



IBM Software Group

WebSphere Business Modeler V6.2

Technical attributes



@business on demand.

© 2009 IBM Corporation
Updated April 21, 2009

This presentation discusses the technical attributes specification in WebSphere® Business Modeler V6.2.

Goal

- To discuss the technical attributes specification in *WebSphere Business Modeler* and how they impact the implementation model in *WebSphere Integration Developer*



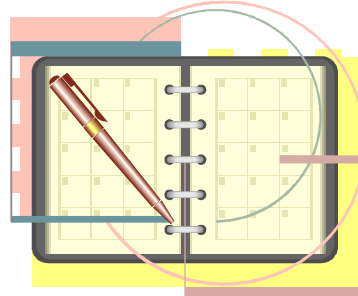
When modeling for deployment the business model is used to generate the implementation artifacts used in WebSphere Integration Developer.

The technical attributes view in WebSphere Business Modeler provides a way for the business analyst or integration developer to specify the names to be used by WebSphere Integration Developer for key implementation artifacts.

This presentation will focus on specifying the technical attributes and how they are used when generating the implementation artifacts.

Agenda

- How the technical attributes are used
- How generate the default values
- Map the technical attributes to WebSphere Integration Developer



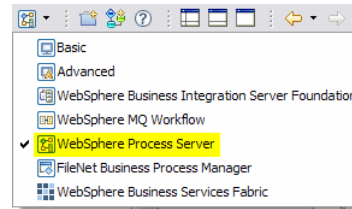
The discussion begins with a brief explanation of why there are technical attributes in WebSphere Business Modeler. You will learn how to locate them in WebSphere Business Modeler and generate the default values.

After learning about what the technical attributes are, you'll then see how they are mapped to the implementation in WebSphere Integration Developer.

With this information you can then develop the best practices for business modeling in your organization.

Technical attributes view

- Use when *modeling for deployment to WebSphere Process Server*
- Override the default interface names and namespaces
- Specify implementation specific details
 - ▶ Interfaces – WSDL
 - ▶ Data definitions - XSD
 - ▶ Used to override default code generation patterns



When a business model is created, business items are specified along with the inputs and outputs. WebSphere Business Modeler is designed such that only the information needed for the given goal is required. In many cases default names are used for the inputs and outputs. The business user might not be concerned with this level of detail, therefore the defaults are acceptable.

When the business model is exported using the WebSphere Integration Developer export type, the default names are used to create the interfaces and data definitions used in WebSphere Integration Developer. Interface names and namespaces are very important to the integration developers. Typically a software development shop has naming conventions that must be adhered to.

The technical attributes view provides the mechanism whereby the interface names and namespaces can be specified in WebSphere Business Modeler, before they are exported to WebSphere Integration Developer.

This presentation will go over the technical attributes and show you how they get mapped to implementation artifacts in WebSphere Integration Developer.

WebSphere Integration Developer

- There several places is *WebSphere Integration Developer* where the defaults are not appropriate and should be managed in *WebSphere Business Modeler*.
 - ▶ InputCriterion and OutputCriterion
 - mapped to operation name
 - ▶ Input and Output of processes and tasks
 - used in the operation interface
 - ▶ WSDL message and part names
 - Used for the WSDL names

The screenshot displays the WebSphere Integration Developer interface. On the left, a tree view shows a 'Simple Process' containing 'Interface Partners', 'SimpleProcessPartner', 'SimpleProcess', and 'InputCriterion'. A yellow callout labeled 'operation' points to 'InputCriterion'. In the center, the 'Operation - InputCriterion' details are shown, including a table with columns for Interface, Operation, Input, and Output. A yellow callout labeled 'operation' points to the 'Input' and 'Output' rows. On the right, a table shows 'InputCriterion' and 'InputCriterionResponse' with a yellow callout labeled 'WSDL part names' pointing to the 'InputCriterion' column.

Interface	Operation	Input	Output
SimpleProcess	InputCriterion	Input	Output
		WorkOrder	WorkOrder

The screen captures shown here are from WebSphere Integration Developer.

They show some of the places where the default names from WebSphere Business Modeler are used.

The most noticeable place is in the interface list, for the interface and reference partner links. Here, the WSDL operation name is displayed. In this case it is showing the default name, InputCriterion.

