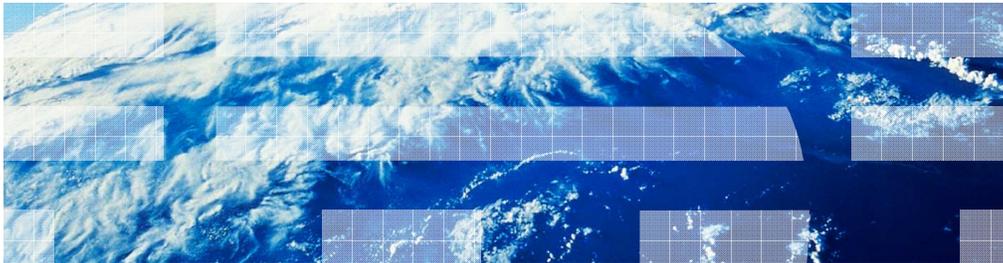


# WebSphere Business Modeler V7.0

## Modeling for asynchronous returns



This presentation provides an introduction to business process modeling with asynchronous returns.

## Goal

- Demonstrate the features in WebSphere® Business Modeler V7 that are available for modeling asynchronous communications in a business process



Call me back

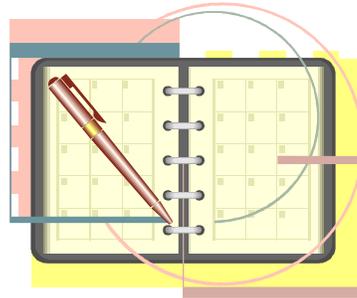
Modeling for execution is a very important feature of WebSphere Business Modeler. With each new version the gap between what can be modeled in WebSphere Business Modeler and what can be run in WebSphere Process Server is getting smaller.

WebSphere Business Modeler V7 addresses the gap by providing features for modeling asynchronous returns that can be transformed into the appropriate BPEL receive and receive choice constructs.

The goal of this presentation is to demonstrate these new features and illustrate how to use them.

## Agenda

- Asynchronous responses
- Simple receive
- Correlation keys
- Receive with call back
- Service interfaces with multiple operations



In this presentation you will learn about modeling your business processes so that they can receive responses from other applications asynchronously. A couple of common scenarios are covered. The 'simple receive' is where the return information is passed explicitly to the person or application that is responding back. And the 'receive with callback', is where the return information is embedded in the outbound message to a service.

In all cases, the concept of correlations is required. The details of defining and using the correlation keys will also be covered.

The final topic will discuss setting up and using a service interface that has multiple operations.

## What is an asynchronous return?

- Business processes do not always flow smoothly from one task to the next
- Sometimes a business process needs to be interrupted or needs to wait for information to be returned
  - Either of these events can happen at anytime
- Typically these tasks or activities, are completely outside the control of your business or business process.
- It might be that there is an external event that you need to anticipate
- Examples:
  - Cancellation of the all or part of the business process after it has been stated.
  - A status update
  - An update to key business parameters that require reworking forecasts
  - A time limit for a specific activity is exceeded
  - Receive customer response or feedback
- Modeled as a *Receive task*
  - It is like the mailbox
  - The *endpoint reference* is the address



Sometimes a business process needs to be interrupted or needs to wait for information to be returned.

Asynchronous means that they will contact you when they are ready, so don't stop doing what you need to do.

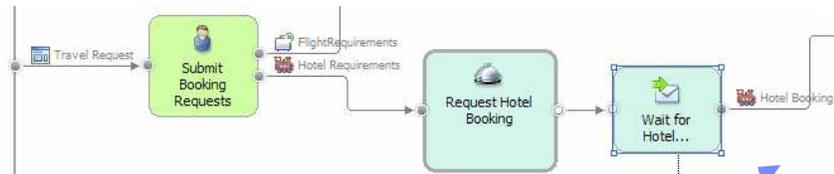
Because these responses are outside the control of your business process, you need to provide a way for them to contact you in their timeframe.

You need to provide them with an address and a mailbox. The mailbox is the "receive task" and the address is a combination of the endpoint reference and the correlation key.

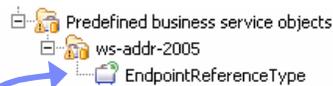
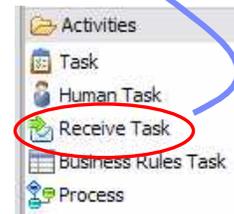
You also need to have multiple entry points into a business process. You can create multiple inputs at the beginning of the business process but this is not always the way it happens or needs to be modeled.

