



IBM Software Group

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

## *Selectors*



@business on demand.

© 2007 IBM Corporation  
Updated April 30, 2007

This presentation will cover WebSphere Process Server and WebSphere Integration Developer V6.0 selectors.

## Goals

- Introduce selectors with respect to WebSphere Integration Developer V6.0 and WebSphere Process Server V6.0
  - ▶ Cover what selectors are, what they do, and how they work
  - ▶ See how selectors work in the tools and at runtime
  - ▶ Go over some examples to better explain selector functionality

The primary goal of this presentation is to introduce selectors, focusing on WebSphere Integration Developer V6.0 to create and maintain, and WebSphere Process Server V6.0 to maintain and run them. First, a definition of selectors will be provided, followed by the concepts and an overview in order to explain what selectors are, what do they do, and how they work. This will be followed by a discussion of how selectors are used in WebSphere Integration Developer and in the WebSphere Process Server runtime. Finally, some real life examples will be provided to demonstrate how selectors work.

## Agenda

- **Selector concepts and overview**
- How do selectors work
- Selectors in tools and runtime
- Examples
- Summary



This section will cover the concepts and provide an overview of Selectors.

## Concepts and overview

- Determine dynamically which target SCA component to invoke based on some defined set of criteria
  - ▶ Currently decision is based on a date/time only
- Abstract the client application away from the target implementation
  - ▶ Change of implementation does not require the client to change
- SCA target implementations can be added to the Selector dynamically without requiring a restart of the application or server
- Think of it as a router
  - ▶ Routing BO messages to the appropriate SCA component based on a certain date (past, current, or future)

The primary function of the selector is to dynamically determine which implementation to invoke based on some defined set of criteria, currently, based on a date and time.

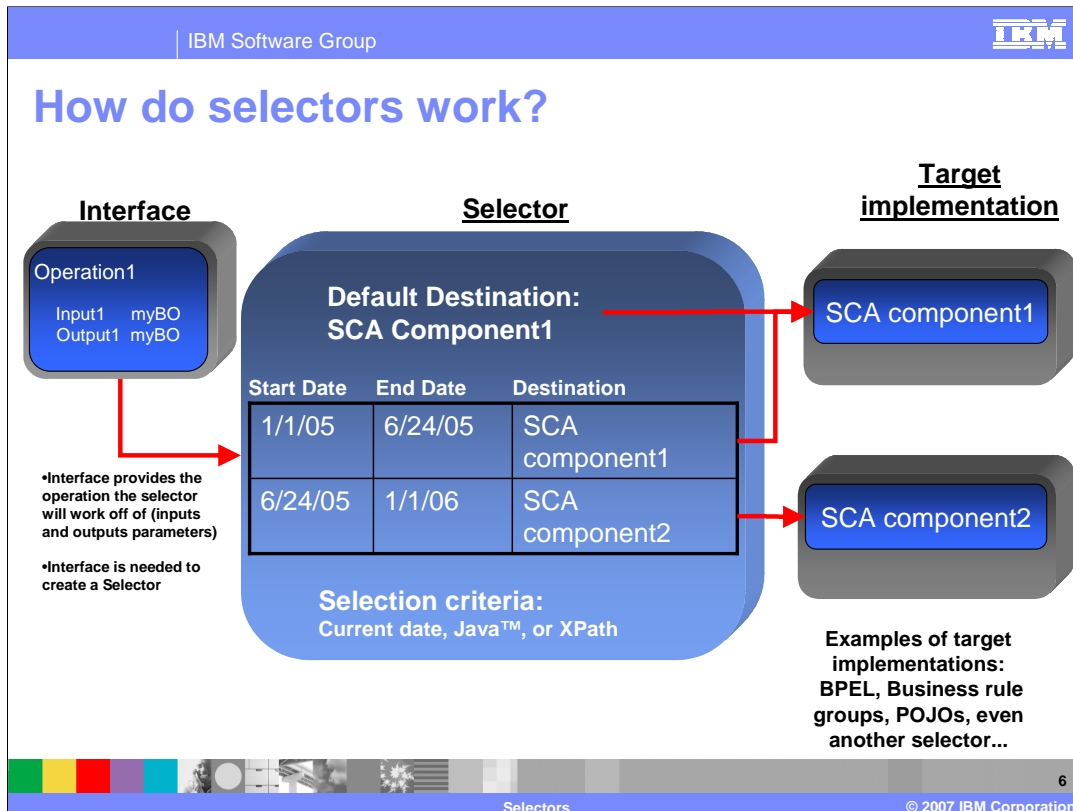
The selector decouples the client application from a specific target implementation. The abstraction of the implementation from the calling client allows a change of target implementation without a change being made to the client. It does not matter to the client how the target implementation is called, as long as it receives the correct data with which to proceed. In addition, SCA target implementations can be added to the selector dynamically without requiring a restart of the application or server, resulting in no downtime.