Names such as Input for the input and InputCriterion for the operation name convey no meaning for the integration developer that is doing the work of creating the implementation. If all the processes and process elements are left unspecified, then everything will have the same names and debugging becomes problematic. Integration developers generally have strict naming conventions that must be followed. The names can be changed in WebSphere Integration Developer after the import, but this will make it difficult to synchronize the models with subsequent iterations. It is best to work top down, making the changes to the model and the interfaces in WebSphere Business Modeler.

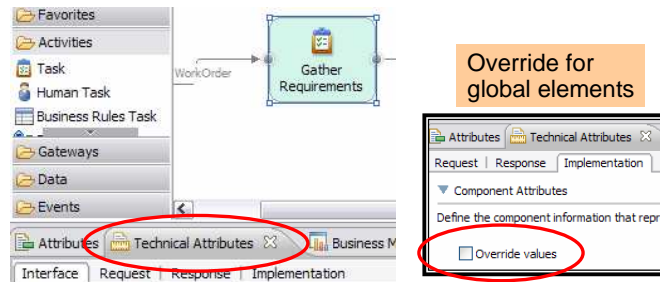
The next few slides in this presentation will review the technical attributes specification and show you how to generate the default values so that you can see what they are and where they come from. You will then be able to override values to meet your specific requirements.

Technical attributes

- Global entity
 - ▶ Open from the project tree
 - ▶ Exception is the process element



- Local
 - ▶ Select the element in the process diagram
 - Locate the technical attributes tabs



If you are working with a global entity, with the exception of the business process element, open the element from the project tree and then navigate to the technical specification tab at the bottom of the pane. Notice that the 'apply default values' button is on the very first page of the global technical attributes specification.

If you are working with a "global process" element, when you open it, a business process diagram is opened instead of a specification form. To work with the attributes of the business process then select the canvas background. The attributes and technical attributes are in the viewer section below.

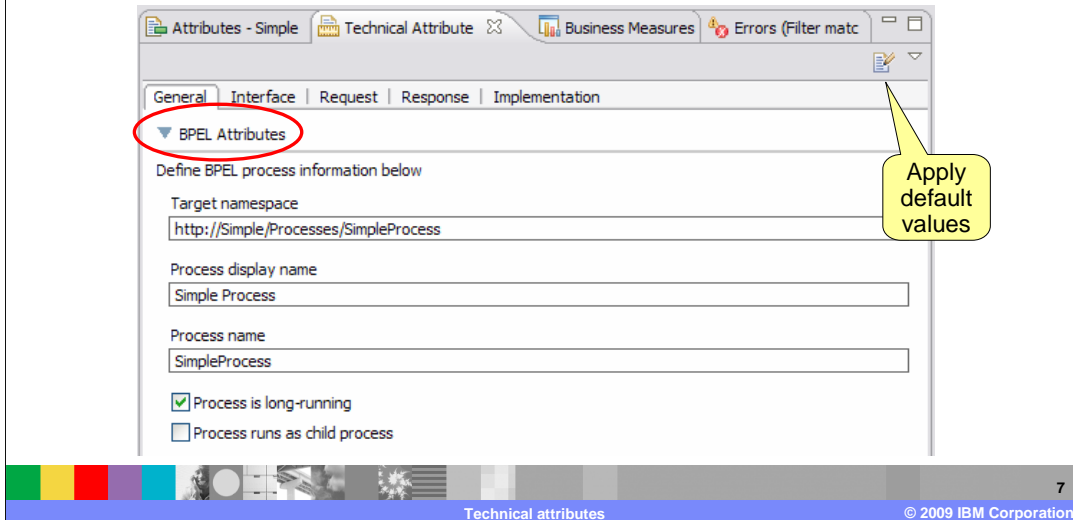
To work with the attributes of any of the local elements in the business process, select the element and then look for the technical attributes tab in the view section below.

It might be that you have a global element in your business process diagram. If you select it and try to work with the technical attributes the same as a local element, then you can override the global values. The override is only for the instance of the global entity used by the current business process. See the inset on the bottom right.

The business process - general



- The tabs and data will vary depending on the element



To work with the technical attributes select the element in the business process diagram. In the case of the global business process, select the canvas background.