One way to think of the receive task is to think of it as an alternate entry point into the business process, that you can insert anywhere you need it to be.

## Receive task



- A new feature introduced with V7
- Allows for the case where the business process must wait for some external input...
  - after the process has been started
- Appropriate BPEL constructs are generated when the process is imported to WebSphere Integration Developer
  - If a Business Service with multiple operations is
    - associated with a receive
      - a receive choice is generated in WebSphere Integration Developer
- Two basic patterns
  - Without the callback information
  - With the callback information



5

Modeling for asynchronous returns

© 2010 IBM Corporation

As mentioned on the previous slide, the receive task will wait for some correlated input.

When this is used in a modeling for execution scenario, all of the appropriate BPEL constructs are created when imported into WebSphere Integration Developer.

If there are multiple operations associated with the business service interface you can create a single receive that uses all of the operations.

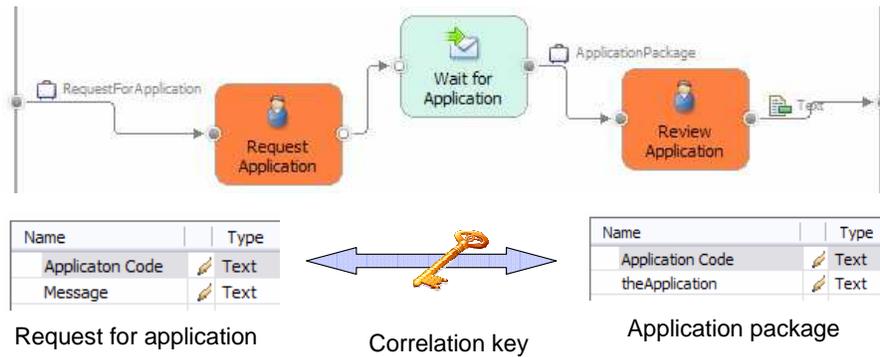
When imported into WebSphere Integration Developer, this is mapped to a BPEL 'receive choice' activity.

There are two basic patterns for working with the receive task.

One style is very simple. It assumes that the person responding to complete the business process, has access to the system and the necessary correlation information. There is no 'call back' information embedded in the outgoing message, if there is one. This is easy to use and test but requires inside knowledge of the business process.

The second style is more flexible but requires a little more work. With this style the callback information is embedded in the outgoing message. The receiver has everything they need to programmatically respond back to the business process. As you will see, there is a special 'endpoint reference' data type that is used to convey the callback information.

## Simple receive - without endpoint reference



- A request is sent out to a selected individual to submit an application for a position
- In this scenario there are two important points
  - 1. The end-user must be given the unique piece of information that is the correlation key
  - 2. The end-user must have access to the application

6

Modeling for asynchronous returns

© 2010 IBM Corporation

In this scenario the user-task “Request Application” represents a person that will send out an e-mail request to a specific individual. After the request has been sent out, the business process must wait for an indeterminate period of time for a response. This is what the receive task, “wait for application” does.

Instructions on how to respond are part of the request, it includes an application code.

The applicant will log into the system and use the application code when responding.

The application code is the correlation key that binds the incoming receive to the business process instance.

Because the endpoint reference, the return address to the process is not included in the message, you must log into a Business Space or use something like the Business Process Choreographer in order to interact with the business process.

## Establishing the runtime correlation

Forms

Correlation Keys

More

- Identify a combination of attributes that will uniquely identify the business process instance
  - One or more attributes from a business item
    - An ID or a contract number
  - Could be a combination of attributes from several business items
    - a customer number + a purchase number
- Set the correlation key
  - Identify where, in the business process to set the values for the correlation key
    - Business process
    - Business service
- Use the correlation key
  - Identify where in the business process the correlation key is used
    - This is the *receive task*
- Warning:
  - Do not confuse with the correlation specification used for simulations

### Correlation used by simulation

Set attributes used by data-driven simulation. These settings are not used to correlate process instances at run time.

Input criterion has correlation logic

7

Modeling for asynchronous returns

© 2010 IBM Corporation

Once you've identified the unique set of attributes to use for the correlation key, and the business items you'll use to get the values, you can then decide where and when to set them.