The selector can be thought of as a router, routing inputs and outputs to the appropriate target component based on a certain date, whether past, present, or future.

## Agenda

- Selector concepts and overview
- **How do selectors work**
- Selectors in tools and runtime
- Examples
- Summary

This section will provide an explanation of how selectors work.



This graphic details what a selector looks like.

On the left is an interface. Every selector needs an interface to operate and you cannot create a selector in the tools without an interface. The interface defines which operation (inputs and output parameters) the selector will pass to the target implementation. When a client calls the selector, it already knows what interface the selector is going to use and passes the appropriate inputs and outputs through the selector. Think of the selector as an SCA router.

Once a client call comes in to a selector, there must be some computation done. Starting from the bottom of the selector module and moving up, you notice there are three supported parameter methods; current date, Java, or XPath. Current date will use the current date and time of the system running the WebSphere Process Server. When Java or XPath is selected, a date and time will be mined using Java or an XPath parameter.

Once a date and time are selected, it goes to the date and time table to see which target implementation to use. The date and time table has a start date and time and an end date and time. When the client call comes in with a certain date that falls between the start date and end date of a certain target destination, that interface information will be passed to the target implementation.

Inside the selector is a default destination so that if none of the dates are found, you can still pass information to a component. This could be an error generating component or a basic SCA component that does minimal work on the information passed in the form of the interface.

On the right side of the graphic, you see target implementations. Selectors offer the freedom to select any SCA component, whether it is a POJO, business rule, BPEL, or even another selector. A selector can point to almost anything you can drop on the assembly editor as a target component. What separates a selector from a business rule group is that a business rule group can only point to business rules, rulesets and decision tables, where selectors can point to any SCA component.

## Agenda

- Selector concepts and overview
- How do selectors work
- **Selectors in tools and runtime**
- Examples
- Summary and reference



This section will cover Selectors from a tools and runtime perspective.

## Tools

- Tools - WebSphere Integration Developer
  - ▶ Easy to use tools for defining, running, and managing selectors
  - ▶ Eclipse based tools for selector development

The screenshot displays the WebSphere Integration Developer interface. On the left, the 'Active destinations' table is visible, showing a list of destinations with their start and end dates. On the right, the 'Business Integration' view shows a tree structure of components, with the 'Selectors' folder circled in red.

Start Date	End Date	Destination
5/19/05 12:00:00 AM	5/19/06 12:00:00 AM	POJO
5/19/06 12:00:00 AM	5/19/07 12:00:00 AM	BusinessRuleGroup
5/19/07 12:00:00 AM	5/19/08 12:00:00 AM	BPEL
5/19/08 12:00:00 AM	5/19/09 12:00:00 AM	HumanTaskManager
5/19/09 12:00:00 AM	5/19/10 12:00:00 AM	BusinessStateMachine

WebSphere Integration Developer is the development environment used to create and maintain selectors.

In the image on the right hand side you see the business integration view. Circled in red is the module menu area for selectors, where you will see any existing selectors in the module. Behind the business integration view is the physical resources view. This view shows what is going on behind the scenes and allows you to see the .sel and .selt files associated with the selector.

The image on the bottom left shows what the selector looks like in WebSphere Integration Developer. The default destination area becomes a drop down list where you can choose from the SCA components in the module.



