



IBM Software Group

# WebSphere® Process Server V6.0 WebSphere Integration Developer V6.0

## *Business rules*



@business on demand.

© 2007 IBM Corporation  
Updated April 5, 2007

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

## Goals

- Provide an overview of business rules with respect to WebSphere Integration Developer V6 and WebSphere Process Server V6
  - ▶ Find out what a business rule is and how they are organized
  - ▶ Learn why there are business rules and who uses them
  - ▶ See what WebSphere Integration Developer and WebSphere Process Server runtime offer for business rules
  - ▶ Cover a few examples to better understand the functionality better

A definition of business rule will be 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 will be split apart to cover the details. Examples will also be provided to help you understand how business rules can be used.

## Agenda

- **What is a business rule**
- How are business rules organized
- Why there are business rules and the roles who use them
- Business rules in tools and runtime
- Business rule examples
- Summary and references

This section will provide a definition of business rules.

## What is a business rule?

- 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 implementation
- Business rules are assembled into business rule groups
- Business rules can be implemented in one of two ways...
  - ▶ Rule sets
  - ▶ Decision tables

### Business rule group



Business rules can be thought of as an if-then business decision. For example, If the discount is 10%, the total would equal the total minus the total times the discount rate of 10%. So a total of 55 with a 10% discount would be 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.

Organization will be covered next, but you should know that 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.

## Agenda

- What is a business rule
- **How are business rules organized**
- Why there are business rules and the roles who use them
- Business rules in tools and runtime
- Business rule examples
- Summary and references

This section will cover the organization of business rules.

## 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 will use
  - ▶ Each operation in the rule group is implemented only by two ways
    - Rule set
    - Decision table
- 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 and fit into the rule group's date and time table to determine dynamically which business rule implementation will be used
  - ▶ Example: If between 1/1/05 12:00am and 1/1/06 12:00am, use DecisionTable1. If any other date, use RuleSet1)
  - ▶ The date and time can be either the current date and time, "mined" from the input parameters using Xpath, or determined using a Java™ snippet
  - ▶ 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

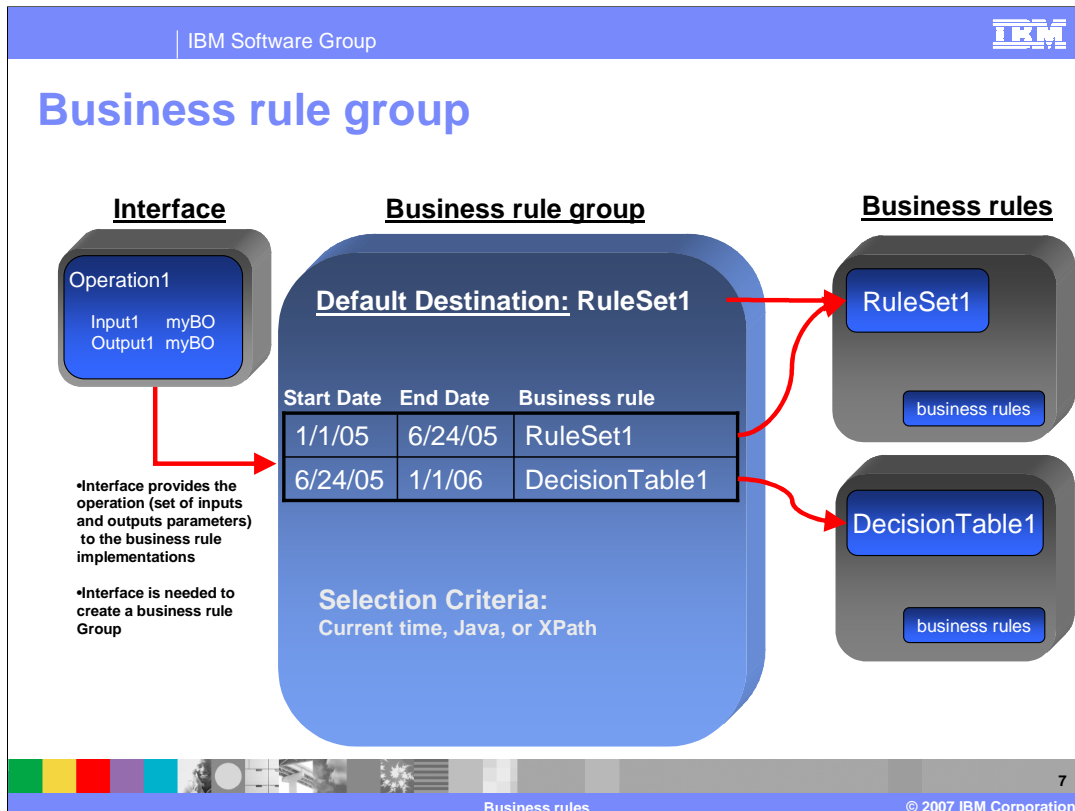
6

Business rules

© 2007 IBM Corporation

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.

