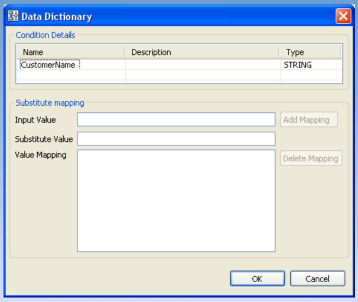

Conditions

Define a list of condition variables to be included in the decision table.

Name	Description	Data Dictionary	Type
ProcessName		New...	STRING
CustomerName		New...	STRING

Defining Conditions: Data Dictionary

- Data Dictionary provides an option to substitute an input value with something else before rule evaluation
- Substitution may be required in scenarios where data is coming from another system and values used are coded.



Name	Description	Type
CustomerName		STRING

Input Value:
Substitute Value:
Value Mapping:

- Data Dictionary Mapping
  - Input value: value of UDA e.g. *EXEC*
  - Substitute Value: mapping value to be used for rule evaluation e.g. "*Executive*"

Example:

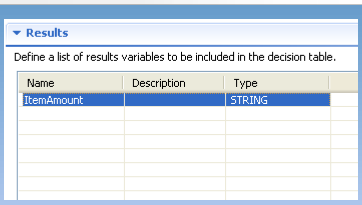
Customer Type

Executive – EXEC

President Club – PRSCLB

Defining Results

- Define result attribute names and data-types
- A rule may have multiple result attributes, each maps to a UDA
- After rule execution, result UDAs are updated.
- Attribute names can be different from UDA names, mapping is done separately.

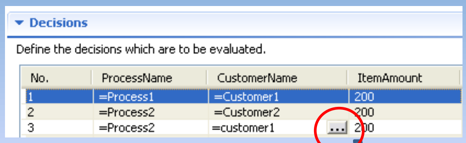


Define a list of results variables to be included in the decision table.

Name	Description	Type
ItemAmount		STRING

Decisions

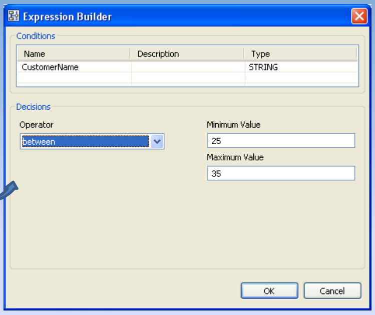
- Decisions section is created based on information provided in Condition and Result section.



Decisions

Define the decisions which are to be evaluated.

No.	ProcessName	CustomerName	ItemAmount
1	=Process1	=Customer1	200
2	=Process2	=Customer2	200
3	=Process2	=customer1	200



Expression Builder

Conditions

Name	Description	Type
CustomerName		STRING

Decisions

Operator: between

Minimum Value: 25

Maximum Value: 35

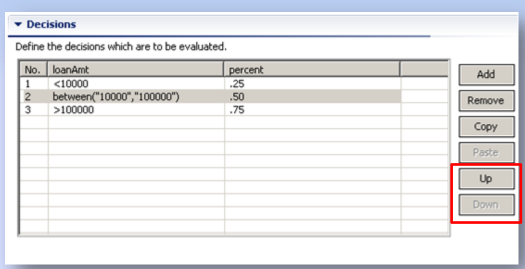
  - Type-in values to create rules
  - Use operators to create conditions

  - Operators

    =, !=, <, >, <=, >=, in, between, like, notlike

Rule Execution

- Rules in a DT are evaluated sequentially
- Once a rule matches input condition, execution stops and results are returned.
- DT does not validate rules for overlapping condition
  - e.g. >=100 and <=100 both conditions will be true for a value of 100



Decisions

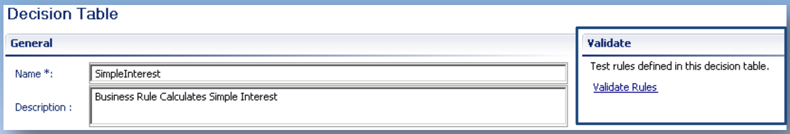
Define the decisions which are to be evaluated.

No.	loanAmt	percent
1	<10000	.25
2	between("10000","100000")	.50
3	>100000	.75

- Rule sequence can be changed by using "Up" and "Down" buttons

Testing Decision Table

- Test the rule by selecting “Validate Rules”



Decision Table

General

Name *: SimpleInterest

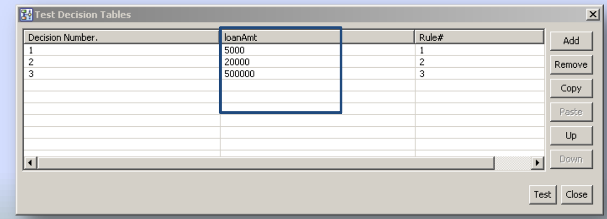
Description : Business Rule Calculates Simple Interest

Validate

Test rules defined in this decision table.