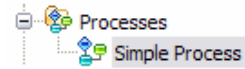
Initially the fields are all blank. To find out what the default values are you can use the menu option from the view, 'apply default values.' When you select this option, you'll be asked if you want to overwrite the existing values. If it is your first time, then you can safely say yes. If you have already customized some of the values, then this option will let you keep those and generate the values for the empty fields.

Generating the default values is a great place to start. It lets you see the values that are going to be used for generating the WSDL and XSD and some of the other runtime options that are available.

Here you see that for the global business process there is an option to run as a long-running process and or as a child process of the calling process. The average business user won't know what to do with these options but the experienced integration developer will recognize them immediately. These options and how they are used in WebSphere Process Server are documented in the WebSphere Process Server information center. The focus here is on the interface components.

The technical attributes being defined here, are the BPEL attributes. These will effect the namespace, display name and the actual process name in the BPEL application space. BPEL is a business process execution language that is an XML markup language used to describe business process flows. It uses WSDL to describe the interfaces and XSD to describe the data definitions.

The business process - interface



- PortType
 - ▶ The name of the interface used in WebSphere Integration Developer
- Callback
 - ▶ Used to return information in the case of a one-way request

The screenshot shows the 'Technical Attribute' view in WebSphere Integration Developer. The 'Interface' tab is selected, displaying the 'WSDL Interface Attributes' section. The 'Target namespace' is set to 'http://Simple/Processes/SimpleProcess/SimpleProcessInterface'. The 'PortType name' is 'SimpleProcess'. A yellow callout bubble points to the 'PortType name' field with the text 'Interface name'. An orange callout bubble points to the 'WebSphere Integration Developer' logo in the bottom right corner. The 'Request' tab is also visible, showing fields for 'Callback PortType target namespace' and 'Callback PortType name'. The bottom of the screenshot shows a tree view with 'SimpleProcess' selected under 'Interfaces'.

Moving to the interface tab will display a new set of fields.

As mentioned previously, WSDL is used to describe the interfaces in BPEL. Here you see that the namespace is extended from the namespace used for the business process and that the port type is given the name of the business process.

In WSDL a port type is synonymous with interface. Here you just define the name and the namespace.

The callback namespace and port type are used to define the names to be used when you have a process that has a one-way operation. A one-way operation is an asynchronous message that does not wait for a return. In this case the callback namespace and port type are used by the receiving service if they need to return information at some point in the future.

You can specify a one-way operation on the request tab of the technical attributes view.

Using the name of the business process for the port type name is acceptable. There might be a need to use a namespace, other than the one provided. In that case the integration developer can change it here.

IBM Software Group IBM

The business process - request

Attributes - Simple Process | Technical Attributes | Business Measures | Errors (Filter matched 0 of 0 items)

General | Interface | Request | Response | Implementation

▼ Input criterion section

Select an input criterion in the table to see the WSDL and BPEL information in the sections below

Input criterion name	Corresponding output crit...	Valid operation type	Selected operation type
Input Criterion	Output Criterion	Request and response, o...	Request and response

▼ Operation type

Select the type of operation

Request and Response operation
 One way operation

▼ WSDL Attributes

Define the WSDL interface information

Operation name

▼ WSDL Message Details

Define the WSDL message that represents the input of an operation

Message name

Part name

▼ BPEL Attributes

Define the BPEL receive activity information

Activity display name

Activity name

Row must be selected to see sections below

Business process request maps to a BPEL receive activity

Technical attributes © 2009 IBM Corporation

Looking at the request specification tab, there are several things to notice.

First, there is the 'operation type' option. This is where you can specify a one-way operation if you need to.

Second, take a look at the names for the operation, message and part in the WSDL attributes section. By default they are named InputCriterion. InputCriterion is the default name used for the input criteria which is specified in the input logic section of the regular attributes.

This will pose problems for the developer working in WebSphere Integration Developer. If nothing is done, all of the business processes and tasks will have the same operation name.

You can change the names of the operation, message and part here, but it is better to make sure that the input criteria is named properly.

Next, notice the section for the BPEL attributes. The convention used for the activity name is acceptable and once the InputCriterion is named properly, the name will provide useful information. Remember that a business process can have multiple entry points. An InputCriterion is used to specify a single entry point. In this case the internal name for the receive activity is qualified with the name of the InputCriterion but the name displayed in the BPEL flow will only show the first part of the name.

