



IBM Software Group

WebSphere® Process Server V6.1 WebSphere Integration Developer V6.1

Business rules



@business on demand.

© 2008 IBM Corporation
Updated April 15, 2008

This presentation will cover business rules technology in WebSphere Process Server and WebSphere Integration Developer V6.1

Goals

- Provide an overview of business rules support in WebSphere Integration Developer V6.1 and WebSphere Process Server V6.1
 - ▶ Explain what a business rule is and how they are organized
 - ▶ Discuss why there are business rules and who uses them
 - ▶ See what WebSphere Integration Developer and WebSphere Process Server runtime offer for business rules
 - ▶ Explain the usage patterns and best practices for business rules in solutions deployed to WebSphere Process Server



A definition of a business rule is provided first, followed by an explanation of how business rules are organized. This will help to establish the foundation of how business rules work. A discussion of why business rules exist and which employee roles use them will follow. To show the different offerings between tools and runtime, the two are split apart to cover the details. Examples are also provided to help you understand how business rules can be used.

Agenda

- What a business rule is
- How business rules are organized
- Business rules and the roles that use them
- Business rules at runtime
- Business rule examples
- Summary and references



This section covers a definition of business rules.

What a business rule is

- A business rule is a representation of how business policies or practices apply to a business activity
- Think of a business rule as an if-then business decision
 - ▶ Example: **If** discount == 10,
Then total = total – (total * discount)
- A business rule's job is to...
 - ▶ evaluate the input data (if)
 - ▶ based on its state - compute new values, determine if the input is valid, or invoke a service (then)
- A business rule is used to help abstract the client from the implementation



Business rules can be thought of as an if-then business decision. For example, If the discount is 10%, the total equals the total minus the total times the discount rate of 10%. Therefore, a value of 55 with a 10% discount becomes 49.5 after the discount.

Business rules evaluate the business rule on an input using an if-then construct, then change a state or save or return some output. In the previous example, the variables "total" and "discount" were the inputs and outputs.

A business rule is used to help abstract the client from implementation. So a calling client, whether it is BPEL or a selector, does not care how a business rule performs its function, they send in the inputs and outputs and expect something to change.

Business rules in BPM

- **WebSphere Business Modeler**
 - ▶ Business analysts can define or capture business logic for processes in business rules where it can be easily change in production
- **WebSphere Integration Developer**
 - ▶ Integration specialists can add or extract integration logic into business rules where it can be easily changed in production
 - ▶ Integration specialists can modify or implement business rules defined in Modeler to add additional previously unknown details
- **WebSphere Process Server**
 - ▶ Runtime for applications with business rules and business policies defined with above products
 - ▶ Runtime management of business rules modifying business rules and business policies
- **WebSphere Business Service Fabric**
 - ▶ Application architects define business policies on business service that can be modified in production for composite business services applications



Business rules are supported across the Business Process Management product set. In WebSphere Business Modeler, the business analyst can model business rules as part of the business logic for the processes being modeled. In WebSphere Integration Developer, integration specialists can modify or implement the business rules, including those that were defined in WebSphere Business Modeler. In WebSphere Process Server the business rules and policies become part of the runtime. The business rules can also be managed using the business rules manager included in WebSphere Process Server. Finally, business rules are supported within WebSphere Business Service Fabric for use by application architects within a business service.

Section

How business rules are organized



This section covers the organization of business rules.

Business rule organization

- Business rules are assembled into business rule groups
- Business rules can be implemented in one of two ways...
 - ▶ Rule sets allow for multiple rule conditions and multiple actions to be performed
 - ▶ Decision tables allow for multiple rule conditions and a single set of actions to be performed



Business rules are assembled into business rule groups and can be implemented as rule sets or decision tables, as shown in the graphic on the right.