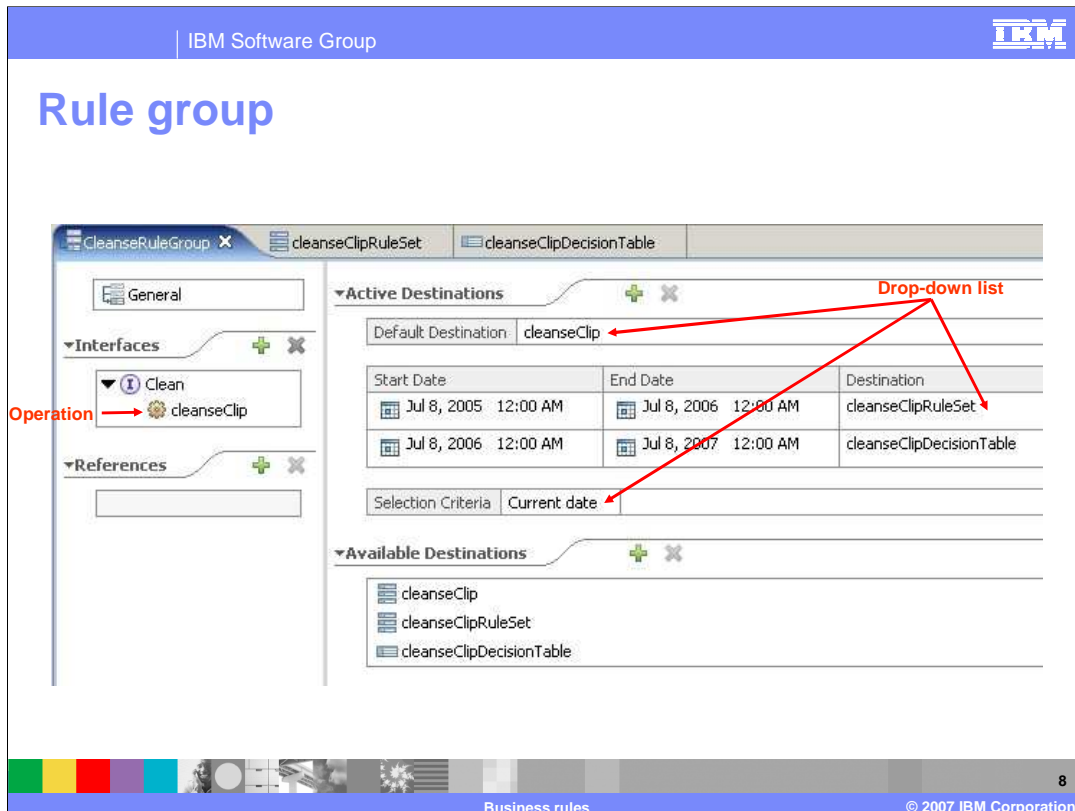
One of the most important functions the rule group does is define a date and time range for when a business rule is going to be used. For example, between 1/1/05 12:00am to 1/1/06 12:00am, use DecisionTable1. If any other date, use RuleSet1. Which business rule to use is determined by the date and time 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.



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 will be 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 will then be added to specify which business rules will be used for those certain dates and time.

The parameter method states how the date and time will 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.



This screen capture shows a rule group in WebSphere Integration Developer. In this example, the “clean” interface for cleanseClip is used. It has a default destination of cleanseClipDecisionTable and then two date and time selections splitting up 2005. The parameter method being used is Java, which means a Java snippet will be used to “mine” for a date and time used for the two selections above it.

The possible target destinations lists all the rule sets or decision tables found that the rule group could 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 1 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 will be the first one evaluated when called upon at runtime. The second will be 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.

