

IBM WEBSHERE 6.1 – LAB EXERCISE

Model synchronization – A simple example

What this exercise is about	2
Lab requirements	2
What you should be able to do	2
Introduction	3
Exercise instructions	4
Part 1: Iteration 01	5
Part 2: Iteration 02.....	15
Part 3: Iteration 03.....	26
What you did in this exercise	37
Solution instructions	38

What this exercise is about

The objective of this lab is to provide the student with a simple and easy to understand introduction to the new features in WebSphere Business Modeler V6.1 and WebSphere Integration Developer V6.1 for keeping the business model and the implementation model in synch when using iterative development processes.

It is assumed that the student has a working knowledge of creating and editing business processes in both WebSphere Integration Developer and WebSphere Business Modeler.

Lab requirements

List of system and software required for the student to complete the lab.

- WebSphere Business Modeler V6.1
- WebSphere Integration Developer V6.1
- WebSphere Business Modeler V6.1 and WebSphere Integration Developer V6.1 are **not installed with shell sharing**.

When installing both products on the same machine using the new Installation Manager program, *shell sharing* is achieved by sharing the same installation package. This is the default. To install without shell sharing, put each product in a separate installation package.

Note: For the purpose of this exercise, the WebSphere Process Server test environment or runtime are not required. This exercise does not take you through the steps of deploying and testing the application.

What you should be able to do

At the end of this lab you should be able to:

- Begin with a baseline business model, iteration 01, export it to WebSphere Integration Developer V6.1, make changes in the implementation model, run the compare merge to generate a delta file and then analyze the implementation changes in WebSphere Business Modeler V6.1
- Understand and interpret the results of the WebSphere Integration Developer V6.1 synchronization wizard.
- Understand the best way to make use of the new model synchronization features.
- Develop iterative development processes and guidelines that will help your team use the new model synchronization features with minimal disruption to the overall development process.

Introduction

WebSphere Business Modeler V6.1 and WebSphere Integration Developer V6.1 provide new features for tracking and synchronizing changes made to the models during iterative development.

This lab exercise covers several iterations of the model and build cycle, capturing baselines, comparing and accepting new business model changes, implementing the model and feeding the implementation changes back to WebSphere Business Modeler. In WebSphere Business modeler, the changes made during the implementation phases will be analyzed to determine the impact on the business model.

Upon completion of this lab you will have an understanding of these key new features and will be able to plan and manage a project that requires traceability between the business models and the implementation models.

Note: This exercise is not intended to be an example of best practices. It is used to illustrate some of the pitfalls that can occur when using this feature.

Exercise instructions

The instructions in this exercise are for a WebSphere Business Modeler V6.1 installation that is **not shell sharing** with WebSphere Integration Developer V6.1.

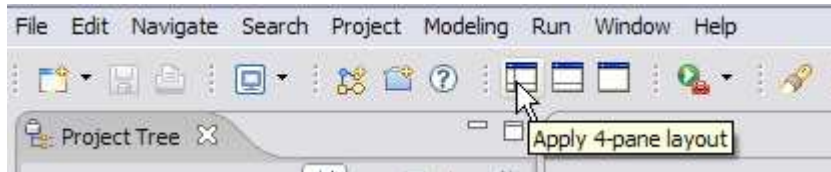
Some instructions in this lab are Windows® operating-system specific. If you plan on running the lab on an operating-system other than Windows, you will need to run the appropriate commands, and use appropriate files (.sh or .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references, as follows:

Reference variable	Windows location	AIX®/UNIX® location
<WAS_HOME>	C:\WebSphere60\AppServer	/usr/WebSphere60/AppServer /opt/WebSphere60/AppServer
<IRAD_HOME>	C:\Program Files\IBM\RS DP\6.0	
<LAB_FILES>	C:\Labfiles60	/tmp/Labfiles60
<TEMP>	C:\temp	/tmp

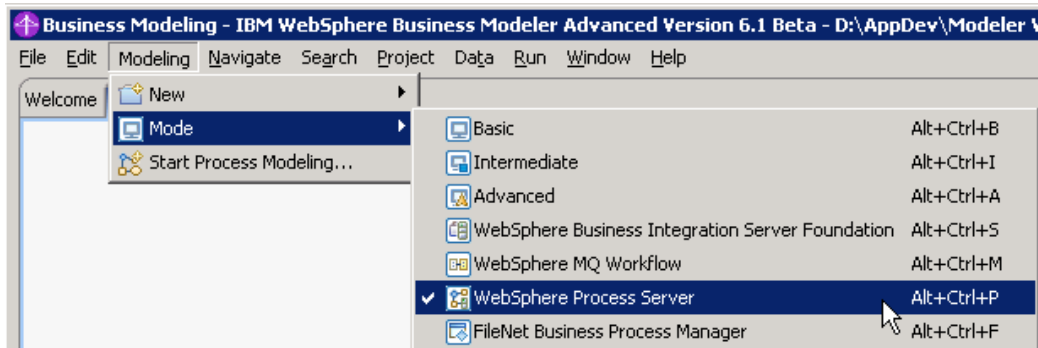
Note for Windows users: When directory locations are passed as parameters to a Java program such as EJBdeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, replace C:\LabFiles61\ with C:/LabFiles61/

Part 1: Iteration 01

- ___ 1. Download and extract the WBIV61_Labfiles_Modeler.zip from the IBM Education Assistant.
 - ___ a. It is under the additional resources where you downloaded this document from.
 - ___ b. Unzip it to the root of your drive.
- ___ 2. Start WebSphere Business Modeler V6.1 using a new workspace
 - ___ a. Call the workspace Simple.
 - ___ b. Wait a few moments until the WebSphere Business Modeler opens.
 - ___ c. Close the welcome page
 - ___ d. Apply the 4-pane layout from the toolbar



- ___ 3. Load and review the baseline business model
 - ___ a. Set the modeling mode to *WebSphere Process Server*

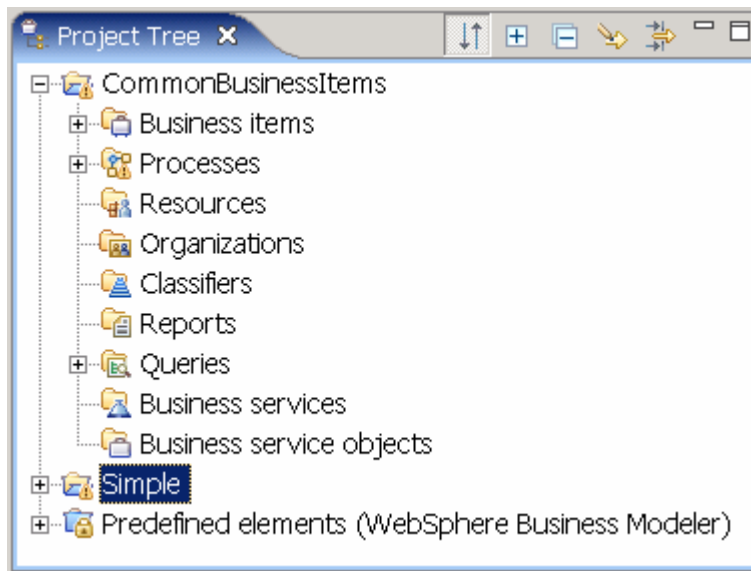


- ___ b. Import the Simple baseline model. This is a 6.1 model archive that has the project structure information in it; therefore you do not need to create the projects ahead of time.
 - 1) \\Labfiles61\WBModeler 6.1\Simple.baseline.mar
- ___ c. Explore the model. The features of interest are:
 - 1) The project called CommonBusinessItems
 - a) The purpose of this project is to provide a common library of the business items, taking advantage of the new 'cross project referencing' feature.