Business rule group

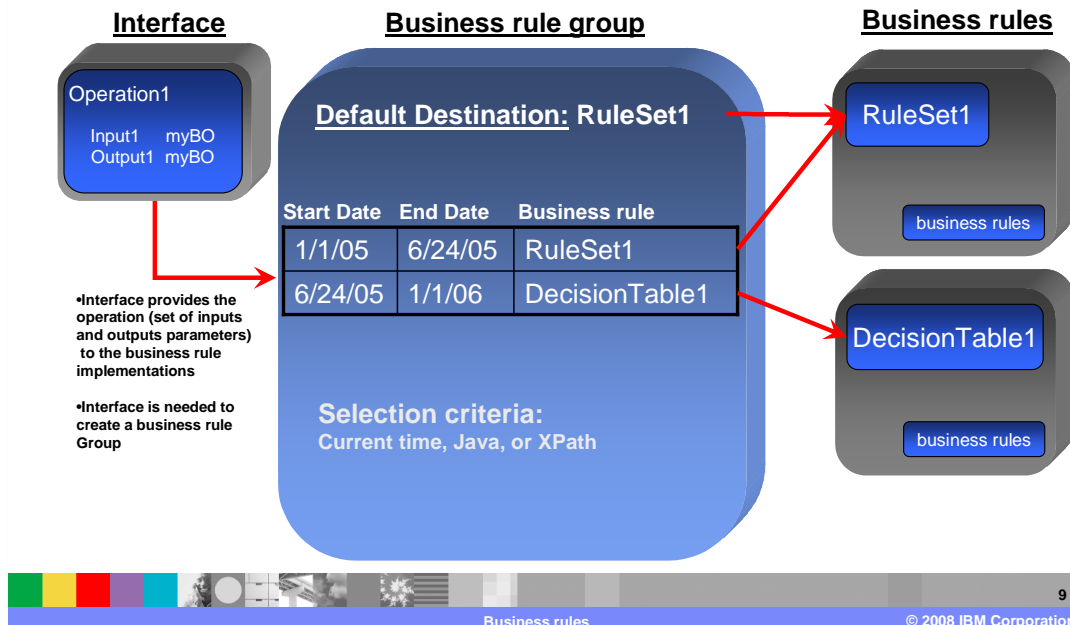
- Rule group is the highest level implementation component (umbrella)
 - ▶ Is exposed in the runtime as an SCA component
 - ▶ Defines the interface and operation the business rules use
- A rule group defines a date and time range to decide which business rule implementation is going to be used
 - ▶ A date and time is provided during invocation -- either “mined” from the input parameters using XPath, determined using a Java™ snippet, or the current date and time
 - ▶ The provided date and time are compared to the rule group’s date and time table to determine dynamically which business rule implementation is used
 - ▶ Example: If between 1/1/05 12:00am and 1/1/06 12:00am, use DecisionTable1. If any other date, use RuleSet1
- Rule implementations can be “scheduled” for a point in the past, present, or future
 - ▶ Able to invoke a process as it was in the past
 - ▶ Able to switch over to next version of a business rule on a future date



Business rule group is the highest level implementation component for a business rule. The main idea behind a rule group is to gather rule sets or decision tables that share a common business focus. However, it also serves as the gateway to the business rules since it is exposed in the runtime as an SCA service. All the services invoke a business rule on a request sent by a client, such as BPEL or selector through the rule group. A rule group also provides the interface for rule sets and decision tables to work off. Therefore, a rule group must be established before a rule set or decision table can work.

One of the most important functions the rule group does is define a date and time range for when a business rule is going to be used. For example, between 1/1/05 12:00am to 1/1/06 12:00am, use DecisionTable1. If any other date, use RuleSet1. Which business rule to use is determined by the date and time provided during invocation. There are three ways to get a date and time to determine which business rule to use. It can be either the current date and time, “mined” from the input parameters using XPath, or “mined” using a Java snippet. Mined means using code that finds a date and time parameter inside the input sent to the business rule or from an SCA service able to be accessed by the code. Rule implementations can be “scheduled” for a point in the past, present, or future. This provides the ability to invoke a process as it was in the past, or switch over to the next version of a business rule on a future date.

Business rule group



This graphic illustrates how a business rule group works. From left to right an Interface, which is an XML based WSDL file, provides the rule group with some inputs and outputs that are used in a business rule. Again, the rule group needs an interface in order to work.

Once the rule group is created, a default business rule must be selected. This gives the rule group the ability to always call on a business rule no matter what dates are supplied. Date and time selections are then added to specify which business rules are used for those certain dates and time.