## Rule set

The screenshot displays the 'Business rule group' containing a 'Rule set'. The 'Rule set' is expanded to show its configuration:

- Rule Set:** Detailed properties for this rule set.
- Interface:**

Input(s)	Name	Type
	inClipBG	ClipBG
Output(s)	Name	Type
	outClipBG	ClipBG
- Variables:**

Name	Type
temporaryVar	string
- Rules:**

Name	Presentation	Action
Action1		temporaryVar = "NULL"
Rule1	IF	inClipBG.Clip.color == "red"
	Then	outClipBG.Clip.color = "red+#"
- Templates:** (Empty)

A red arrow points to the 'Rule1' entry in the 'Rules' section, which is labeled as a 'business rule'. A context menu is visible over the 'Rule1' entry, showing options like 'Undo Delete Rule', 'Redo', 'Delete Item', 'Show in Properties', 'Add Action Rule', 'Add IF-Then Rule', 'Add Template Rule', 'Debug', and 'Save'.

The visual representation of a rule set is shown in this WebSphere Integration Developer screen capture. A rule set is part of a business rule group and operates off the interface inputs and outputs provided by the business rule group. Moving sequentially down the picture, there is a Variables section where you can create temp variables. Next is the business rules section of the rule set. This is where the business rules will be listed or created. The rule set offers two types of business rules to create, an Action and an If-Then rule. An action is a “do” statement, which performs an action without an if-else condition. The If-Then Rule takes a condition in order to make a decision on the output.

Notice that a template was created from this business rule. Templates are a very important part of business rules and will be discussed in the runtime section.

## 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 1 or more conditions, fires only one rule
  - ▶ Most rules will 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 1 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.

## Decision table

The screenshot displays the IBM Business Rules Editor interface. A blue arrow points from the 'Business rule group' label to the 'Decision table' configuration window. The window shows the following details:

- Decision Table:** Detailed properties for this decision table, including a button labeled 'cleanseClipDecisionTable'.
- Interface:** A table defining inputs and outputs.
 

	Name	Type
Input(s)	inClipBG	ClipBG
Output(s)	outClipBG	ClipBG
- Table:** A table defining conditions and actions.
 

Conditions		
inClipBG.Clip.color	"red"	"blue"
inClipBG.Clip.color	"red#5"	"blue#2"
Actions		

The bottom of the screenshot shows a status bar with 'Business rules' and '© 2007 IBM Corporation'.

Decision tables also derive the interface inputs and outputs from the business rule group. However, the decision table is much different from rule sets. In this example, the top row of the decision table represents the input (input1) and the output is represented on the second row (output1). It reads like "If inClipBG.Clip.color equals "red", then outClipBP.Clip.color gets "red#5"" and so on. You can add new Conditions, Condition Cases, and Actions, and convert the decision table to a template using the editor.

## 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 is 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 is 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 sequential, and allows you to enact multiple actions for the 1 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.

## Agenda

- What is a business rule
- How are business rules organized
- **Why there are business rules and the roles who use them**
- Business rules in tools and runtime
- Business rule examples
- Summary and references

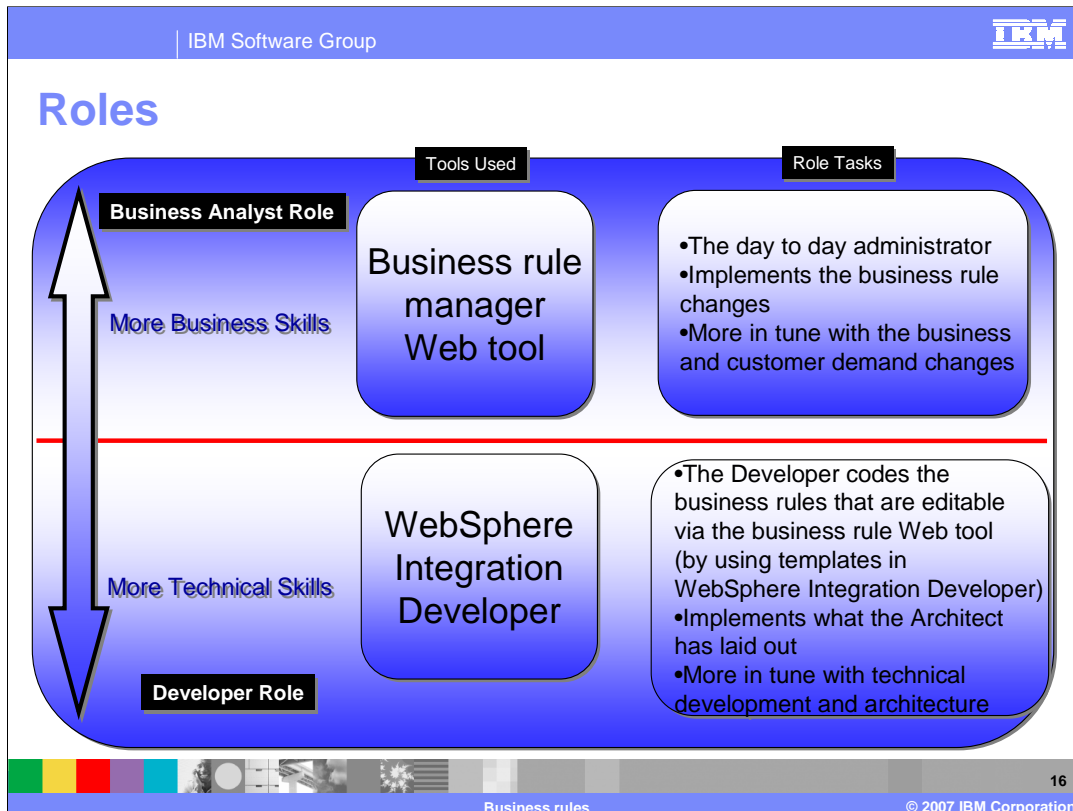
This section will cover the purpose of using business rules and discuss the employee roles that use business rules