## Tools

- Assembly editor is used to connect selectors to SCA components
- Cannot start a wire from a selector
- Dynamic wires for target components



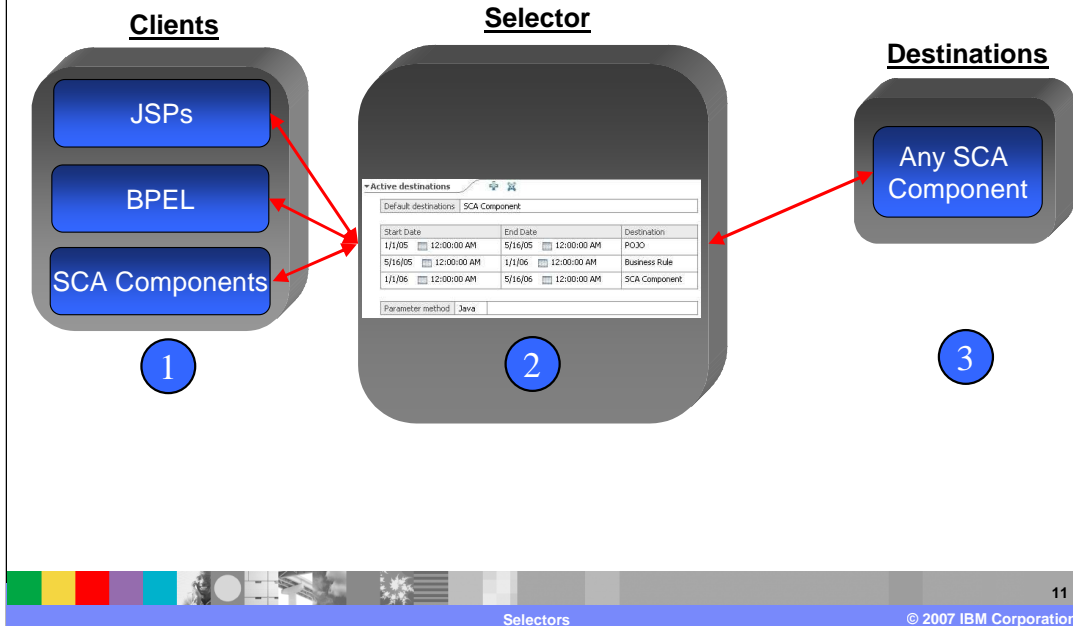
As with all SCA components, selectors can be added to the assembly editor in WebSphere Integration Developer, allowing you to connect the selector to other parts of the module or to a calling module through an export. The icon for selectors is shown in the red box at the bottom of this slide. However, there are some things you need to know about working with selectors. For instance, you cannot start a wire from a selector because they make use of dynamic wiring, meaning the selector has predefined wires or target implementations that cannot be configured with simple wiring. You would need to open the selector and add an entry to the date and time table. Dynamic wires mean hidden wires on the assembly editor. In the example shown here, the SelectClean selector had a date and time table with Component1 and the CleanseRuleGroup Business Rule.

## How do selectors work at runtime?

- Client: The client component makes a call on the selector component
- Selector: The selector component chooses which target implementation to invoke using declared selection criteria (date and time)
- Target implementation: The set of SCA components available to be invoked by the client
  - ▶ any component or import in the same module
  - ▶ any export in another module