The parameter method states how the date and time come in when being invoked. The three options again are to use the current system time or "mine" for the date and time using either XPath or a Java snippet.

Rule group

Operation →

Drop-down list

Start Date	End Date	Rule Logic
Mar 17, 2008 12:00 AM	Mar 17, 2009 12:00 AM	getRate
Mar 17, 2008 12:00 AM	Mar 17, 2009 12:00 AM	RateTable

Selection Criteria: Current date

Available Rule Logic:

- RateTable
- getRate

Business rules © 2008 IBM Corporation 10

This screen capture shows a rule group in WebSphere Integration Developer. In this example, the “getRate” interface is used. It has a default destination of rateTable and then two date and time selections. The parameter method being used is Java, which means a Java snippet is used to “mine” for a date and time used for the two selections above it.

The available rule logic lists all the rule sets or decision tables found that the rule group can use. The references are used to connect a business rule group with BPEL so the BPEL can call a business rule in the rule group.

Rule set

- Rule set is a set of one or more **if-then** conditions or action statements that are processed sequentially
 - ▶ Evaluated from first listed to last
 - ▶ Able to choose any number of Action or If-Then conditions
 - An Action does not have an If-Then condition, it initializes or sets variables
 - An If-Then condition to decide on a variable change
 - ▶ Provides the implementation for one operation in the rule group
- Evaluates multiple conditions and can fire multiple rules
 - ▶ needed when a rule is found true and fired, then more processing needs to be done



Business rules can be implemented in one of two ways. The first way is using rule sets.

A rule set is a set of one or more if-then conditions or action statements that are evaluated sequentially. Therefore, the first business rule listed is the first one evaluated when called upon at runtime. The second is evaluated second, and so on until the last one is evaluated. Rule sets hold any number of Action or If-Then conditions. To Explain further, an Action does not have an If-Then condition, it initializes or sets variables, but the If-Then condition does all the work to decide on a variable change. A rule set can evaluate multiple conditions and fire multiple rules. This might be needed when a rule is found true and fired, then more processing must be done to fire any other business rules in that rule set.

Decision table

- Decision table represents a multi-dimensional nested if-then structure
 - ▶ The decision table is broken down into a tree decision structure
 - ▶ A set of “if” conditions with “then” actions are defined at the intersection points of the table
- Conditions are evaluated in a nested order (tree)
- Evaluates one or more conditions, fires one rule
 - ▶ Most rules fall under a decision tree
 - Handles multiple rules efficiently (tree structure instead of sequential)
 - Most business rules are called to fire one rule



The second way business rules can be implemented is using Decision tables.

A Decision table represents a multi-dimensional nested if-then structure that can be thought of as a rule set capable of handling more complex decisions than a simple if-then. The decision table is broken down into a tree decision structure, with a set of “if” conditions with “then” actions defined at the intersection points of the table. Conditions are evaluated in a nested order or tree view. It evaluates one or more conditions, but fires only one rule. Most rules will fall under a decision tree since most business rules are called to fire one rule.

Business rule editors

Rule set and decision table editors provide an easy interface for capturing rule logic

The screenshot displays two editors side-by-side. The left editor is for a Decision Table, and the right editor is for a Rule Set.

Decision Table Editor (Left):

- Name:** RateTable
- Interface:** InternationalRateInterface
- Operation:** getRate
- Input:** input1
- Output:** output1
- Table:**

Conditions	"Canada"	"US"	"Asia_Pacific"	"Europe"	Actions
input1.destination					
output1.rate	0.05	0.05	0.15	0.20	

Rule Set Editor (Right):

- Name:** getRate
- Display Name:** getRate
- Interface:** InternationalRateInterface
- Operation:** getRate
- Input:** input1 (Rate)
- Output:** output1 (Rate)
- Variables:** (Empty table)
- Rules:**
 - Rule1:** Template: Template 1; Presentation: If billingRecord.usage_plan == Pay As You Go then increment = 30
 - Rule2:** Template: Template 1; Presentation: If billingRecord.usage_plan == Regular then increment = 6
- Templates:**
 - Template 1:** Presentation: If billingRecord.usage_plan == stringParam0; then increment = numberParam1;

Name	Type	Constraint	Description
stringParam0	string	None	
numberParam1	int	None	

Here you see the editor for decision tables on the left and the editor for rule sets on the right. For decision tables, you are able to specify multiple conditions to determine what action should be taken. In the decision table editor, the rule is presented in a tabular format that you can edit as needed. This example shows choosing between several telephone call destinations to determine what rate to charge.