## 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 (BPM) because they help any size business proactively change processes for future events or speed response to changing customer demand. Big businesses with a great deal of 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 fast. Business rules are very useful in this case.

All size businesses realize how important flexibility and efficiency is. Business rules help keep business process 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.



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. Business Analyst also works with the Developer role to update and create new business rules for the company.



## Agenda

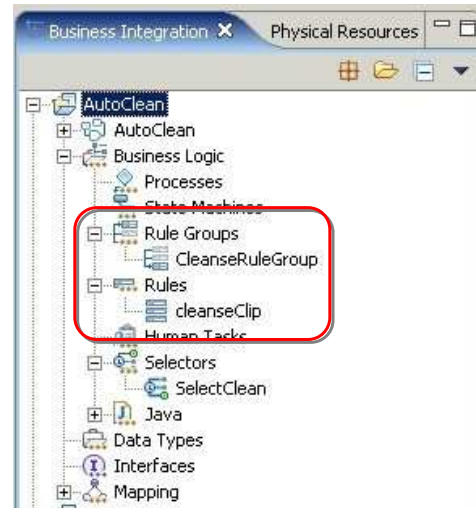
- What is a business rule
- How are business rules organized
- Why there are business rules and the roles who use them
- **Business rules in tools and runtime**
- Business rule examples
- Summary and references



This section will cover the business rules in tools and runtime.

## Tools

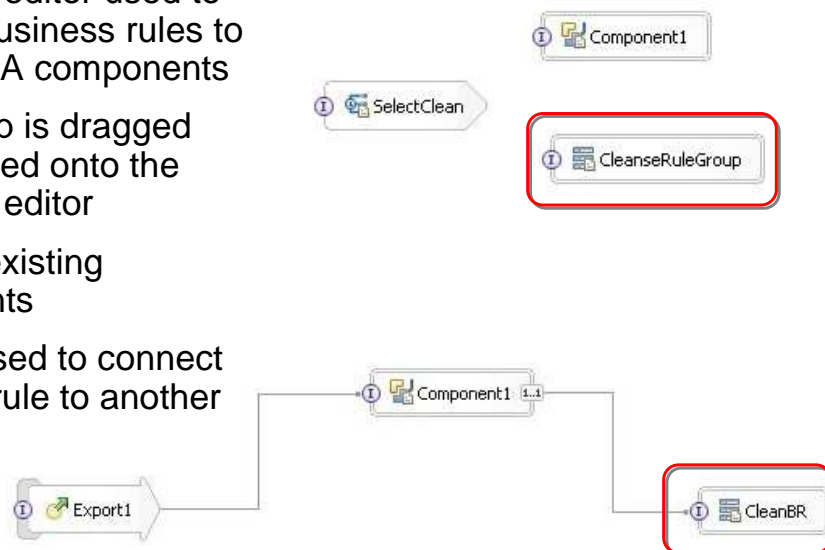
- Tools
- WebSphere Integration Developer
  - ▶ Easy to use tools for defining, executing, and managing business rules
  - ▶ Eclipse based tools for business rule development
  - ▶ Focused on the more technical developer role
  - ▶ Business Integration view used mostly for business rule interaction



WebSphere Integration Developer makes up the tools portion for business rules development or creation, along with the business rule manager Web tool, which is connected to the runtime. It is focused on the developer role previously mentioned in this presentation. WebSphere Integration Developer is an Eclipse based development environment that is easy to use to create and manage rule groups, rule sets, and decision tables. The WebSphere Integration Developer Business Integration View provides access to most of the business rule editors and the business rule creation wizards. To view what files are actually being created, you can open a Physical Resources view.