With the business process, the request specification gets mapped to the BPEL receive activity.

IBM Software Group IBM

The business process - response

Processes
 Simple Process

Attributes - Simple | Technical Attribute | Business Measures | Errors (Filter match) | Synchron...

General | Interface | Request | **Response** | Implementation

▼ Output criterion section

Select an output criterion in the table to see the WSDL and BPEL information in the sections below

Output criterion name	Corresponding input criterion name	Response type
OutputCriterion	InputCriterion	Response

▼ WSDL Attributes

Define the WSDL interface information

▼ WSDL Message Details

Define the WSDL message that represents the output of an operation

Message name

Part name

▼ BPEL Attributes

Define the BPEL reply activity information

Activity display name

Activity name

Technical attributes 10 © 2009 IBM Corporation

Row must be selected to see sections below

Business process response maps to a BPEL reply activity

The response specification tab for a business process is very similar to the request specification.

There is a section for the WSDL attributes and a section for the BPEL attributes.

The concerns with the default names are the same.

With the business process, the response specification gets mapped to the BPEL reply activity.

The business process - implementation

Implementation type

- ▶ For known element types, such as process, human task and business rules task
 - The type is fixed
- ▶ For a generic task
 - There is a drop down selection menu

Attributes - Simple Process | Technical Attributes | Business Me

General | Interface | Request | Response | Implementation

▼ Component Attributes

Define the component information that represents the target implementation

Component display name
Simple Process

Component name
SimpleProcess

Component description
SCA component; assembly diagram

Implementation type
Process

▼ Implementation attributes

Define the implementation information

Implementation description

Technical attributes | © 2009 IBM Corporation

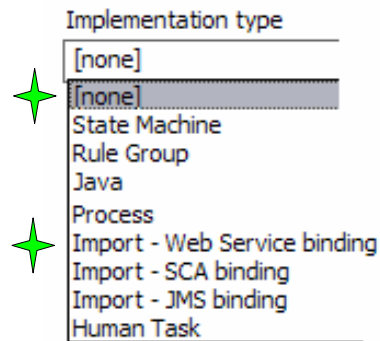
For the modeling elements that have concrete implementations in WebSphere Process Server, such as the process, human task and business rules task, the implementation type is set and cannot be changed. When working with the generic task, this field can be set by the business user using a drop down selection menu.

There is one more area where important design information can be captured and forwarded to the implementation development team; the component description. In terms of implementation, the component is an SCA component. The description entered here is visible by selecting the component in the SCA assembly diagram and looking at the properties below.

The implementation description is not used at this time.

Implementation type

- Default type is *none*
- “None” and “Import – Web service binding”
 - ▶ Map to mediation flow components in the implementation assembly
 - Extra level of indirection introduced
 - Use *WebSphere Service Registry and Repository* to get endpoint information
- The other implementation types are mapped to implementation skeletons based on their type



Component in the assembly diagram
_impl



The presentation on “model synchronization” discusses how the implementation type is used to determine what to generate in the implementation and which module to place it in. Those artifacts such as the human task and business rules, which can be mapped to concrete implementations, are placed in the *business logic* module. Those types that cannot be mapped to concrete implementations, such as the Java™, State Machine, or none, are placed in the implementation module. Implementation skeletons are created for the artifacts that are placed in the implementation module and the component is wired up in the assembly diagram.

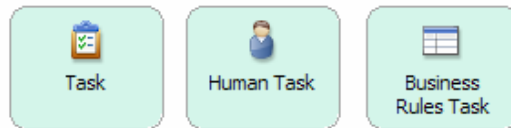
With the introduction of the new “interactive process design” feature, a new code generation pattern has been added.

With interactive process design the business process can be directly deployed and run on a test server. When the running business process flow reaches a generic task, it tries to invoke it. By default it will use the interface name and attempt to locate the endpoints in the WebSphere Service Registry and Repository. In WebSphere Process Server, the redirection to the WebSphere Service Registry and Repository is achieved using the mediation flow component.