In the rule set editor, a series of if-then statements determine what action is taken. A business rule takes the form of an interaction between one condition and one action. The rule is presented in a tabular format that you can edit as needed. There is also the ability to specify rule templates from within the rule set editor. Use a rule set template to define the implementation and parameters for an if-then or action rule. This template can then be used to create new instances of the same rule using different parameters.

When to use decision tables or rule sets

- Use decision tables when..
 - ▶ You want rules in a table format
 - ▶ When the same decision criteria are used in many rules
 - ▶ You want to fire only one rule
 - Can be multiple actions for the one rule
- Use rule sets when...
 - ▶ You want rules written in natural language
 - ▶ You have rules with a few clauses and variables in the conditional statements
 - ▶ You want to fire multiple rules



Here are some criteria to help you decide when to use a Rule Set and when to use a Decision Table.

Use decision tables when you want rules in a table format or when the same decision criteria are used in many rules. More importantly, use a decision table when you want to fire only one rule. It handles multiple rules efficiently, evaluates in a tree structure instead of sequentially, and allows you to enact multiple actions for the one rule that is fired (for example found true)

Use rule sets when you want rules written in natural language or you need to fire multiple rules. Rule sets are the easier of the two to understand since their user interface shows a straight representation of an if-then construct. Evaluating sequentially can be inefficient when there are many rules.

Section

Business rules and the roles that use them



This section covers the purpose of using business rules and discusses the employee roles that use business rules

Why there are business rules

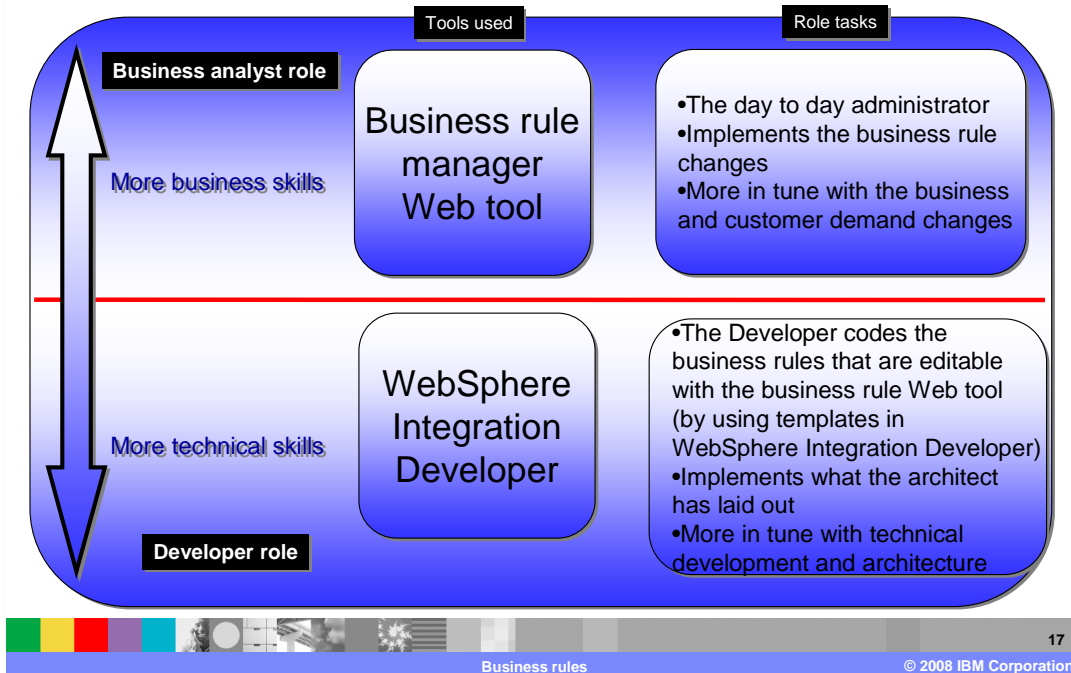
- Business rules are important for supporting Business process management (BPM)
 - ▶ Business rules help any size business proactively change processes for future events or speed response to changing customer demand
- Increasingly important technology for business flexibility and efficiency
 - ▶ Developer does not have to be bothered every time a simple rule needs to be changed
 - A business analyst can make that change swiftly
 - ▶ Multiple processes can call a business rule at the same time