## Tools

- Assembly editor used to connect business rules to calling SCA components
- Rule group is dragged and dropped onto the Assembly editor
- Wired to existing components
- Exports used to connect business rule to another module



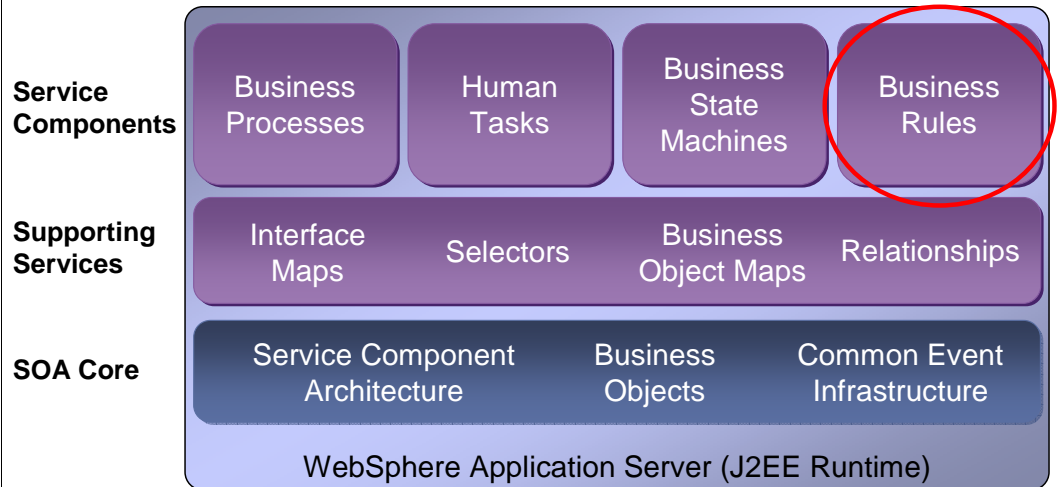
In addition to the business rule editors, the Assembly editor will be used to connect business rules into a process. The rule group can be easily dragged and dropped from the Business Integration view on to the Assembly editor. Rule sets and decision tables cannot be dropped onto the Assembly editor and an error will be thrown if you try because they are not exposed as SCA components. Any call to an SCA component must go through the rule group in order to access the business rule, which is why a rule group can only be placed on the Assembly editor. Once the rule group is dropped onto the Assembly editor, it can be wired to calling components in the same module, or an export can be wired to connect it to the import of another module. This allows an SCA module in a different component to call the business rule

## Runtime

- WebSphere Process Server runtime exposes business rule 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 a straightforward Web tool
  - ▶ It is included with WebSphere Process Server V6.0 as an optional install

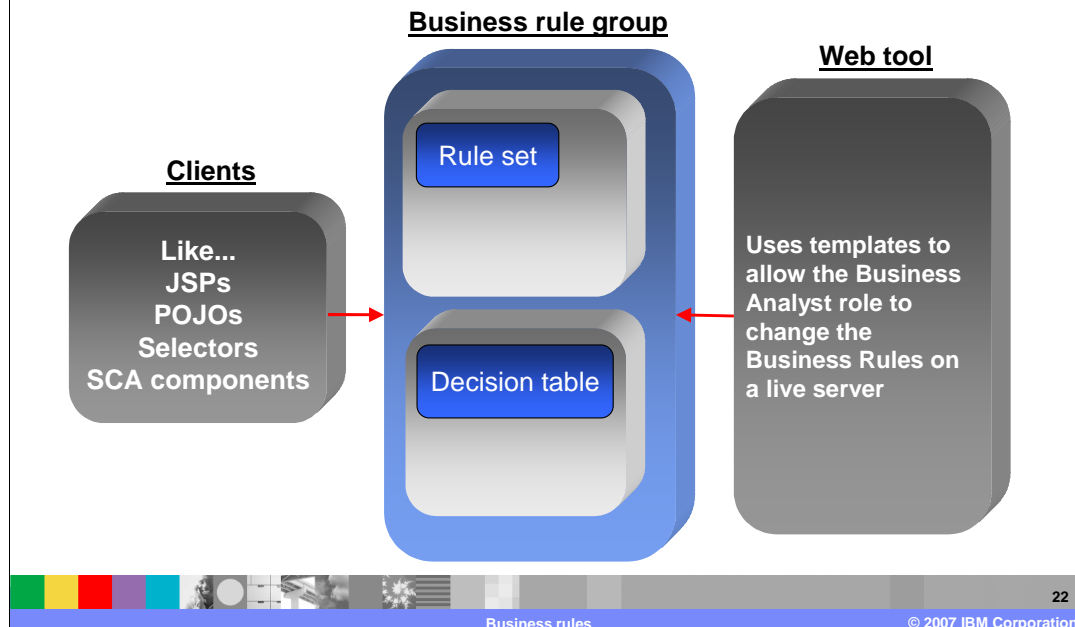
WebSphere Process Server V6 provides the runtime environment. Once a business rule is deployed to the WebSphere Process Server, the business rule group exposes the business rule as an SCA service. The WebSphere Process Server handles 80% of typical integration rules, but does not handle the other 20% of complex rules, like inferencing. IBM research indicates that almost all the Business Rules used today fit within that 80% supported by the WebSphere Process Server. Complex rules, like inferencing, are left to vendor rule engines. 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 do 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 utilize each others services.