The underlying assumption is that if the task is a Web service then the WebSphere Service Registry and Repository is used to locate the service implementation at runtime. The second assumption is that if the implementation type is not known, then it will probably be some other kind of service that can be located using a mediation flow component.

This is covered again when learning about interactive process design.

Task, human task, business rules task

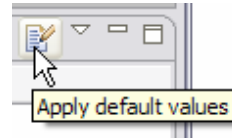


- The technical attributes are essentially the same as for the process element
- Human task
 - ▶ No specification in the technical attributes for the implementation

With respect to the WSDL and BPEL names, the generic task, the human task and the business rules task are specified the same as for the business process. The business process has an extra specification for general information and the human task does not have an implementation specification.

Best practice when modeling for deployment

- Provide your names for
 - ▶ Inputs and outputs
 - ▶ OutputCriteria and InputCriteria
 - ▶ Specify all the names
- Generate the default names for the technical attributes
- Fully specify the names in the technical attributes
 - ▶ Operation, message, part
- Provide descriptions where possible
 - ▶ They help the integration developer down stream



When using WebSphere Business Modeler to model for deployment to WebSphere Process Server, it is important to know which modeling elements are used for generating the names and interfaces for the implementation artifacts.

There are three places that you need to check; the input and output names for the processes and tasks, in the input criteria in the input logic of the processes and task and the WSDL names in the technical attributes pages.

Once the names in the model are specified, then generate the default values so you can see what they are and change them if necessary.

Making sure that these names are completely and correctly specified will simplify and accelerate the implementation and deployment of your business processes.

The remainder of this presentation will show some examples.

Example – input logic and WSDL request

- Input and output criteria specified on the input and output logic pages

The screenshot displays the 'Input Logic' configuration page for a simple process. The 'Input criteria' section shows a table with the following data:

Name	aNewWorkRequest	Criterion
RequestNewWork	<input checked="" type="checkbox"/>	aNewWorkRequest

Below this, the 'WSDL Attributes' and 'BPEL Attributes' sections are visible. The 'WSDL Attributes' section includes:

- Operation name: RequestNewWorkOp
- WSDL Message Details:
 - Message name: RequestNewWorkMsg
 - Part name: RequestNewWork

The 'BPEL Attributes' section includes:

- Activity display name: Simple Process
- Activity name: SimpleProcess_RequestNewWork

Yellow callouts highlight specific elements: 'Technical attributes after default values are applied' points to the 'Input Logic' tab, and 'Suffix added manually' points to the manually entered suffixes in the WSDL message name and BPEL activity name.

Here you can see where the input criteria for the simple business process has been given the name of RequestNewWork in the input logic specification.

When the default values for the technical attributes are applied, the value from the input criteria is used to populate the WSDL and BPEL names. The suffixes are added to the operation name and the message name manually. This makes all three names unique and provides clues that are helpful when developing the implementation.

The WSDL message name is not exposed to the integration developer in WebSphere Integration Developer, but it is a good practice to make sure it has a unique name, distinct from the operation and the part.

Example – output logic and WSDL response

- Input and output criteria specified on the input and output logic pages

The screenshot displays the configuration interface for a process. The 'Output Logic' tab is selected, showing a table of output criteria. A callout bubble points to the 'Output Logic' tab, stating 'Technical attributes after default values are applied'. Below this, the 'WSDL Attributes' section is expanded, showing 'WSDL Message Details' with fields for 'Message name' and 'Part name'. A callout bubble points to the 'Message name' field, which contains 'WorkRequestResultsMsg', stating 'Suffix added manually'. The 'BPEL Attributes' section is also visible, showing 'Activity name' set to 'SimpleProcess_WorkRequestResults'.

Name	Probabilit...	aCompletedWorkRequest	Criterion
WorkRequestResults		<input checked="" type="checkbox"/>	aCompletedWorkRequest

WSDL Attributes

Define the WSDL interface information

WSDL Message Details

Define the WSDL message that represents the output of an

Message name: WorkRequestResultsMsg

Part name: WorkRequestResults

BPEL Attributes

Define the BPEL reply activity information

Activity display name: Simple ProcessReply

Activity name: SimpleProcess_WorkRequestResults

The names used for specifying the interface details for the response is similar to the request. The names are based off the output criteria and a suffix is manually added to the message name.