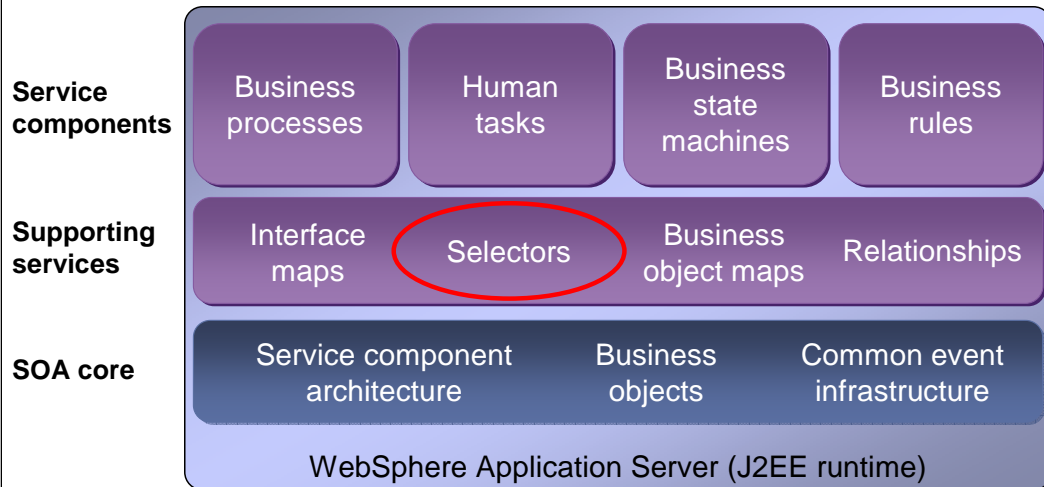
An introduction to how selectors function in the runtime environment will help you understand how they work. The client component makes a call on the selector component, which routes the call or selects the appropriate target implementation based on declared selection criteria, which is currently a date and time range only. The target implementation is then passed the interface defined information, which is already set in the selector inside the date and time table.

## At runtime



To further illustrate, the calling clients could be JSP, BPEL, plain old Java object (POJOs), or any other SCA component. The client sends the request to the selector acting as an intermediary to choose which target implementation is appropriate. Once the date and time are checked, the request is sent on to the target implementation, which could be BPEL, another selector, or an import.

## Where do selectors fit



12

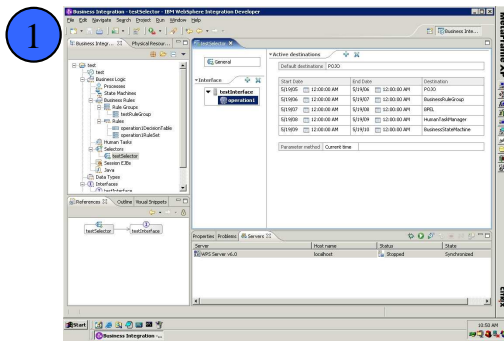
Selectors

© 2007 IBM Corporation

To understand where Selectors fit in with the rest of the components that make up the WebSphere Process Integration, this cake chart shows that selectors fall into the supporting services layer. Although a selector is an SCA component, the primary function of a selector is to support the service component layer.

## Development process

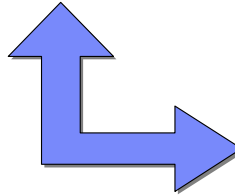
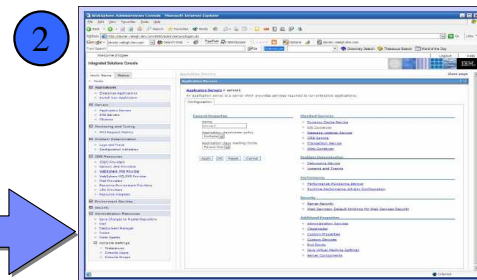
### WebSphere Integration Developer



- 3 Administrative console
- dynamically change date/time ranges
  - Add/remove target implementation
  - Export changes to development env



### WebSphere Process Server



The development process is quite simple. The first image shows WebSphere Integration Developer, where selectors are created and maintained by a person in the developer role. Next, the application is exported and deployed to a WebSphere Process Server, shown in the second image. Once running, you can use the WebSphere Application Server administrative console to change date and time ranges and target implementations. It also allows the export of those changes to a development environment so runtime and development environments are synchronized.

## Agenda

- Selector concepts and overview
- How do selectors work
- Selectors in tools and runtime
- **Examples**
- Summary



This section will provide some examples of how selectors work.