## On runtime



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 will be 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
  - ▶ Via 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"/>

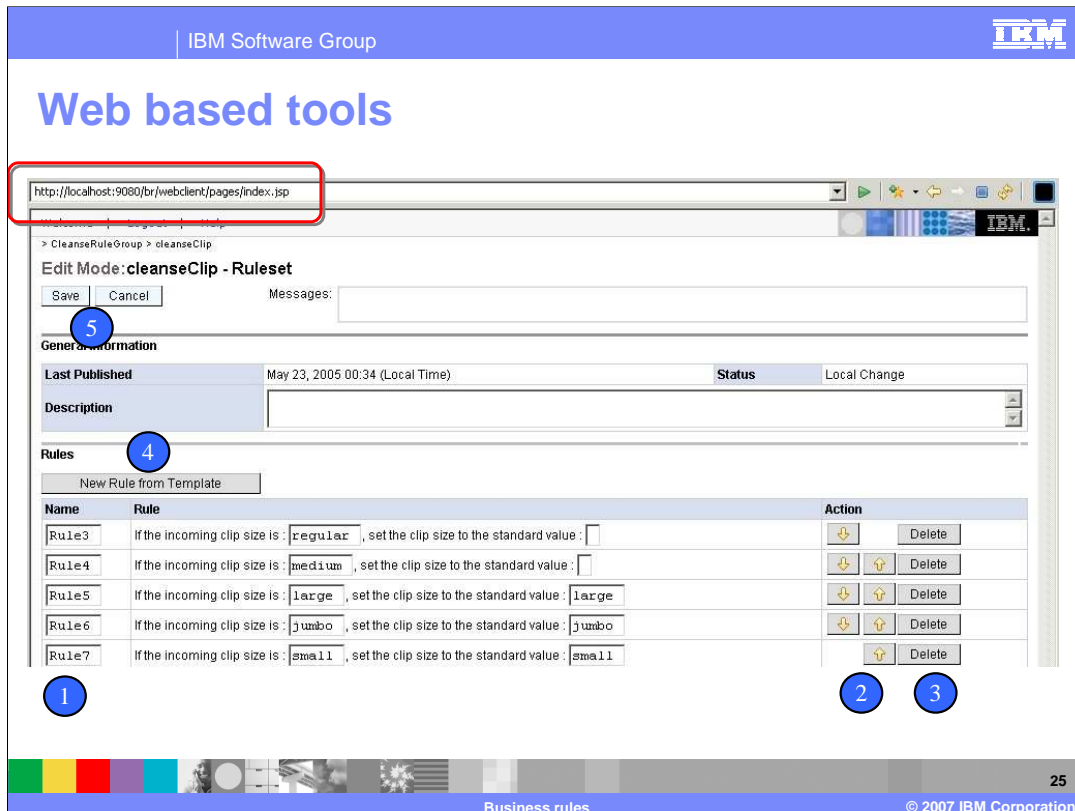
Templates are the key between WebSphere Integration Developer and the business rules manager Web tool. Templates allow Business Analysts to change business rules, including rule sets and decision tables, without having to restart the project or application server using the runtime business rule manager 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.