You set the values for the correlation properties at a point in the business process, before the receive task is reached. This can be done at the process level, using the attributes that are on the business items coming into the business process or at the business service level. If you choose to use a business service, then you can use either the input or the output business items.

However you choose to do it, you will select the item in the editor and then work with the settings in the 'Correlation Keys' tab of the element attributes.

Do not get confused by the correlations settings used by the simulation engine. You will see these settings in the 'input logic' tab of some elements.

**Correlation key management**

To correlate communication between process instances at run time, create and manage the required correlation keys for this process.

Name	Description
ApplicationKey	The correlation key used by the applicant when returning their applicaton.

Correlation keys for the process inputs

To add a correlation key for the process inputs, associate the correlation key with a process input criterion. Then, specify the IDs, or properties, whose unique runtime values will be used for inter-process communication.

Correlation key	Input criterion	Correlation ...	Type	Matching input
ApplicationKey	Input Criterion	AppCode	Text	Input.Applicaton Code

8 Modeling for asynchronous returns © 2010 IBM Corporation

The correlation keys are created and managed at the process level. Select the background of the business process diagram, then attributes, and then the 'Correlation Keys' tab.

The 'Correlation-key management' section is where you create the correlation key. A correlation key is a data structure that aggregates a set of discreet properties. The properties are simple types and you can have a many as you need. The goal is have a set of properties whose instance values will uniquely identify the business process instance. There is more on this on the next slide.

The section labeled, 'Correlation keys for the process inputs', is where you associate the key to the part of the business process that will set the values of the properties. In the example shown here, the key has a correlation property called 'AppCode' of type text that is set using the 'Application Code' attribute of the object being passed into the business process. The value is set as soon as the business process is started.

The other place where you can set the value of the correlation properties is in the 'Correlation Keys' tab of a business service. You will see an example of this in a few more slides.

## Creating the correlation key

**Define Correlation Key**

**Create or edit a correlation key**  
Define the IDs, or the correlation properties, for identifying a specific process instance at run time.

Name  
AccountServicesMainKey

Description  
This is used to reenter an operation after it has been initiated

Correlation properties

Name	Type	Description
AccountNumber	Text	The customer account number

Add  
Remove

OK Cancel

9

Modeling for asynchronous returns

© 2010 IBM Corporation

Selecting the add or edit buttons on the right side of the 'Correlation-key management' section will invoke the dialog shown here.

Provide a name for your correlation key that will help you remember how you intend to use it. Fill in the description so anyone reviewing your model is able to quickly figure out how you intend to use it.

Next, add one or more properties to compose your property set.

## Setting the correlation key values

### Correlation keys for the process inputs

To add a correlation key for the process inputs, associate the correlation key with a process input criterion. Then, specify the IDs, or properties, whose unique runtime values will be used for inter-process communication.

Correlation key	Input criterion	Correlation ...	Type	Matching input
ApplicationKey	Input Criterion	AppCode	Text	Input.Applicaton Code
<Select a correlation key>				
ApplicationKey				
<New Correlation Key...>				

Invokes a dialog to select from available inputs

- The add button will add a row to the table
- Edit the table columns directly
  - Starting from the left and moving to the right
- Select an existing key or create a new one
- Identify the input criterion
- Select the input properties to use for setting values in the key

**Specify Matching Input**

Select an input or input attribute

Click OK.

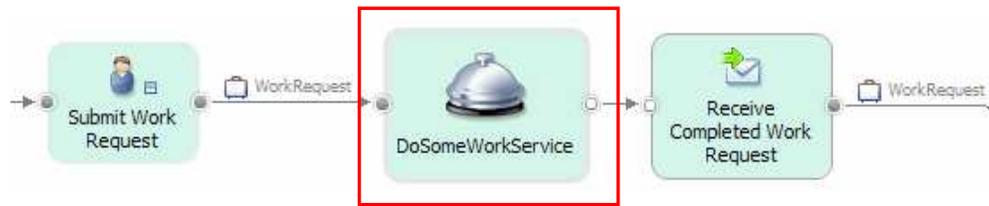
- Processes
  - Simple Receive
    - Input
      - Applicaton Code
      - Message

First select a correlation key. Then select the input criterion.

As you move from left to right your options are narrowed for you based on your previous selections.

When you get to the 'matching input' column, select the table cell to get the edit button. The edit button will then invoke a dialog that will present you with your available options.