Example - input and output names

- Show up in WebSphere Integration Developer
 - Simplified view of the WSDL using WSDL editor

The screenshot displays the 'Inputs' and 'Outputs' tabs in the WSDL editor. The 'Inputs' tab shows a table with the following data:

Name	Associated data	State	Min
aNewWorkRequest	WorkOrder		1

The 'Outputs' tab shows a table with the following data:

Name	Associated data	State
aCompletedWorkRequest	WorkOrder	

The simplified WSDL view shows the following structure:

```

SimpleProcess
  RequestNewWork
    input aNewWorkRequest WorkOrder
    output aCompletedWorkRequest WorkOrder
  
```

Here you can see how the names used for the input and the output of the business process element gets used as part of the WSDL interface description in WebSphere Integration Developer.

WSDL editor – default and simplified views

WebSphere Integration Developer

View: Simplified

SimpleProcess		
RequestNewWork		
input	aNewWorkRequest	WorkOrder
output	aCompletedWorkRequest	WorkOrder

View: default

SimpleProcess		
RequestNewWork		
input	RequestNewWork	RequestNewWork
output	WorkRequestResults	RequestNewWorkResponse

Technical attributes © 2009 IBM Corporation

WebSphere Integration Developer has several tools for viewing and editing the WSDL files. The two most common are the interface editor and the WSDL editor.

The interface editor is a high level editor for creating the interface whereas the WSDL editor is designed for the developer that understands the Web service description language and needs to see and work with the WSDL files directly.

With the two views of the WSDL editor, you can see how the input and output are used in the simplified view and the WSDL part names are used within the default view.

The WebSphere Integration Developer interface editor uses the simplified view too.

The BPEL editor

- Uses the simplified view of the WSDL
 - ▶ Input and output names from modeler are displayed
 - ▶ The WSDL operation name is displayed with the suffix, as specified

The screenshot shows the BPEL editor interface. On the left, a panel titled 'Operation - RequestNewWorkOp' displays a 'Description' section with a table of technical attributes. On the right, a tree view shows the process structure, including 'Simple Process', 'Interface Partners', 'SimpleProcessPartner', 'SimpleProcess', 'RequestNewWorkOp', 'Reference Partners', 'GatherRequirementsPartne', 'GatherRequirements', 'AutomaticWorkPartner', 'AutomaticWork', 'Variables', 'WorkOrderVariable', 'Correlation Sets', and 'Correlation Properties'.

Operation - RequestNewWorkOp	
Description	
Interface	SimpleProcess
Operation	RequestNewWorkOp
Input	aNewWorkRequest
Output	aCompletedWorkRequest

The BPEL editor in WebSphere Integration Developer will also display some of the names that originate in WebSphere Business Modeler.

Here you can clearly see that the operation name that was specified with the OP suffix is being used.

You can also see that the user-specified input and output names are being used in the description section that is displayed when the operation is selected.

Decision and multiple choice

- Default is to represent as BPEL links
- Option to represent as a BPEL switch activity

The image shows two diagrams and a screenshot of the IBM Business Process Modeler interface. The top diagram is a simple diamond-shaped decision node with two outgoing paths labeled '50.0% Yes' and '50.0% No'. The bottom diagram shows a similar diamond node labeled 'Decision:2' with two outgoing paths labeled '50.0% Output Condition' and '50.0% Output Condition:2'. To the right is a screenshot of the 'Technical Attributes' dialog box for a 'BPEL Switch activity'. The dialog has a 'General' tab and a section for 'BPEL Attributes'. A checkbox labeled 'Represent exclusive decision as BPEL Switch activity' is checked. Below this are fields for 'Activity display name' and 'Activity name'. A yellow callout bubble points to the 'BPEL Attributes' section with the text 'Also known as Choice'. Another yellow callout bubble points to the 'Activity name' field with the text 'Real or internal name'. The bottom of the screenshot shows a status bar with 'Technical attributes' and '© 2009 IBM Corporation'.