## Web based tools

- 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 a very easy way 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 install (installBRManager.jacl)
- 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

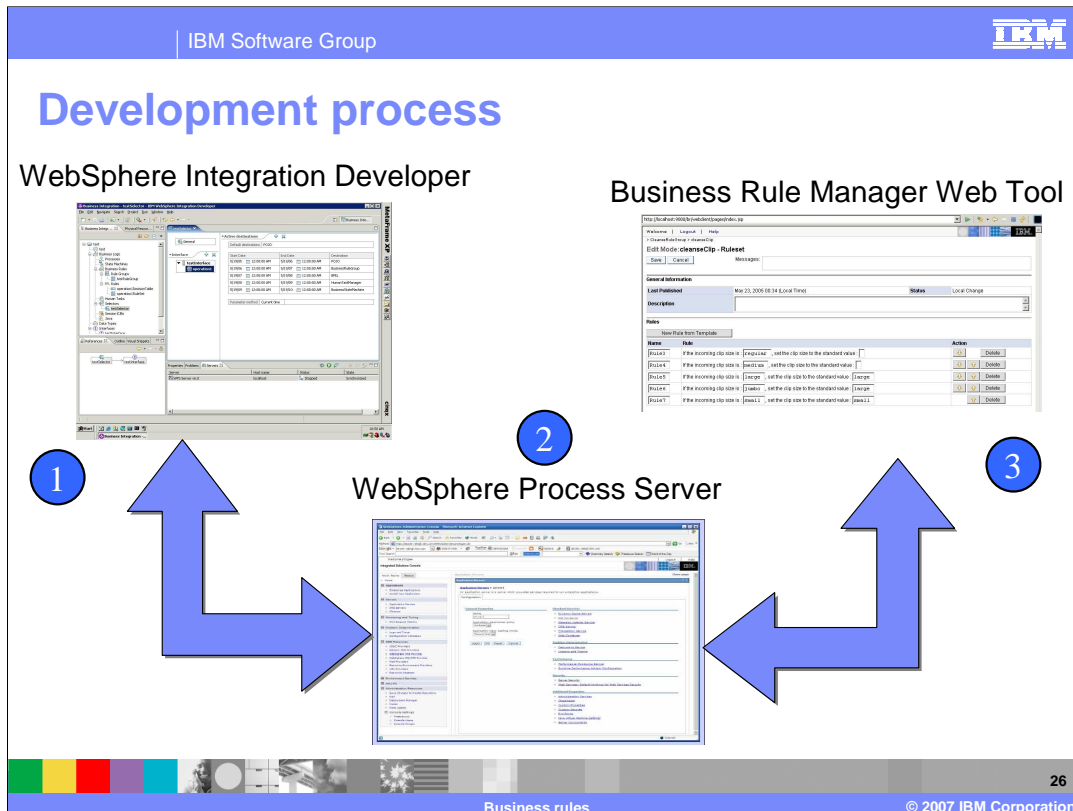
The business rule manager Web tool is built specifically for the Business Analyst. It has a manageable subset of full authoring functionality found in WebSphere Integration Developer. The only way a Business Analyst can see rules in the Web tool is if templates are built by the developer. This gives 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 install by running the installBRManager.jacl. 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.





The business rules manager Web tool can be accessed by opening a browser and entering the URL shown above circled in red. 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 will be 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 large 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



This graphic depicts how all the pieces work together in the development process.

1: The process starts with the Developer role either importing or creating new business rules in WebSphere Integration Developer. The Developer can then test the business rules using the integrated WebSphere Process Server runtime before deploying those business rules to a pre-production or production WebSphere Process Server runtime.

2: The pre-production or production WebSphere Process Server runtime runs the business rules along with any other business integration applications that access the business rules. Access to the deployed application is through the Administrative Console Enterprise Applications list, where you can stop, start, and uninstall the application. A business rule manager application comes with the WebSphere Process Server (BRManager.war). However if you want to use it, you must install it by running `installBRManager.jacl`.

3: The business rule manager Web tool is an installed enterprise application made up of JSP and servlets that pull and push business rules to and from the WebSphere Process Server. It provides the same functionality as the tools and allows the Business Analyst the power to change business rules templated variables, the order of rules, delete rules, and create new rules from the template. In addition, the Business Analyst can Save and Deploy those changes without having to restart the application or the application server the business rules are running on.

## Agenda

- What is a business rule
- How are business rules organized
- Why there are business rules and the roles who use them
- Business rules in tools and runtime
- **Business rule examples**
- Summary and references

This section will provide some examples of business rules.