Business rules are also important for supporting business process management because they help any size business proactively change processes for future events or speed response to changing customer demand. Big businesses with many processes can be slow-moving when responding to customer demand changes. With business rules, processes can be changed in minutes so the big business can be more agile. Small businesses need the ability to scale and grow fast, to add processes as needed, and change them quickly. Business rules are very useful in this case.

All sizes of businesses realize the importance of flexibility and efficiency. Business rules help keep business processes flexible by freeing up development from having to do tasks that a more business experienced person could do. Meaning, a developer does not have to be bothered every time a simple rule needs to be changed. A Business Analyst is a person that is more in tune to business needs and can make that change quickly. Business rules also provide the ability to have multiple processes call a business rule at the same time. This multi-threading allows multiple processes to run at the same time.

Roles



Developer and business analyst roles are displayed here in this graph to highlight the role tasks.

The developer and business analyst role are separated by the red line shown in the middle of the graph. More technical skills are required for the developer role. The developer uses WebSphere Integration Developer to create or manage the technical details of the business rules. The developer role works with the architect to implement the business process plan.

On the other side is the business analyst role. This person is more in tune with the changing business needs of the business. They use the business rule manager Web tool to change the business rules whenever necessary. The business analyst also works with the developer role to update and create new business rules for the company.

Section

Business rules at runtime



This section covers the business rules in tools and runtime.

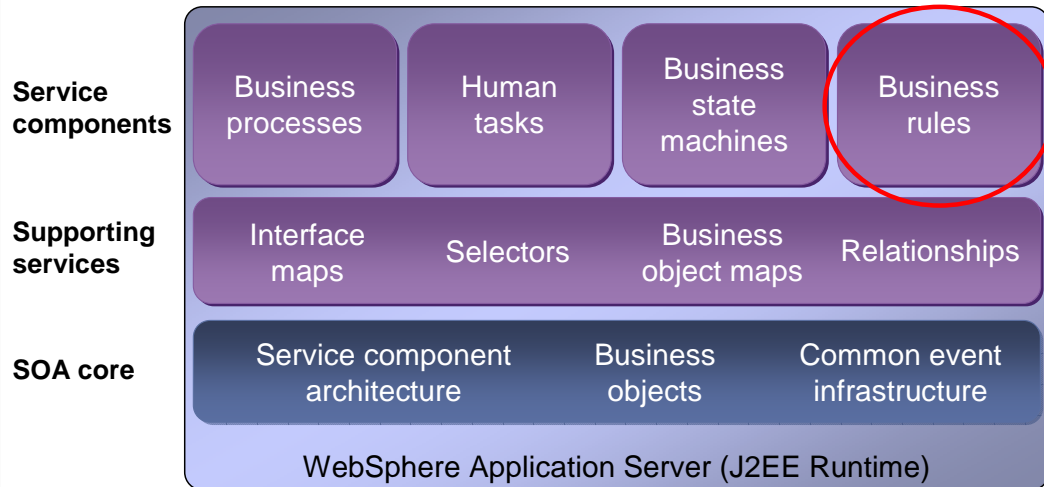
Runtime

- WebSphere Process Server runtime exposes business rules as an SCA component (using business rule groups)
- Handles typical integration rules
 - ▶ 80% of all business rules used today can be handled by the WebSphere Process Server
 - ▶ Complex rules, like inferencing, are left to vendor rule engines
- Standard J2EE™ deployment model (.ear file)
- Adds the ability to edit and update business rules in real-time using the business rules manager tool
 - ▶ It is included with WebSphere Process Server V6.1 as an optional installation