Possible best practice: to create a library of common business items that will also be shared with WebSphere Integration Developer V6.1 as common business objects, create an XML schema and import the data entities as Business Service Objects.

This provides rigorous control over the data definitions which means that the business analyst in modeler cannot create new data elements ad hoc.

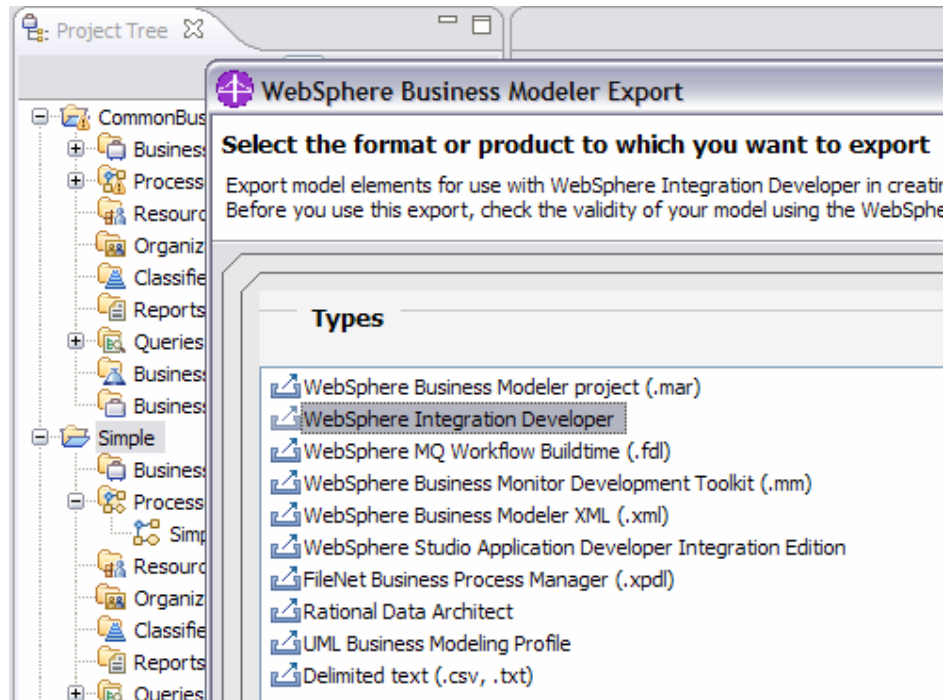
- b) Expand the Common Business Items and take a look at them. The *Application* business item is used as the input and output for all the tasks.
- c) Expand the Processes folder and you'll see that there is an empty business process called UtilityP1. This was created by mistake and will be fixed in subsequent iterations



- 2) Expand the **Simple** project and then the processes folder.
- 3) Open the business process called **SimpleProcess1** by double clicking on it.
 - a) This is a very simple example with two tasks, **Do It** and **Log It**
 - b) Notice that the technical attributes for the tasks and the Simple business process have been fully specified per the guidelines for XYZ Corp.

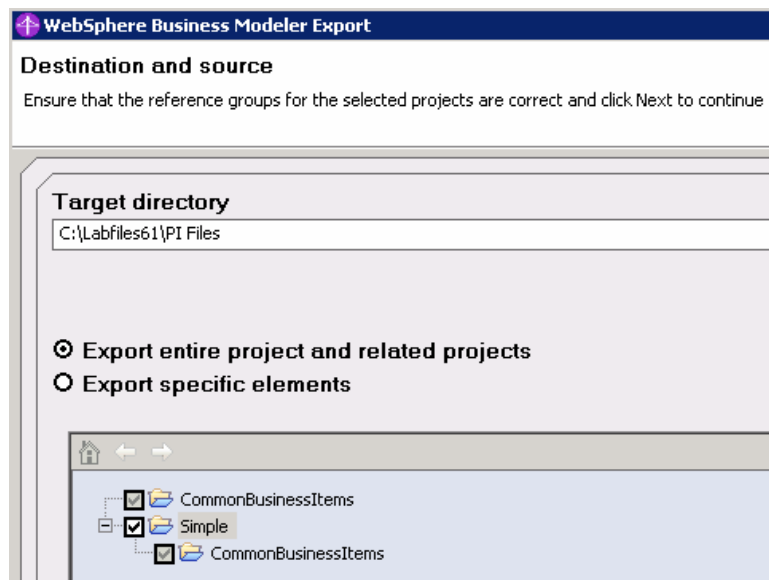
___ 4. Export the Simple business process for WebSphere Integration Developer.

- ___ a. Right mouse click on the Simple project and select **Export** from the pop-up menu. Select **'WebSphere Integration Developer'** from the **WebSphere Business Modeler Export** dialog



___ b. Click **Next**

___ c. Set the target directory and export the entire project



___ d. Use the **recommended export option**. This is selected by default. When this option is used, three projects are created for you in WebSphere Integration Developer V6.1 based on the separation of concerns principle discussed in the lecture.

Name the file **Simple.Modeler.I01** and append the timestamp. Naming the iterations consistently is very important. The convention used here indicates that this Project Interchange file comes from WebSphere Business Modeler and is the first iteration

WebSphere Business Modeler Export

WebSphere Integration Developer export details

Specify export details, then click Finish to export. Use the recommended export option for interoperability between WebSphere Business Modeler and WebSphere Integration Developer.

Select the export option

Recommended Export Option

This option creates three related projects. It separates the business logic from the implementation details. Choose this option for best interoperability when you are working iteratively between WebSphere Business Modeler and WebSphere Integration Developer.

Export using the standard project interchange format for other environments

Project Interchange Name

Simple.Modeler.I01

Append timestamp to project interchange name

Target Project Names

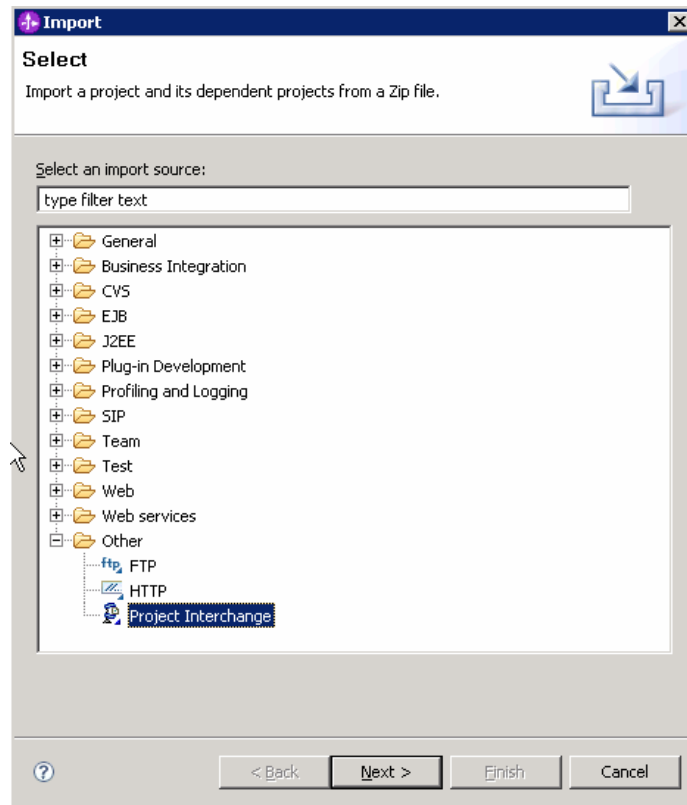
Modeler Project Name	Business Logic Module Name	Implementation Module Name	Library Name
CommonBusinessItems	CommonBusinessItems	CommonBusinessItems_impl	CommonBusinessItems_lib
Simple	Simple	Simple_impl	Simple_lib