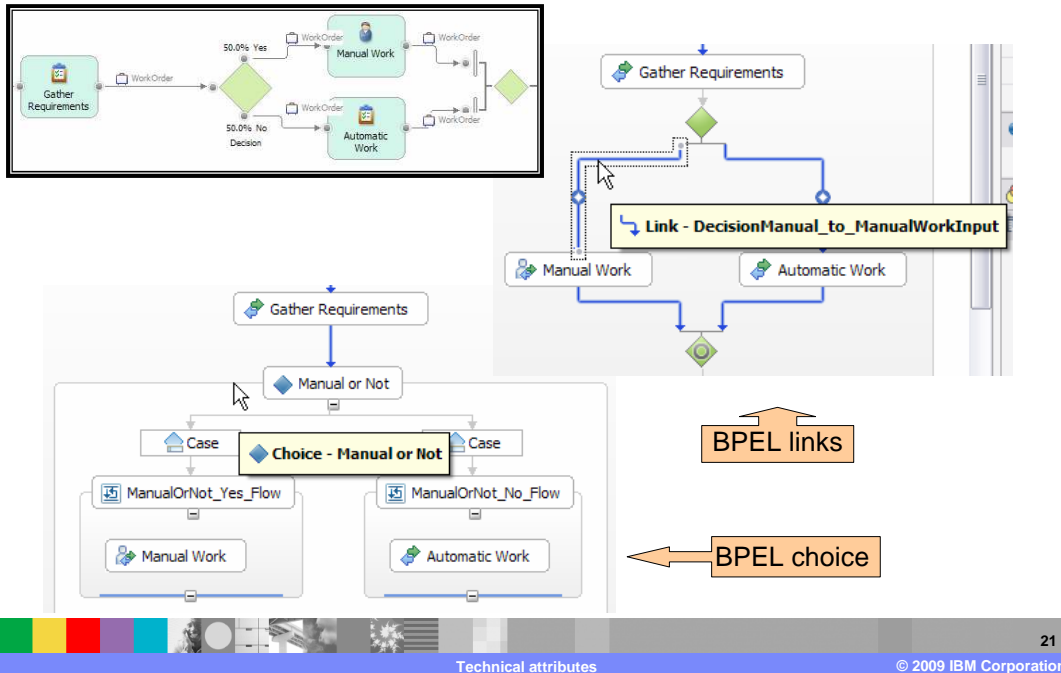
Exclusive decisions can be implemented in a couple of different ways. By default they are mapped to BPEL links.

A BPEL switch is also known as a 'choice' activity.

In addition to specifying how the decision can be implemented, you can define the display name and the real name of the activity.

The real name is the one used internally.

BPEL links and switch



Shown here are the two different ways of implementing the exclusive decision shown in the inset on the upper left.

With the links, the logic is embedded in the links. With the switch or choice, the logic is in the case node. The choice is easier to read and manage if there are many paths. It also provides a clear 'otherwise' case which is the default if all the other cases fail.

The BPEL links are new with V6.2.

Repositories

- The repository used in WebSphere Business Modeler maps to a BPEL variable



Technical attributes

Attributes - Local Informati Technical Attributes

General Repository Details

This section provides general information about this local repository.

Name
Local Information Repository

Description

Associated data
WorkOrder

Attributes - Local Informati Technical Attributes

General

BPEL Attributes

Define BPEL Variable information below

Variable name
ListOf1ToUnboundedWorkOrdersVariable

ListOf1toUnboundedWorkOrdersVariable

Attributes - Lo Technical Attri Business Meas Errors (Filter) Synchronize Asset Reposit

General Repository Details

This section shows the detailed information for this local repository.

Capacity
Unlimited

The items are in a specific order The items are unique

Technical attributes © 2009 IBM Corporation 22

The repository element in WebSphere Business Modeler is mapped to a BPEL variable in WebSphere Integration Developer.