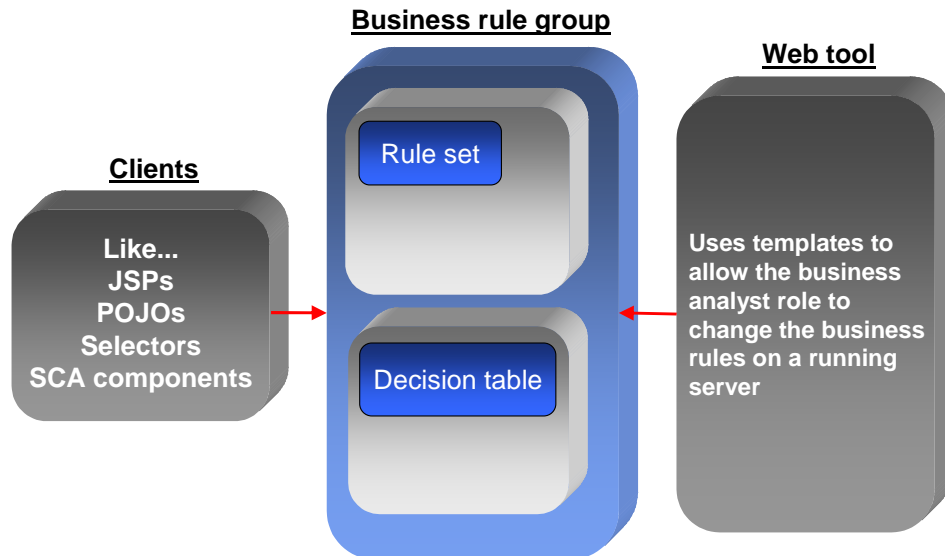
WebSphere Process Server V6 provides the runtime environment. Once a business rule is deployed to the WebSphere Process Server, the business rule group exposes the business rule as an SCA service. The WebSphere Process Server handles 80% of typical integration rules, but does not handle the other 20% of complex rules, like inferencing. The application server follows the standard J2EE deployment model, transforming .ear files for modules. The runtime also adds the ability to edit and update business rules in real-time using a straightforward Web tool that is included with WebSphere Process Server V6. This is discussed in more detail later in this presentation.

Where business rules fit



Business rules sit along side the stack of service components included in WebSphere Process Server. These SCA Components are exposed on the runtime so they can use each others' services.

At runtime



21

Business rules

© 2008 IBM Corporation

When a business rule is part of a running business process application, clients access a business rule group to find which implementation to use and to fire one to many business rules. These clients could be non-SCA components, called stand-alone references in WebSphere Integration Developer, such as JSP or other SCA components like Plain Old Java Object (POJO), selectors, BPEL, or other business rules. While the business process is running, there is a Web tool that comes with the server that provides the capability to edit business rules created as templates. Templates and the Web tool are discussed later in this presentation.

Templates

- Allow business analysts the power to change business rules (rule sets and decision tables) without having to restart the project or application server
 - Using the runtime's business rules manager Web tool
- Templates change XML to parameterized Java for a business rule conditions or actions
- Presentation section allows business analysts to read text instead of code from the business rule manager Web tool

A templated rule set in WebSphere Integration Developer...

Name	Rule5
Template	Template_Rule2
Presentation	If incoming clip size is : <input type="text" value="large"/> , set the clip size to the standard value: <input type="text" value="large"/>
Name	Rule6
Template	Template_Rule2
Presentation	If incoming clip size is : <input type="text" value="jumbo"/> , set the clip size to the standard value: <input type="text" value="jumbo"/>
Name	Rule7
Template	Template_Rule2
Presentation	If incoming clip size is : <input type="text" value="small"/> , set the clip size to the standard value: <input type="text" value="small"/>

...becomes a viewable, editable rule set in the business rule manager Web tool

If the incoming clip size is :	<input type="text" value="large"/>	, set the clip size to the standard value :	<input type="text" value="large"/>
If the incoming clip size is :	<input type="text" value="jumbo"/>	, set the clip size to the standard value :	<input type="text" value="jumbo"/>
If the incoming clip size is :	<input type="text" value="small"/>	, set the clip size to the standard value :	<input type="text" value="small"/>