___ e. Select the Finish button.

___ 5. Minimize or close WebSphere Business Modeler V6.1 and open WebSphere Integration Developer V6.1 on a new workspace (for WebSphere Integration Developer), call it **SimpleImpl**

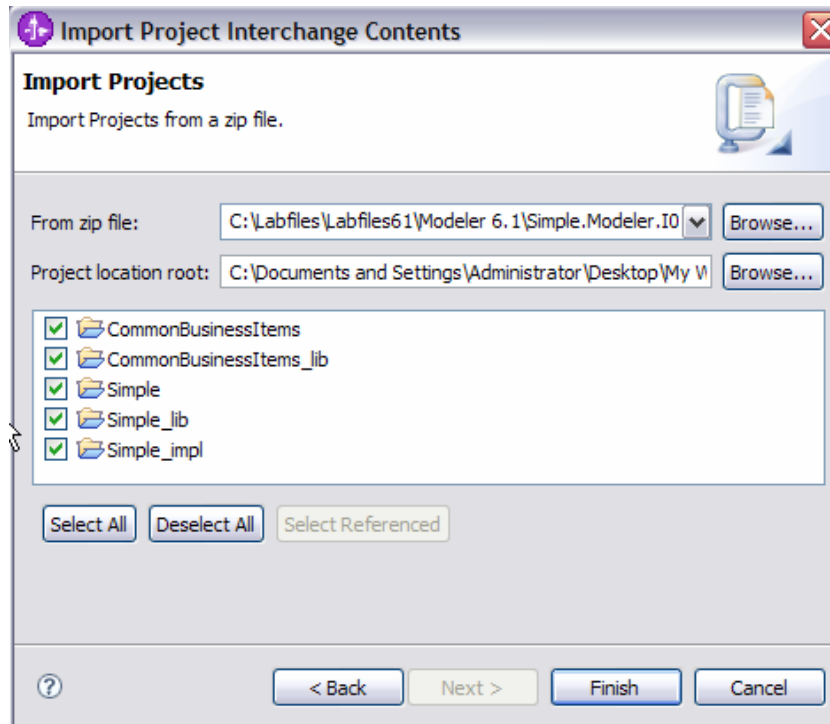
___ 6. Import the baseline model, I01 into WebSphere Integration Developer V6.1 **SimpleImpl** workspace.

___ a. Import the Simple.Modeler.I01 project interchange file into WebSphere Integration Developer

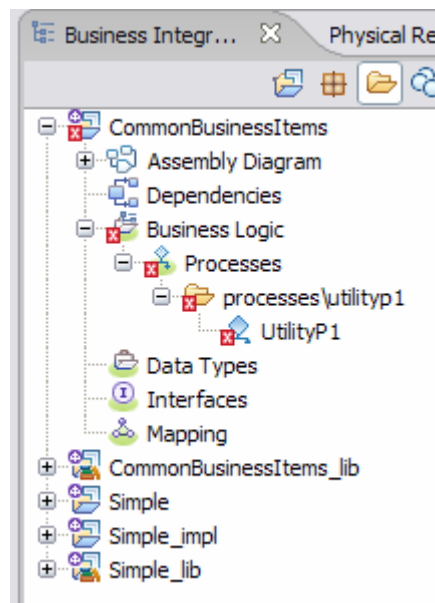


___ b. Select all of the projects

Notice all of the projects (modules) that are imported. This is because you used the option to export using the 'recommended export option', which creates the business integration module, implementation module and the library module. There is no CommonBusinessItems_impl module because there was no content to the UtilityP1 business process, nothing to implement.

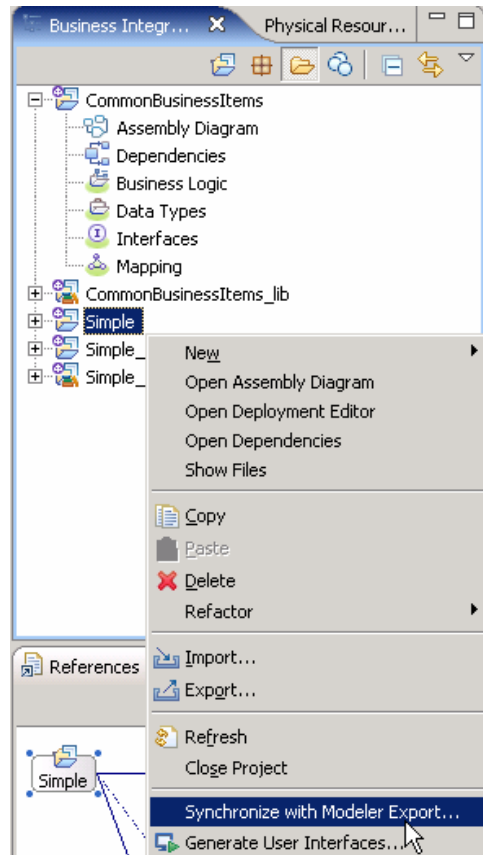


__ c. Next, notice the errors in the CommonBusinessItems module. Expand this module and you'll see the error is related to the incomplete UtilityP1 business process. No surprise here.