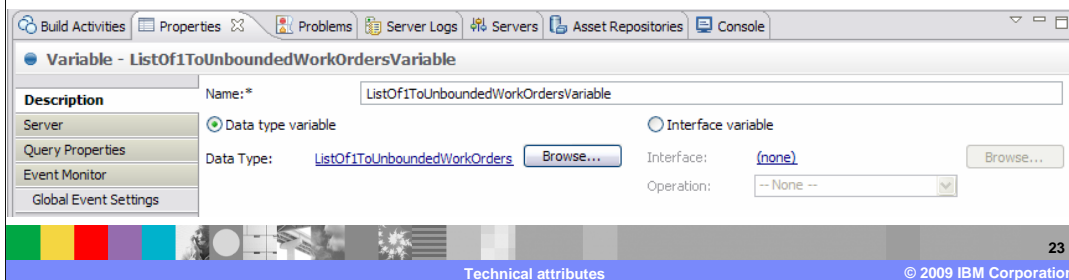
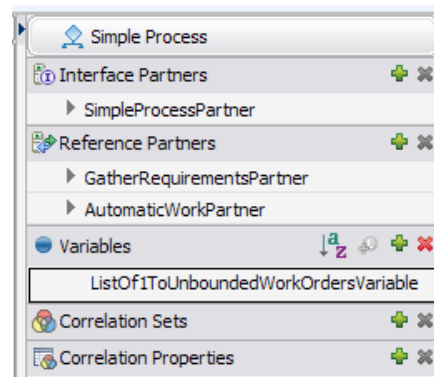
In the technical attributes the only thing that can be specified is the name, which can be very significant.

Shown here is the default variable name that is generated for a collection of WorkOrders. The multiplicity is indicated by selecting the unlimited option, highlighted with the red circle. The name is ListOf1ToUnboundedWorkOrdersVariable. This name describes the function of the variable very well, but might not meet the needs of the implementation team.

If it needs to be changed to something simpler, such as 'WorkOrders,' this is the place to do it.

BPEL variable

- The variable name is displayed in the menu and the properties.
- It will also be used when composing Java snippets or manipulating data with assigns



Here you see where the variable name is most visible to the integration developer. This is what they will see in WebSphere Integration Developer if the default is used.

While loops and sub processes

- There are two more elements that have some technical attributes that can be specified in WebSphere Business Modeler

- While loop
- Subprocess



Attributes Technical Attributes

General

BPEL Attributes

Define BPEL While activity information below

Activity display name
While Loop

Activity name
WhileLoop

Default names used are the names of the elements

Attributes Technical Attributes

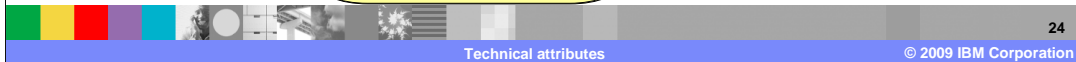
General

BPEL Attributes

Define BPEL Scope activity information below

Activity display name
Subprocess

Activity name
Subprocess



The last two elements to consider are the while loop and the local subprocess. The only technical attributes available are the names. You can specify both the display name and the real name.

Summary

- It is all in the names
- Names created in WebSphere Business Modeler effect what is created in WebSphere Integration Developer
- Take home lesson
 - ▶ Completely specify
 - Input and output names
 - Input and output criteria names
 - WSDL names in technical attributes
 - Ensure names are unique
 - ▶ Generate the default values
 - Before specifying the WSDL names in the technical attributes



25

Technical attributes

© 2009 IBM Corporation

Names are extremely important when developing software systems. Interface names are the most important because they are viewed and used by the external world, the client of the business process or the service. The interface name and the names of the input parameters indicate what the function can do, what it needs in order to accomplish the task and what you can expect to receive when its done.

With WebSphere Business Modeler it is very easy to create a business model and export it so that it can be implemented using WebSphere Integration Developer. There are no validation checks on the names. Default names are created and can be used in the implementation without change. The default names however are not very descriptive in WebSphere Integration Developer. What this means for the business user creating business models in WebSphere Business Modeler, is that they must fully specify the names before exporting to WebSphere Integration Developer.

There are three places that need to be completely specified.

One, the input and output names for the processes and tasks, two, in the input criteria in the input logic of the processes and tasks and three, the WSDL names in the technical attributes pages.

Once the names in the model are specified, then generate the default values so you can see what they are and change them if necessary.

Making sure that these names are completely and correctly specified will simplify and accelerate the implementation and deployment of your business processes.

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_WBPMv62_Modeler_TechAttrib.ppt

This module is also available in PDF format at: ..\\WBPMv62_Modeler_TechAttrib.pdf



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

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both: WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

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

Other company, product, or service names may be trademarks or service marks of others.

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.

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 NONINFRINGEMENT. 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 2009. 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.