## Setting the correlation from a service



Attribut Errors Business Servers Synchro Technic Process Consol Server L Properti

General Inputs Outputs Input Logic Output Logic Resources Correlation Keys More

To correlate communication between process instances at run time, associate a correlation key with a **service input or output criterion**. Then, specify the IDs, or properties, whose unique runtime values will be used.

Correlation key	Input or output crite...	Correlation prop...	Type	Matching input or output
WorkRequestKey	ServiceInputCriterion	WorkNumber	Text	aWorkRequestIn.WorkRequestNumber

Add Remove

11

Modeling for asynchronous returns

© 2010 IBM Corporation

The values for the correlation key properties can also be set from either the input or the output of a business service.

In this example, the service has no output so the input is all that is available.

## Using the correlation key

The diagram illustrates a business process flow with three tasks: 'Submit Work Request', 'DoSomeWorkService', and 'Receive Completed Work Request'. The 'Receive Completed Work Request' task is highlighted with a red box. Below the diagram is a screenshot of the IBM Business Process Modeler interface, showing the 'Correlation Keys' configuration for the 'Receive Completed Work Request' task. The interface includes a table with the following data:

Correlation key	Output criterion	Correlation prop...	Type	Matching output
WorkRequestKey	Output Criterion	WorkNumber	Text	Output.WorkRequestNumber

The interface also includes 'Add' and 'Remove' buttons and a footer with the text '12 Modeling for asynchronous returns © 2010 IBM Corporation'.

Once the correlation key has been defined and you've specified where and how to set the values, you now associate the specific properties of the key with specific properties in the output of the receive task.

The examples used here are simple keys with only one property. When using a multi-part key, you will need to make the association for each property in the key.

For this example, the business process calls out to the 'DoSomeWorkService' and then waits for the service to respond asynchronously. When the service calls back to the business process, it will use the value of the 'WorkRequestNumber' set earlier in the process as the correlation key. There might be several different business process instances waiting at this activity, therefore, the work request number is used to identify the specific instance. This value is used by the business process choreography engine to locate the specific business process instance that is waiting at the 'Receive Completed Work Request'.

## Receive with callback



- The *endpoint reference* is included in the business item
  - Used by the receiver to programmatically respond back to the business process
  - A special, predefined data type
- Must first import the predefined type
- Then add it into you business item
- It is automatically populated at runtime
- The service will need to know how to read and use the *endpoint reference*



#### Import Technical Types

##### Select a Predefined Technical Type

Imported technical types are added to a "Predefined business service objects" catalog in the project.

##### Predefined Types

- Endpoint reference type (ws-addr-2005)
- Case folder type

Name	Type	Min	Max
CallbackAddress	EndpointReferenceType	0	1
Address	AttributedURIType	1	1
ReferenceParameters	ReferenceParametersType	0	1
Metadata	MetadataType	0	1

13

Modeling for asynchronous returns

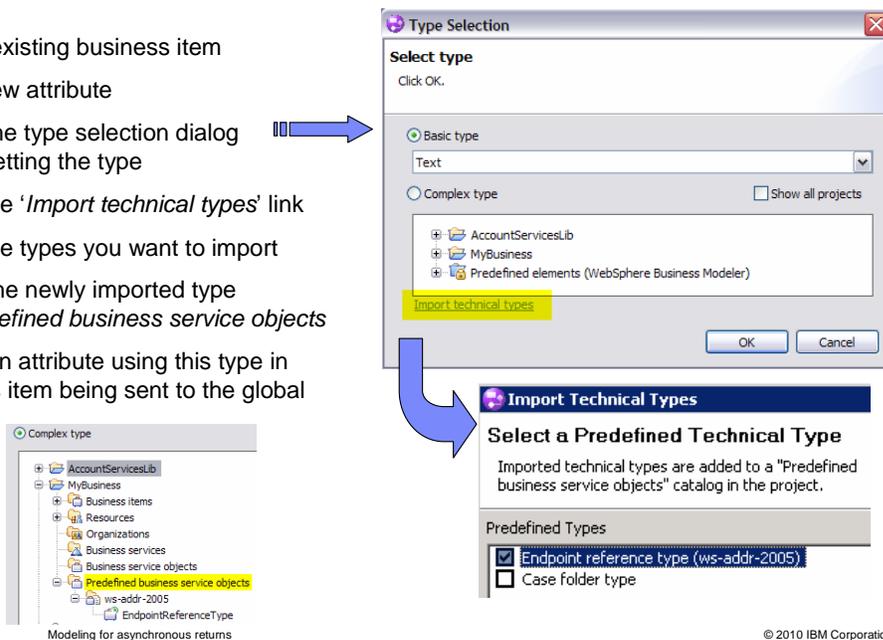
© 2010 IBM Corporation

The key distinction with this pattern is that the endpoint reference information is embedded in the business item being sent to the global service. In this example, it is added as an attribute to the WorkRequest business item.