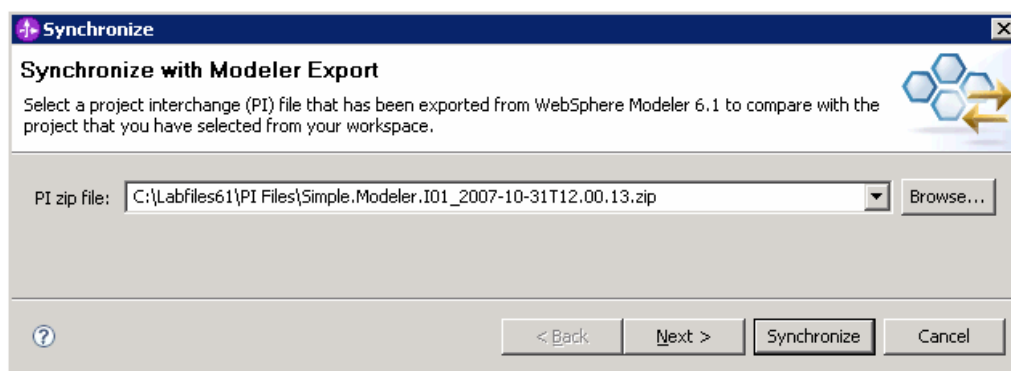


Build Activities Properties Problems Servers			
2 errors, 0 warnings, 0 infos			
Description	Resource	Path	Location
Errors (2 items)			
The parallel activities activity 'UtilityP1_Flow' contain	UtilityP1.bpel	CommonBusinessItems/Proc...	UtilityP1_...
The process is not startable. No receive choice or re	UtilityP1.bpel	CommonBusinessItems/Proc...	UtilityP1

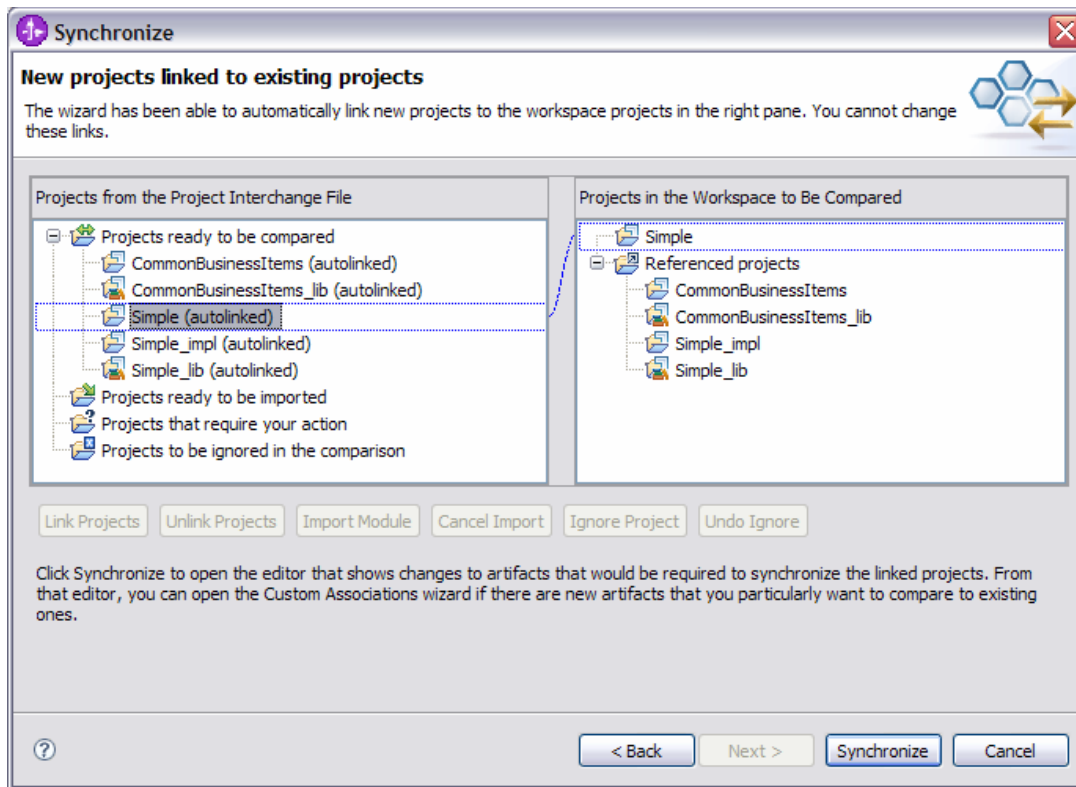
- ___ 7. Keeping it simple, the first change is to correct this mistake by the business analyst.
 - ___ a. Delete the UtilityP1 business process and the associated entries in the assembly diagram for the CommonBusinessItems module. There should be no errors in the problem list after the deletions.
 - ___ b. The next step is to synchronize the implementation model with the business model.
 - 1) Right mouse click on the Simple project and select the **Synchronize with Modeler Export** option



- 2) Set the source directory and file, and then select the Next button. The input file is the **I01** file that was used for the initial import. You'll be comparing the changes you just made in WebSphere Integration Developer to the baseline.

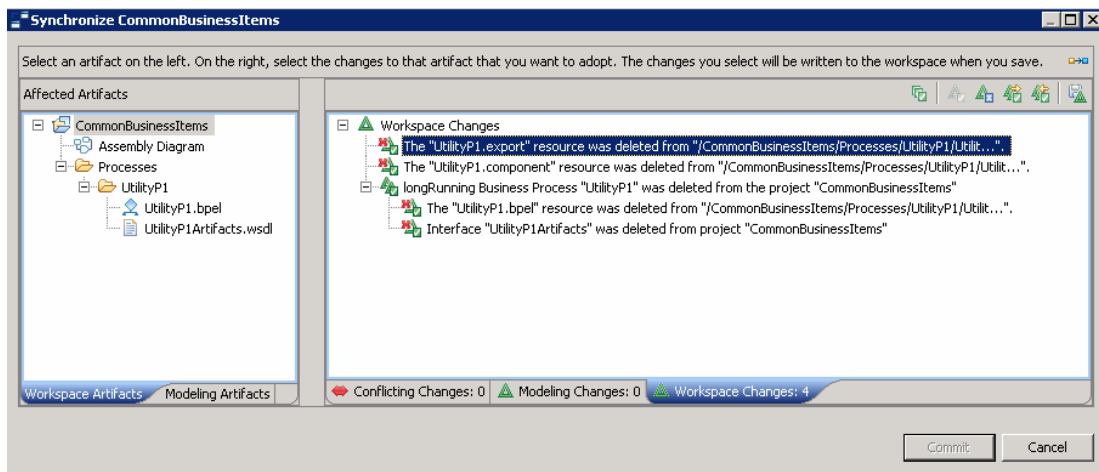


- 3) The panel below shows the projects from the project interchange file and matches them up with the projects in the current workspace. Notice that they're all auto linked. Selecting one on the left will show you which project it matches up to in your workspace



- 4) Select the **Synchronize** button.

- 5) This takes you to the main part of the Synchronization wizard

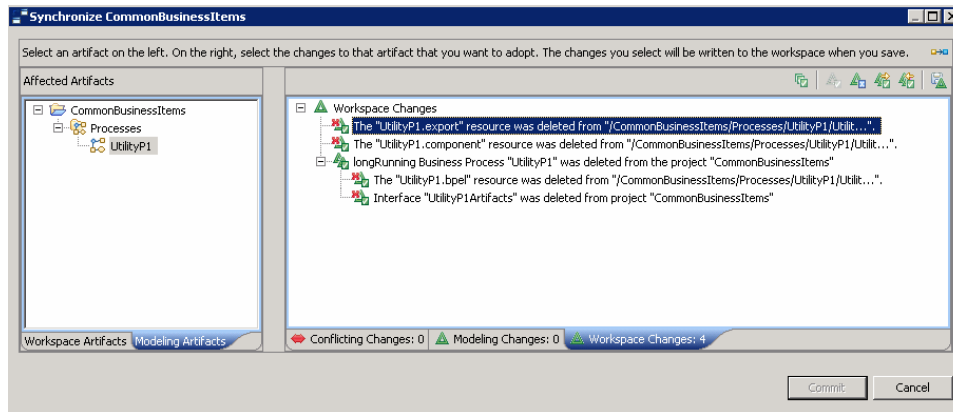


- c. By default the tree on the right is not fully expanded. To understand the change or changes at a high level, only the top level is necessary. In this case there was only one change and it was the deletion of the UtilityP1 process, which is a change to the workspace.