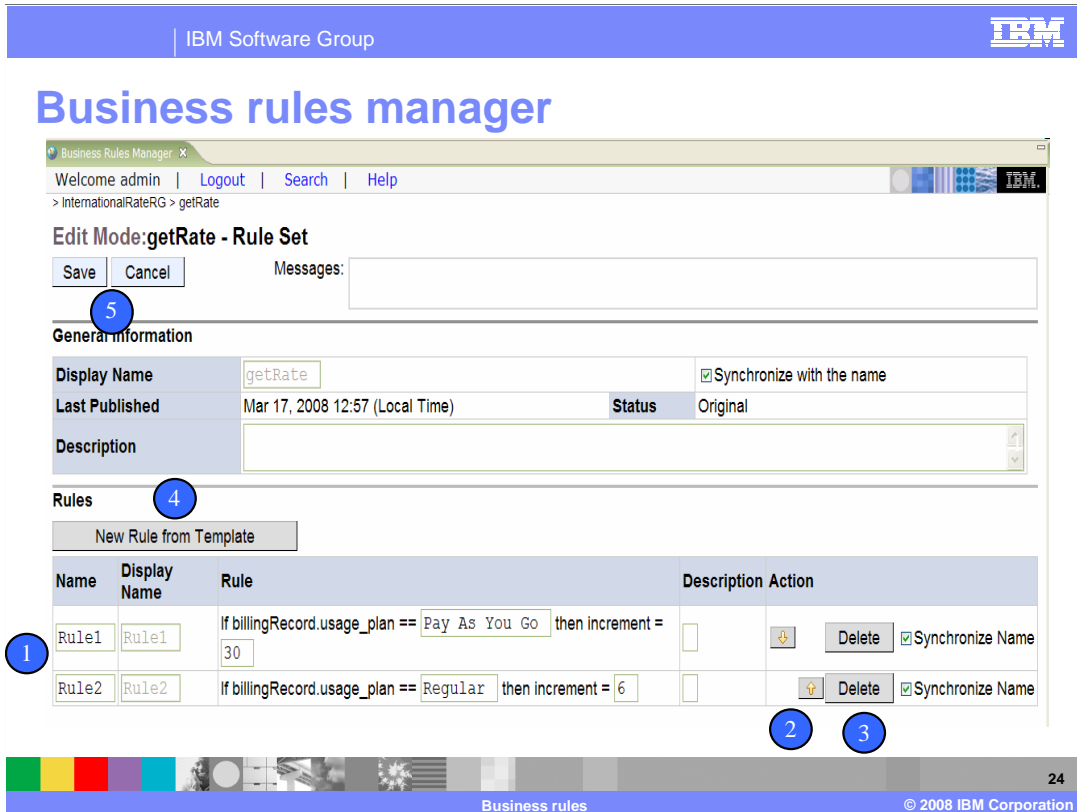
Templates are the key between WebSphere Integration Developer and the business rules manager Web tool. Templates allow Business Analysts to change business rules, including rule sets and decision tables, without having to restart the project or application server using the runtime business rule manager Web tool. Templates change the XML in the tools to parameterized Java on the runtime. The presentation section, seen in the screen capture to the right, allows Business Analysts to read text instead of code from the business rule manager Web tool.

Business rules manager

- Built specifically for the business analyst
- Manageable subset of full authoring functionality found in WebSphere Integration Developer
- Templates built by the developer role allow the business rule analyst role to edit business rules at runtime
 - ▶ Browser is all that is required to use
- It is part of the WebSphere Process Server runtime as an optional installation
 - ▶ using the administrative console, a JACL file (installBRManager.jacl) or through an administrative task (\$AdminTask_configBusinessRulesManager)
- The business rule manager Web tool is a JSP and Servlet based enterprise application that pulls and pushes business rules
- No need to restart application or application server when changing business rules
- Users are authenticated using standard WebSphere authentication and security



The business rule manager Web tool is built specifically for the business analyst. It has a manageable subset of full authoring functionality found in WebSphere Integration Developer. The only way a business analyst can see rules in the Web tool is if templates are built by the developer. This gives the business analyst a very easy way to edit business rules at runtime using a browser. The business rule manager Web tool is part of the WebSphere Process Server runtime as an optional installation by running the installBRManager.jacl script. Alternatively, the business rule manager can be installed using the administrative console or through an administrative task. The BRManager.war is in the installableApps directory of the WebSphere Process Server install path. The business rule manager Web tool is a JSP and Servlet based enterprise application that pulls and pushes business rules. There is no need to restart the application or the application server the business rules are running on when changing business rules. Users are authenticated using standard WebSphere authentication and security.