Validate Rules

- Enter different Condition Values

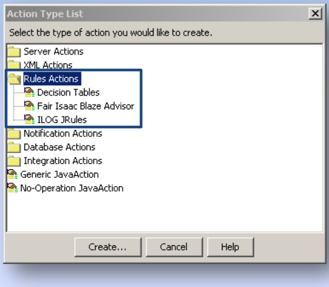


Test Decision Tables

Decision Number.	loanAmt	Rule#
1	5000	1
2	20000	2
3	500000	3

Decision Table Java Action

- Decision tables can be invoked in process definition by adding Decision Table Action at a node or at Process Definition level.
- iLog JRules (IBM) or Blaze Rules engine can also be connected to execute rules using available Actions



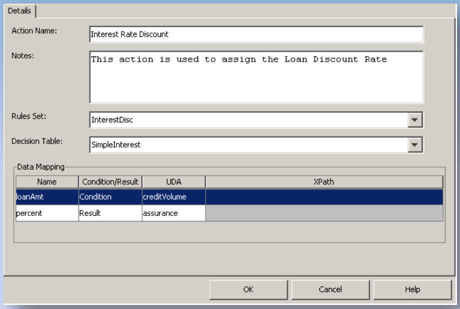
Action Type List

Select the type of action you would like to create.

- Server Actions
- XML Actions
- Rules Actions
  - Decision Tables
  - Fair Isaac Blaze Advisor
  - ILOG JRules
- Notification Actions
- Database Actions
- Integration Actions
- Generic JavaAction
- No-Operation JavaAction

Decision Table Java Action

- Action interface shows all available rules sets and tables within application.
- Configure:
  - Select Decision Table to execute
  - Map UDAs for Condition attribute
  - Map UDAs for result attribute
  - UDAs are filtered based on Data Type
  - XML UDA can be used to map to String condition with XPath


Details

Action Name: Interest Rate Discount

Notes: This action is used to assign the Loan Discount Rate

Rules Set: InterestDisc

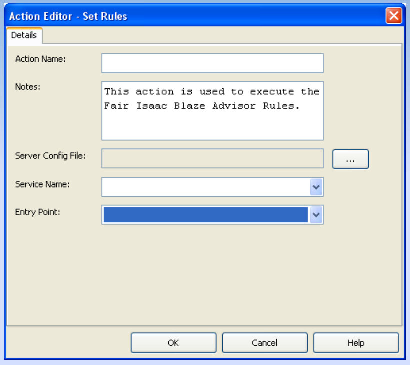
Decision Table: SimpleInterest

Data Mapping

Name	Condition/Result	UDA	XPath
loanAmt	Condition	creditVolume	
percent	Result	assurance	

Blaze Advisor Action

- Blaze Advisor uses “service” and “deployment manager” XML configuration files to connect to engine.
- In order to invoke Blaze Engine from BPM
  - Generate configuration files in Blaze
  - Copy “.server” and “.dmanager” files in “*dms/attachment*” folder in application
  - Browse and select server config files
  - Select service name and entry points


Action Editor - Set Rules

Details

Action Name:

Notes: This action is used to execute the Fair Isaac Blaze Advisor Rules.

Server Config File:

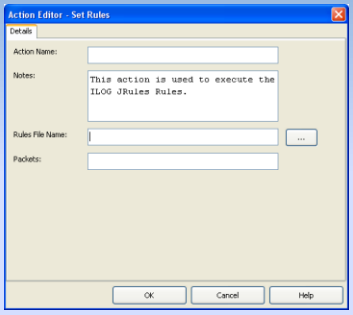
Service Name:

Entry Point:

  - *Note: follow admin guide for setting classpath and other configuration to invoke Blaze from BPM Engine.*

iLog JRules Action

- In order to invoke Blaze Engine from BPM
  - Generate rule file in JRules
  - Copy file in "dms/attachment" folder
  - Browse and select file

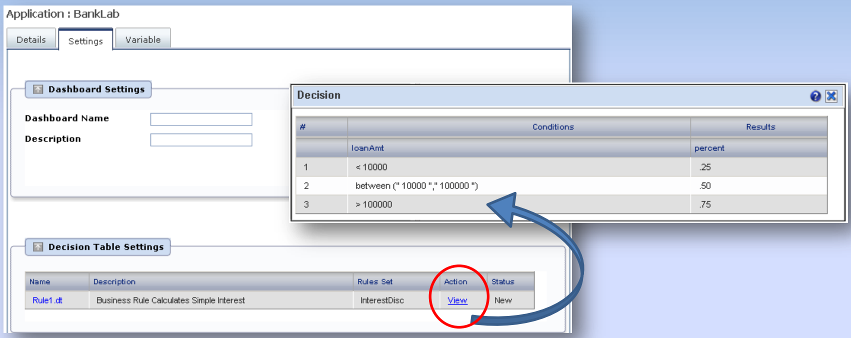


```
This action is used to execute the
ILOG JRules Rules.
```

  - *Note: follow admin guide for setting classpath and other configuration to invoke JRules from BPM Engine.*

Interstage BPM Console View

- View Decision Tables in Interstage BPM Console
  - Browse to application and select the "Settings" tab
  - Select Decision Table to View rules *(edit is not allowed in console view)*



Name	Description	Rules Set	Action	Status
Rule1.dt	Business Rule Calculates Simple Interest	InterestDisc	View	New

#	Conditions	Results
	loanAmt	percent
1	< 10000	.25
2	between (" 10000 "," 100000 ")	.50
3	> 100000	.75

shaping tomorrow with you