- 1) Notice the tabs at the bottom. You're looking at the workspace. There are no new changes in the model (from the project interchange file) and overall there were four changes in the

workspace that are the result of deleting this process and the associated artifacts in the assembly diagram.

- 2) Select the Modeling Artifacts tab at the bottom of the left pane, the Affected Artifacts pane. This provides a view from the perspective of the business modeler.



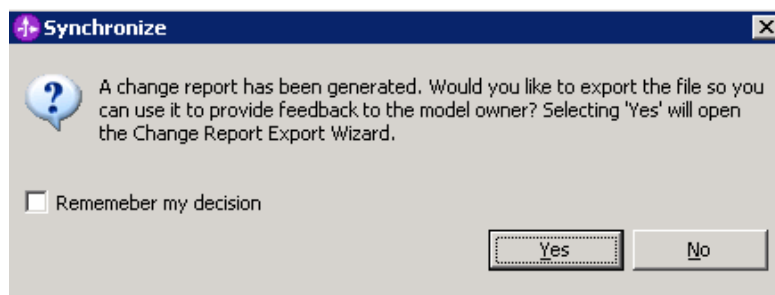
- 3) When you have a more complicated example you'll see that you can select the elements on the left and it will show you the associated changes on the right.
- 4) Notice the icons; the little green check mark indicates that the change has been accepted. Changes can be accepted or rejected. They are automatically accepted because they are changes that have already been done in the workspace and it is a fair assumption that they will be accepted at this stage.

The red X on the upper left of the icon means that this is a deletion from the workspace.

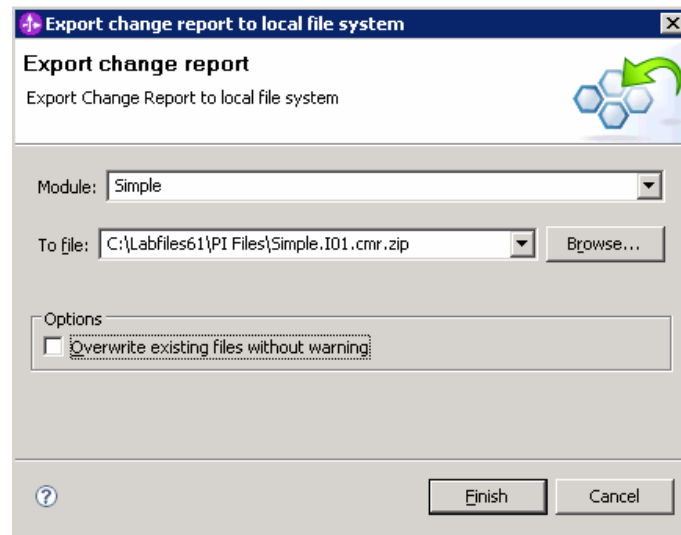
Study the icons carefully and make sure that they reflect your intent. In this case they are saying, "Accept the deletion of the artifact from the workspace".

- 5) It is now time to create the delta file so you can provide the business analyst with your change.

- a) When the changes are committed, you'll be provided an opportunity to create and save a file that has all the changes. This file can then be given to the business analyst and imported into WebSphere Business Modeler V6.1 where the changes can be reviewed and analyzed, and implemented if appropriate.
- b) Select the **Commit** button.
- c) Select the **Yes** button; you want to export the Change Report



- d) Enter the fully qualified path and file name. Notice the use of the naming convention, I01 for the first iteration from WebSphere Integration Developer

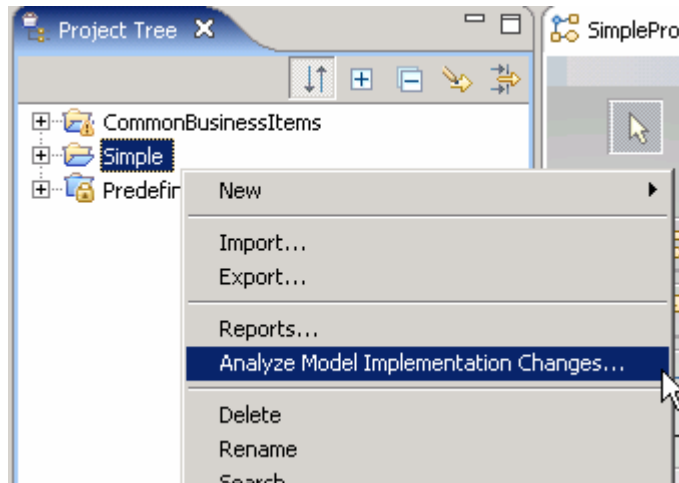


Warning: if you run the Synchronization wizard a second time, after doing the commit, all of the changes will show up on the modeler side. This can become confusing so it is best to wait until you get a new project interchange file from the business analyst before running the wizard again.

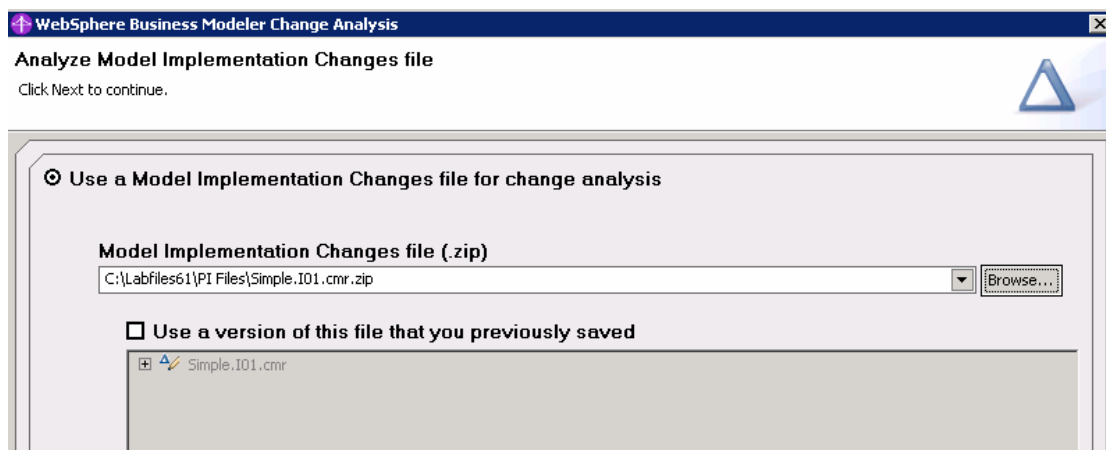
- ___ 8. This concludes the first iteration of the simple exercise. You can minimize the WebSphere Business Integration Developer program for now.