## Examples

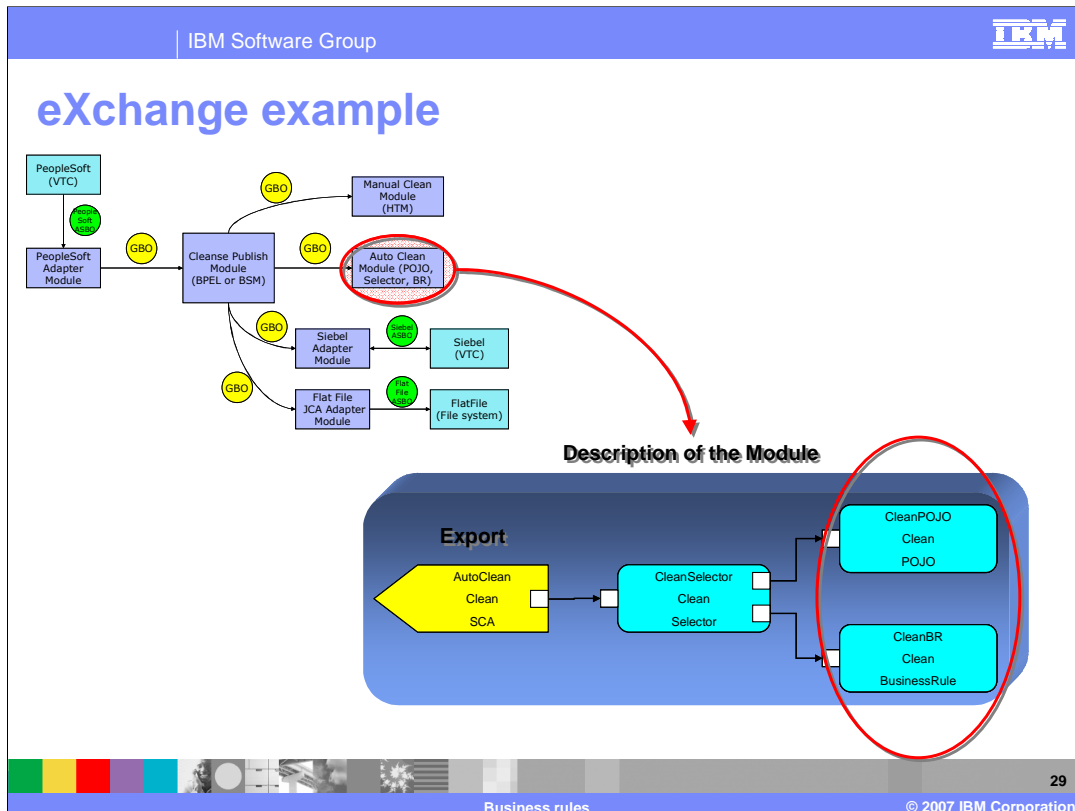
- Rent-a-car corporate discounts
  - ▶ Able to change discount amounts during peak dates and times
- Insurance agency
  - ▶ Ability to invoke a process for an insurance claim that happened in the past
- Updating the company's Web site
  - ▶ Schedule specific dates that a Web site will update to the next development version (on planned future dates) without the end user knowing
- eXchange™ Scenario
  - ▶ Currently a Plain Old Java Object (POJO) does the work of "auto cleaning" up data to be published for appropriate eyes to read (customer vs. employee).
  - ▶ After a certain date, a switch will be made to use the Business Rule instead of the POJO
  - ▶ The same rules being processed by the POJO can now be editable by a Business Analyst (using the Business Rule Web tool)

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



The following is a description of the eXchange Scenario that you can use in the Autoclean lab:

Currently a POJO does the work of “auto cleaning” up data to be published for appropriate eyes to read (customer or employee).

After a certain date, a switch will be made to use the Business Rule instead of the Plain Old Java Object (POJO)

The same rules being processed by the POJO can now be editable by a Business Analyst using the business rule manager Web tool

## Agenda

- What is a business rule
- How are business rules organized
- Why there are business rules and the roles who use them
- Business rules in tools and runtime
- Business rule examples
- **Summary and references**

This section will provide a summary and references.

## 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

Business rule examples

## References

- <http://www.brcommunity.com>
  - ▶ Good articles on the latest Business Rule happenings
- <http://www.businessrulesgroup.org>
  - ▶ Many links and references for Business Rule matters
- WebSphere Integration Developer – Help Contents
  - ▶ Find answers to tools questions with links to working samples and tutorials on Business Rules



## 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](#)

## 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

eXchange, 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 2007. 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.