Shown at the bottom right is the attribute named 'Callback Address' that is of the type EndpointReferenceType. You can name your attribute anything that has meaning for you. It should also give the programmer that is receiving this message a clue as to what it is used for.

## Importing the endpoint reference predefined type

- Use an existing business item
- Add a new attribute
- Invoke the type selection dialog – By setting the type
- Select the 'Import technical types' link
- Select the types you want to import
- Locate the newly imported type – *Predefined business service objects*
- Create an attribute using this type in business item being sent to the global service



14

Modeling for asynchronous returns

© 2010 IBM Corporation

The endpoint reference type is one of two new pre-defined data types that are available with WebSphere Business Modeler V7. To use the predefined types, you must first import them. The import option is available from the type selection dialog that is presented to you when you set the type for an attribute that you've created.

Create a new attribute and set the type. In the example, the attribute is called 'callback address.'

When you get to the type selection dialog, look at the bottom of the dialog. There you'll see the '**import technical types**' link, which is used to invoke the dialog shown in the lower right. Select the type you want. In this case it is the 'endpoint reference type.'

The selected types are imported into your project to the 'predefined business service objects' folder; shown on the bottom left.

From this point on, you use them as you do any other data type.

The forms that you create which are based on business items using the 'endpoint reference type' should not expose the fields that are part of this type. The values are populated automatically by the runtime when used in a message that is being sent to a business service.

## Global service and business service

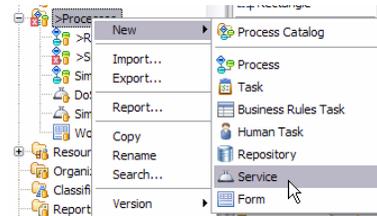
- Global service
  - Defined in WebSphere Business Modeler
  - Process → New → Service
  
- Business service
  - Imported
    - WSDL file
    - Business Service Search (WSRR)
  - Immutable



Global service



Business service



Cannot be used  
with endpoint  
reference

When working with services in WebSphere Business Modeler, there are two kinds of services. There are the *global services* that you as a business modeler design and create, and there are the *business services* that you import. You can easily tell the difference by the default icon that is used.

The hotel service bell is used to represent the global service and the brief case is used to represent the business service.

It is important to mention this because the business service cannot be used in the scenario we're about to discuss. Since the business service is immutable, there is no way to add the endpoint reference attribute to the business item.

## Using the endpoint reference

- Setup the correlation keys
  - Define the key within the process
  - Associate the key with the service calling out
    - This sets the values based on the input or the output
  - Associate the key with the *receive task*
    - This will use the values set by the caller
- Associate the receive with the calling service
  - Add the endpoint reference as an attribute to the business item of the service calling out
  - Specify the endpoint information in the technical attributes of the calling service



Use	Input criterion	Matching input	Receive task	Output criterion
<input checked="" type="checkbox"/>	ServiceInputCriterion	aWorkRequestIn.CallbackAddress	Receive Completed Work Request	Output Criterion
<input type="checkbox"/>				
<input type="checkbox"/>				

16

Modeling for asynchronous returns

© 2010 IBM Corporation

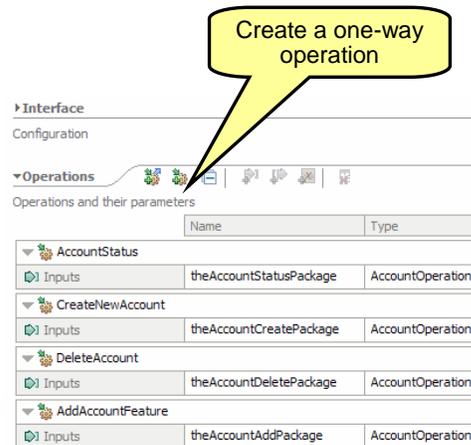
All of the steps except the last one have been covered in previous slides. This last step is easy to overlook because it is part of the technical attribute specification.