Part 2: Iteration 02

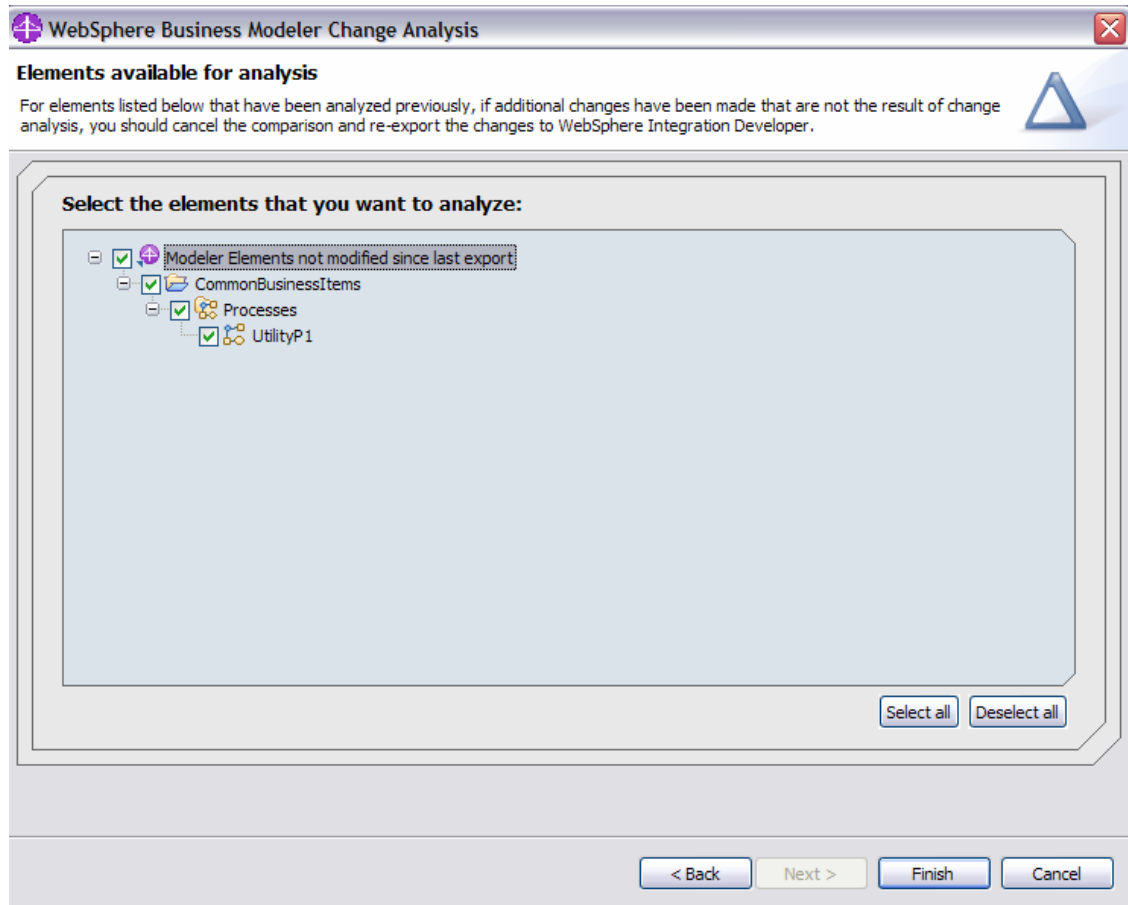
- ___ 9. Bring up WebSphere Business Modeler V6.1 using the Simple workspace.
- ___ 10. Analyze the changes made by the Integration Developer during the first iteration.
 - ___ a. Right mouse click on the Simple project and select **Analyze Model Implementation Changes**



- ___ b. Locate the file and select the **Next** button



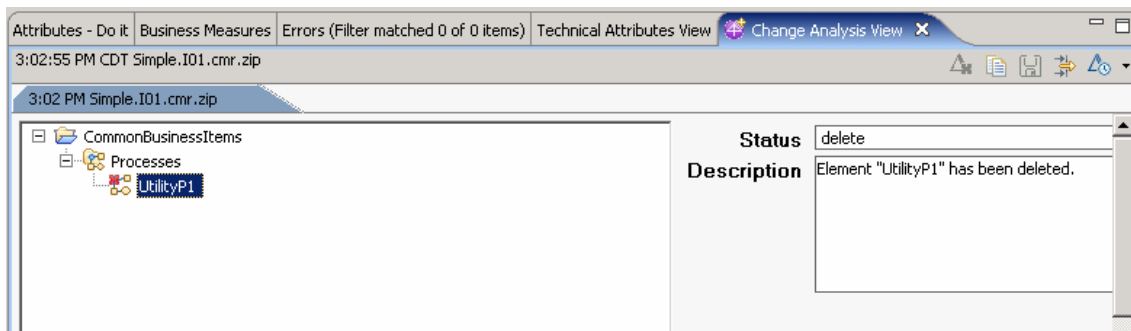
- ___ c. Expand the elements that can be analyzed and select the UtilityP1 process.



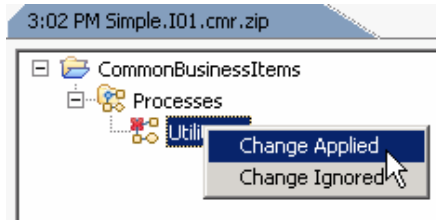
- ___ d. Select the **Finish** button.

- ___ e. Expanding the CommonBusinessItems tree in the *Change Analysis View*, you'll see that the deletion of the UtilityP1 process has been detected. Selecting the UtilityP1 process will show the description on the right.

No changes are automatically made in the modeler. This is strictly informational and it is up to the business analyst to incorporate the changes as they deem appropriate



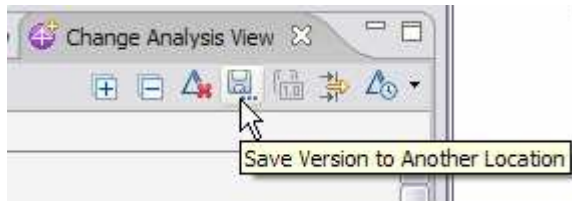
Once a change has been made, the element can be marked as "Change Applied"



___ f. Make the change in the model by deleting the UtilityP1 process and then mark the change as applied



___ g. Save the record of your changes using the save version button.



___ 11. It is now time to carry on with the development of the business model by adding a new task to the Simple Business process.

___ a. Add a new local task to the SimpleProcess1 business process.

1) Name it **New Task**.

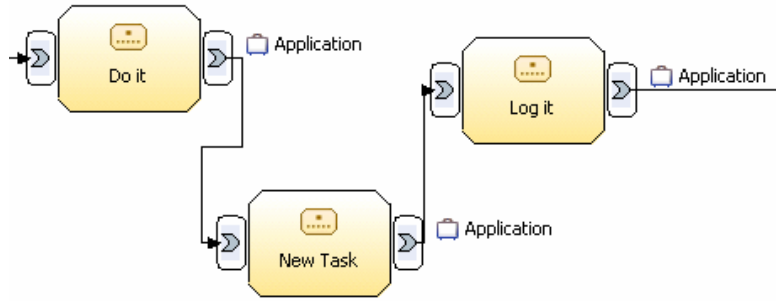


___ b. Insert it between Do It and Log It.

Note: When inserting a new task between two existing tasks, it is very important to move the links in a particular order. If you do not do it this way then the input information on the original task, Log It in this case, will get lost.