## Examples

- Discount service
  - ▶ Rent-a-car corporate discounts separated by a date/time throughout the year
- Updating UI silently
  - ▶ Keep a consistent user interface over time while the backend functionality changes, hidden to the user
- End-to-end scenario
  - ▶ Currently a POJO auto cleans up data to be published for appropriate eyes to read, but after a certain date, a selector is used to change the implementation from the POJO to a business rule

15

Selectors

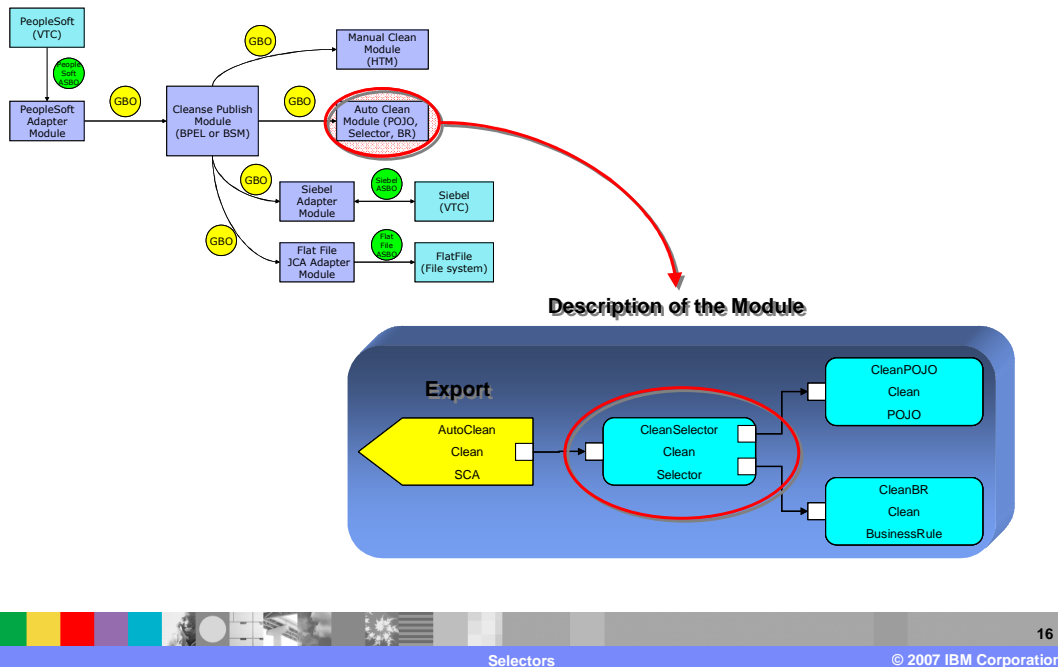
© 2007 IBM Corporation

These three examples demonstrate real-life uses of selectors. The first is a discount service; for example a car rental company wants to manage their corporate discounts separated by date and time throughout the year. During the peak summer months, the corporate discount is less than in the winter months.

The next example uses selectors for updating a Web site or program UI silently; that is, the selector keeps a consistent user interface over time while the backend functionality changes, hidden to the user.

Another example is the end-to-end scenario used in several of the tutorials for this course. Currently a POJO “auto cleans” data to be published for appropriate eyes to read. After a specific date, a selector is used to change the implementation from the POJO to a business rule.

## End-to-end Example



This graphic illustrates the example mentioned previously. The CleansePublish module acts as the client and sends a generic business object, or GBO, to be “Auto Cleaned”. Notice that the AutoClean module is connected to the Cleanse Publish Module by way of an export. This is how the “Cleanse Publish” module is able to access the AutoClean module. The GBO is passed to the CleanSelector to determine which target implementation to use, either the CleanPOJO or the CleanBR business rule. A date can be sent in with the GBO, or the current system date is evaluated to determine which target implementation to use.



## Agenda

- Selector concepts and overview
- How do selectors work
- Selectors in tools and runtime
- Examples
- **Summary**

This section will provide a summary of topics covered in this presentation.

## Summary

- Covered concepts and overview to learn what selectors are, what they do, and how they work
- Saw how selectors work in the tools and runtime
- Went over some examples to better explain selector functionality



This presentation covered concepts related to and provided an overview of selectors, including what they are, what they do, and how they work. It also provided information on how selectors work in the tools and in the runtime environment. Finally, examples were provided to better explain selector functionality.

## 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](#)



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, 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.