This completes what you need to setup in WebSphere Business Modeler for the receive with callback.

To complete the scenario you need to have a service implementation that can read the endpoint reference and send a reply.

## Business service definitions with multiple operations

- Sometimes a service definition will have more than one operation
  - This is done because the operations are closely linked and are managed and delivered as a package
- The WSDL will come from an external source and be imported into WebSphere Business Modeler
  - A business service
- When using this pattern, the operations must be one-way operations.
- Think of each service operation as an entry-point into your business process
- Generates a BPEL *receive choice* when imported into WebSphere Integration Developer



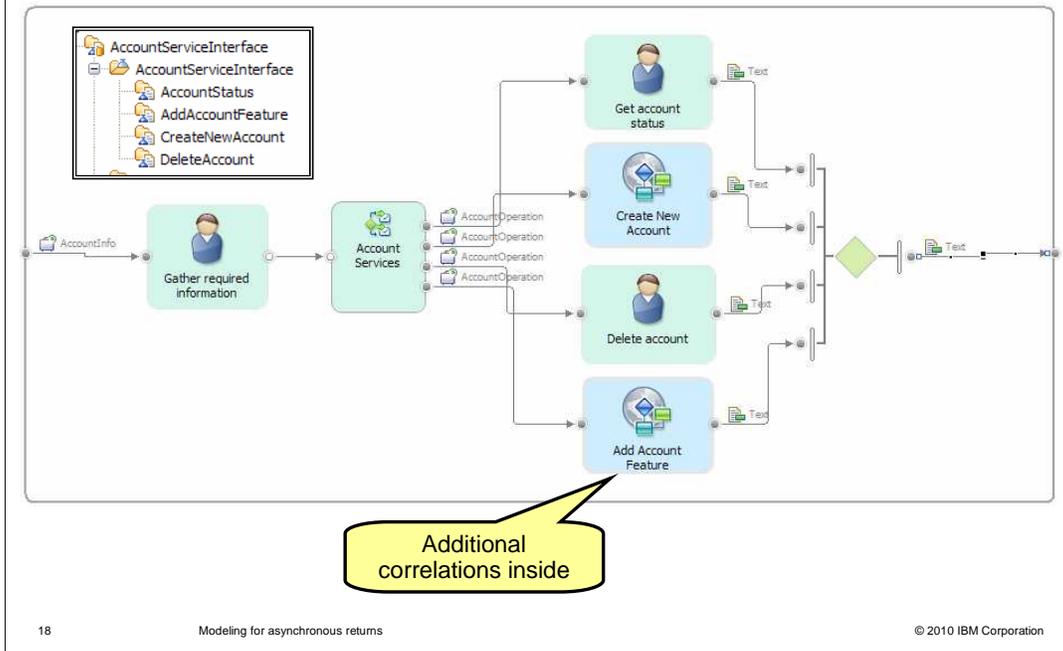
From the interface editor in WebSphere Integration Developer

For whatever reasons, you might find yourself working with a service interface definition that has more than one operation. When presented with this option you can choose to use each operation separately, as discussed earlier, or you can create a multiple receive task.

The screen capture here shows a service definition in WebSphere Integration Developer that has four one-way operations.

The next few slides will take you through an example of how you might use this feature.

## Account management example: a facade pattern



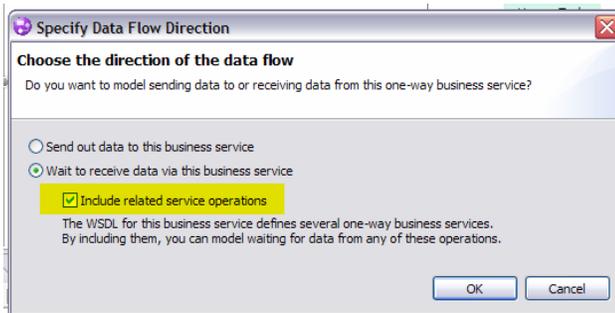
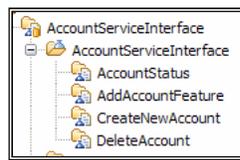
A façade is used to provide a unified interface to a set of interfaces in a subsystem. Facade defines a higher-level interface that makes a subsystem easier to use" [GoF 94]

At this level the business process is very simple. There is a single entry point to initiate the services for a given account. The account number in the account information business item is used to correlate the operations. In the context of each operation there might be additional correlations based on the account number and the operation type. This provides a way for the more complex processes such as create and add feature, to interact with customers or other business entities.