In the business rules manager Web tool, you can view all business rules exposed using the rule groups running on the application server. As an example, the screen capture shows what would be seen if a rule set was opened to be edited.

Starting with #1 on the numbered dots:

1. This is where the templated rule set or decision table is located. The boxed variables can be changed.
2. Here is where the order of business rules can be changed, this is only for rule sets since they are the business rule implementation that is evaluated sequentially from the top-down.
3. The delete button allows you to delete unwanted rules. The developer will need the updated business rules if any significant changes must be made to the application.
4. Use this button to create a new business rule from the templates in the current rule set or decision table.
5. Use these buttons to Save or Cancel the changes. After a Save is made, the newly changed business rules must be deployed back to the running application by way of a Deploy button found on a home page of the business rules manager Web tool

Public APIs for business rules management

- New in WebSphere Process Server V6.1
- Complete set of APIs for retrieving and modifying business rule groups, rule sets, and decision tables
- Custom business rule management clients can be built
- Allows for management of business processes, human tasks, and business rules through a single application



The business rules support for WebSphere Process Server V6.1 has been enhanced with a new public API for business rules management. The public business rules management APIs can be used for building custom management clients, which have many of the same capabilities as the Business Rules Manager. These APIs can be used along with other APIs to build complete management clients where business processes, human tasks, and business rules can all be managed through a single application.

New business rules management APIs

- APIs can be used to
 - ▶ Query business rule groups by name, namespace, custom properties
 - ▶ Modify business rule group operation selection table
 - ▶ Add and delete rules inside a ruleset
 - ▶ Add and delete conditions and actions in decision tables
- APIs must be used in an application installed on WebSphere Process Server
- Use with other APIs to build complete management clients
 - ▶ Manage business processes, human tasks, and business rules through a single application



Through the business rule management APIs, business rule groups can be queried based on the name, namespace, or custom properties. Modifications can then be made to the business rule group operation selection table to change which rules are active. Changes can be made to templated values within rulesets and decision tables. Inside a ruleset, rules can be added or deleted. In decision tables, conditions and actions can be added or deleted.

The business rules management APIs must be used in an application installed on WebSphere Process Server. The APIs do not provide a remote interface; however, they can be wrapped in a façade which is then exposed over a specific protocol for remote execution. These APIs can be used along with other APIs to build complete management clients where business processes, human tasks, and business rules can all be managed through a single application.

Section

Business rule examples



This section provides some examples of business rules.

Examples

- Rent-a-car corporate discounts
 - ▶ Able to change discount amounts during peak dates and times
- Insurance agency
 - ▶ Ability to invoke a process for an insurance claim that happened in the past
- Updating the company's Web site
 - ▶ Schedule specific dates that a Web site will update to the next development version (on planned future dates) without the end user knowing



Examples of business rules include:

A Rent-a-car company's ability to change corporate discount amounts during peak dates and times

An Insurance agency's capability to invoke a process for an insurance claim that happened in the past

Updating a company Web site is simplified when the company schedules business rules to change to updated versions of the Web site on planned future dates

Section

Summary



This section provides a summary.

Summary

- Found out what a business rule is and how they are organized
- Learned why there are business rules and who uses them
- Saw what tools and runtime offer for business rules
- Covered a few examples to better understand the functionality



In summary, this presentation covered business rule definition and organization, the purpose of business rules, business rules tools and runtime, and business rule examples.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WPSWIDv61_BusinessRules_Overview.ppt

This module is also available in PDF format at:

../WPSWIDv61_BusinessRules_Overview.pdf



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM WebSphere

J2EE, Java, JSP, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.