1) Move the link from the output of **Do It** to the output of **New Task**.

2) Next connect the output of the **Do It** to the input of **New Task**



3) Making the connections in the manner will preserve the input information on Log It and the Application data being passed from Do It. If this information is lost, then it will show up in WebSphere Integration Developer as changes in the model. You want the addition of the new task to show up but it is not the intention to change any part of the LogIt task.

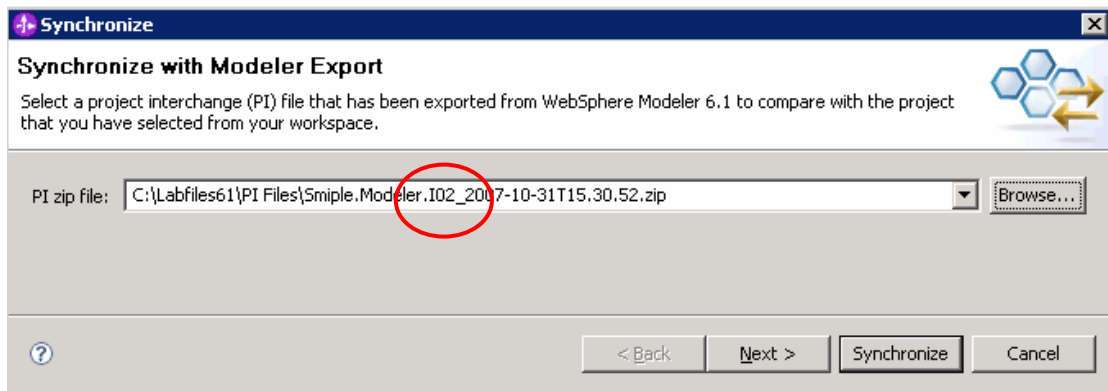
4) Save the changes using **Ctrl-s**.

- ___ 12. Export the new model as a project interchange file for the Integration Developer.
 - ___ a. File name: **Smiple.Modeler.I02**

- ___ 13. Put on your Integration Developer hat and open WebSphere Integration Developer V6.1 on the **Simple Impl** workspace.

Note: The model synchronization information is associated with a given workspace. You should use the same workspace when synchronizing with modeler. If you change to a new workspace at this point, your change history will be lost.

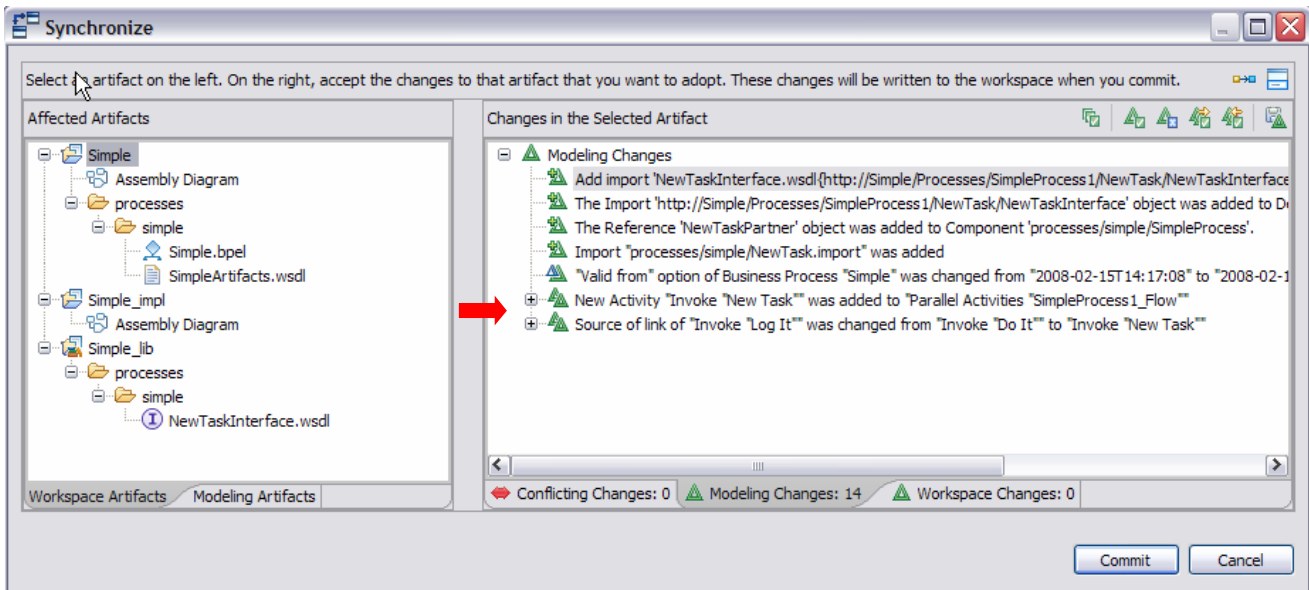
- ___ 14. Right mouse click on the Simple module and select, “Synchronize with Modeler Export”.
 - ___ a. Select the I02 file that you just exported from Modeler.



Select the **Synchronize** button.

- ___ b. This time all of the changes are on the modeler side and none of them have been accepted by default.

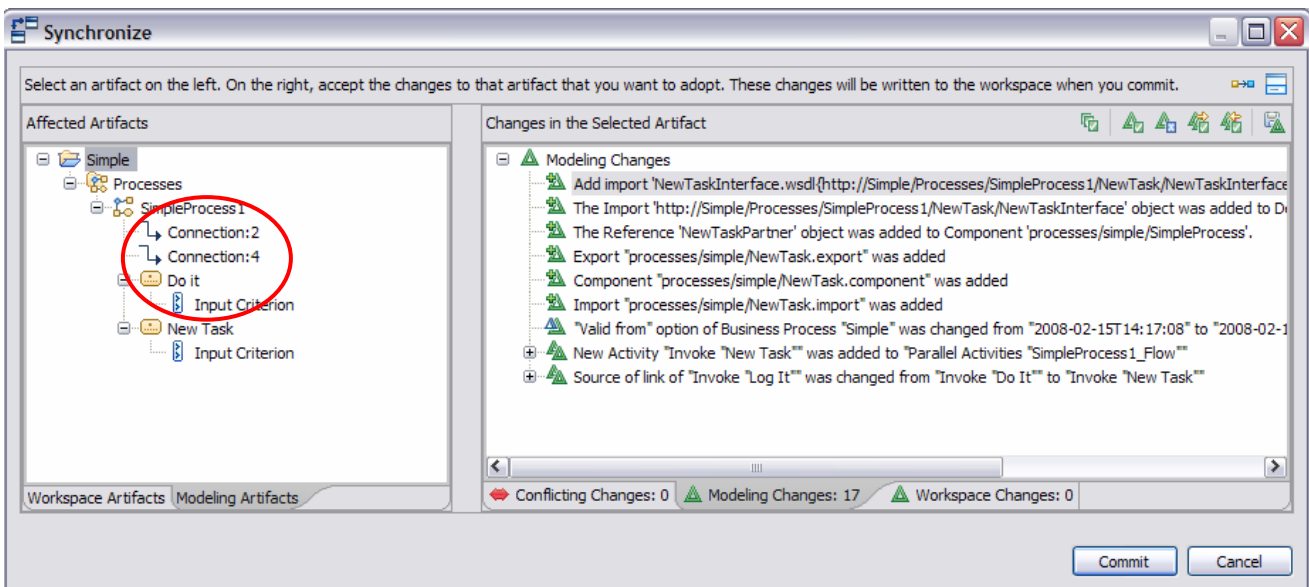
Inspect all the changes to make sure you understand them.