The mechanics of the correlation keys is the same as discussed previously. What is new and different is how to create the multiple receive task. Notice the icon is different from the standard receive task. You won't find this one on the palette.

## Multiple operations on a receive

- Drag the operation onto the canvas
  - The dialog shown below is invoked
- Selecting the check box in the dialog, will cause all of the operations in the service interface to be included
- This is transformed into a *Receive Choice* when imported into WebSphere Integration Developer



19

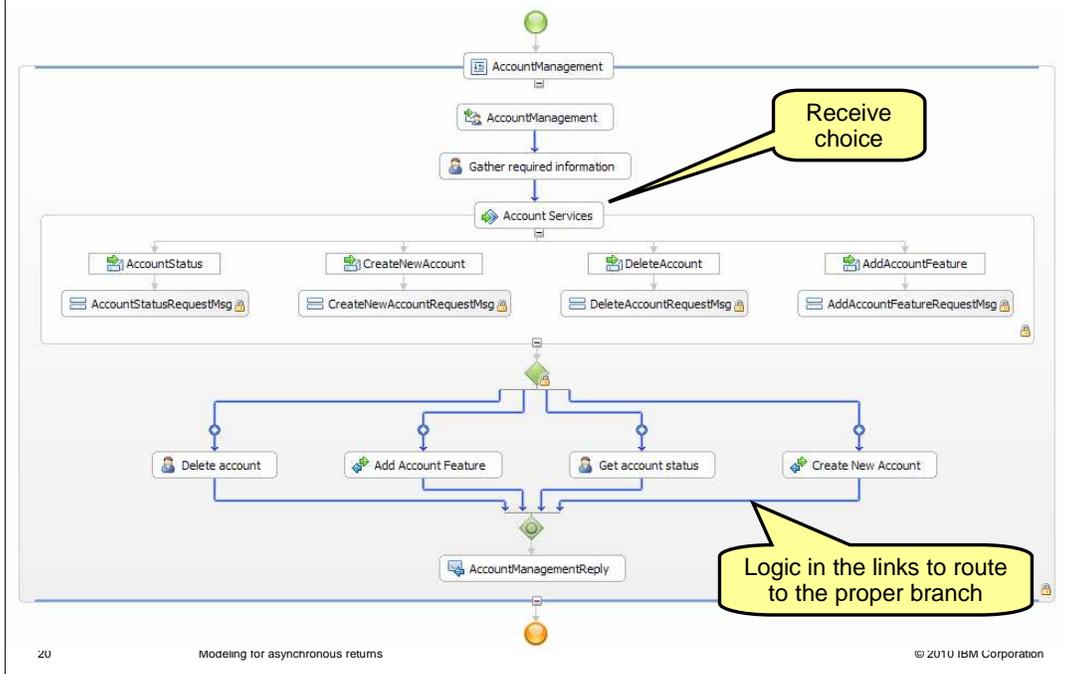
Modeling for asynchronous returns

© 2010 IBM Corporation

The multiple-receive task is only available through the drag-and-drop interface.

When you drag a business service operation that is part of a WSDL with multiple operations, the dialog shown here is invoked. If you select the second radio button and the associated check box, you will get the multiple-receive task. If you don't select the check box, but just the radio button, then you will get a regular receive task that is associated with a single service operation.

## BPEL receive choice



Here you can see how the business model with the multiple receive task is transformed into the appropriate BPEL constructs.

It is using the BPEL 'receive choice'.

## Summary

- Asynchronous responses
- Simple receive
- Correlation keys
- Receive with call back
- Service interfaces with multiple operations



Hello, how are you?  
Hello, how are you?

In this presentation you learned what an asynchronous response is, in the context of business process modeling. It is a way for another person or application to communicate back to your business process in their own time.

The details of using the feature, including how to setup and use correlations were discussed using three basic patterns; simple receive, receive with callback and the façade.



## 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\\_WBPMv70\\_Modeler\\_MidReceives.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WBPMv70_Modeler_MidReceives.ppt)

This module is also available in PDF format at: [..\\WBPMv70\\_Modeler\\_MidReceives.pdf](..\\WBPMv70_Modeler_MidReceives.pdf)

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



## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2010. All rights reserved.