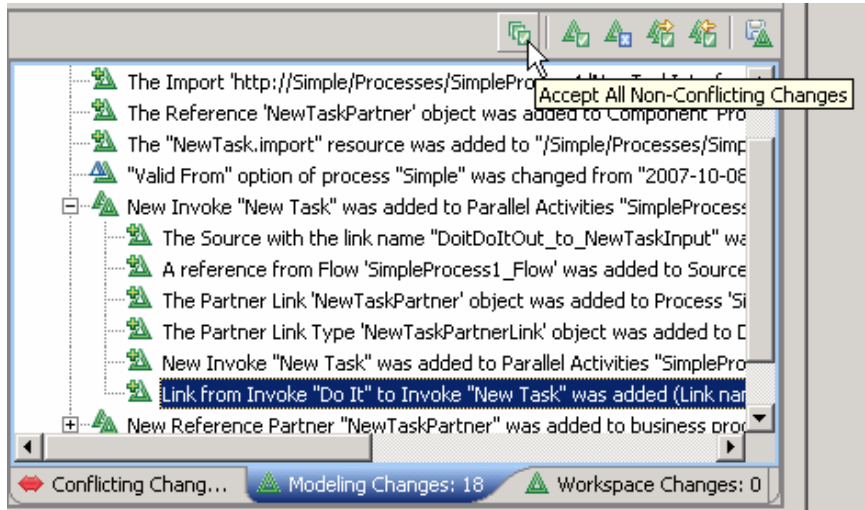
To inspect the changes, select Simple in the *Affected Artifacts* pane on the left and then study the associated *Modeling Changes* in the right pane. To get a quick overview, select each project, Simple, Simple_impl and Simple_lib and see the changes on the right.

With the Simple project selected, notice that one of the changes was the addition of a New Invoke activity called New Task. If you expand that, you'll see all the elemental changes in the workspace that changed in order to achieve this result.

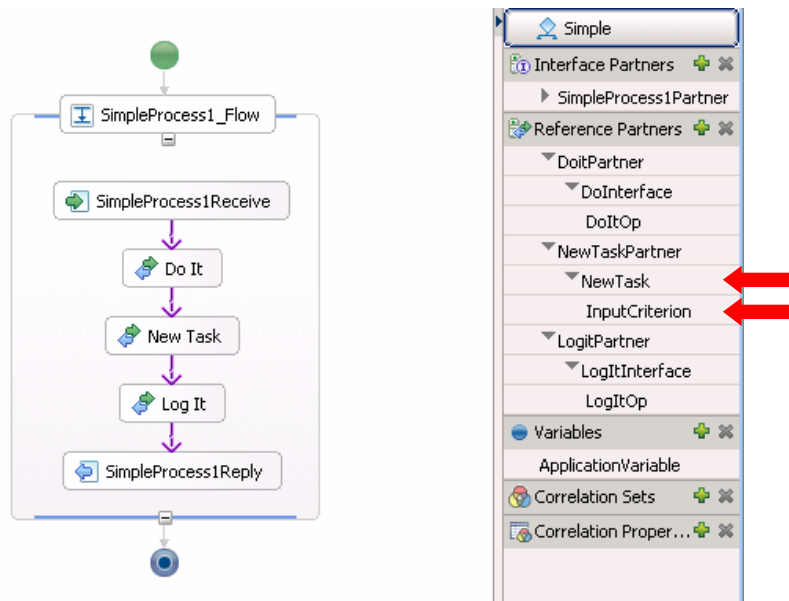
- ___ c. Since in this case the changes are coming from WebSphere Business Modeler, you might be interested in understanding how a given change in the business model affects the implementation model. To see this, switch to the *Modeling Artifacts* tab at the bottom of the *Affected Artifacts* pane and browse each element in the tree and study the related changes in *Modeling Changes* view on the right



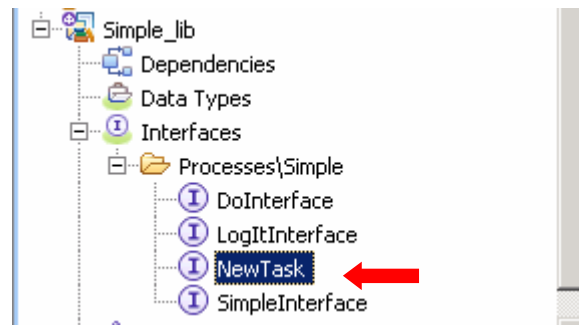
- ___ d. Now that you've had a chance to study and understand all the changes, accept them all using the *Accept All Non-Conflicting Changes* button in the toolbar.



- ___ e. When you commit incoming changes you are resetting the baseline.
 - 1) Notice that there was no reference to the UtilityP1 process that had been deleted. This is because it was also deleted from the business model in Modeler and therefore did not show up as a delta.
 - 2) When you did the commit, you did not get a chance to create the delta file because at this point both models are now in synch. If you run the synchronization wizard again, you'll see that there are no changes.
- ___ 15. Go to the Simple BPEL process in the Simple module and review the changes that you accepted. Right away you notice there are some things amiss. It quickly becomes apparent that in their haste to create the New Task that the business analyst forgot to fill out the technical attributes before they sent the project interchange file to you.
 - ___ a. The name of the interface and operation for the New Task business partner are not correct



__ b. In the Simple_lib below, it should be **NewTask** Interface



__ c. In the interface properties below, the namespace should be,

http://Simple/NewTaskInterface

Property	Value
Artifact	
Name	NewTask
Namespace	http://Simple/Processes/SimpleProcess1/NewTaskInterface
Primary File	{Simple_lib}\Processes\Simple\NewTaskInterface.wsdl
Type	Interface

__ d. Making the changes here in WebSphere Integration Developer would be problematic. Since this is essentially a top down process, changes in the model should be made by the business analyst in Modeler. Making the changes in the business model will also avoid the synchronization and integration headaches that may occur if there are subtle differences introduced by the Integration Developer.

An e-mail has been sent to the Business Analyst informing them of the omission. The correction will be picked up on the next iteration.

Possible best practice: The model is the responsibility of the business analyst, and arguably the *technical attributes* are in the gray area between the business analyst and the integration developer. For consistency and traceability use the technical attributes in the WebSphere Business Modeler for managing the names and namespaces. This will typically mean that someone on the integration development team will need to be familiar with WebSphere Business Modeler and work closely with the business analyst to ensure that the technical attributes are correctly specified.

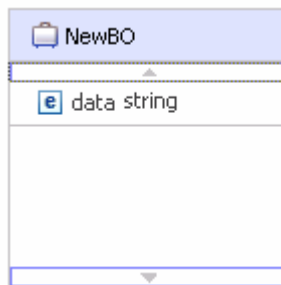
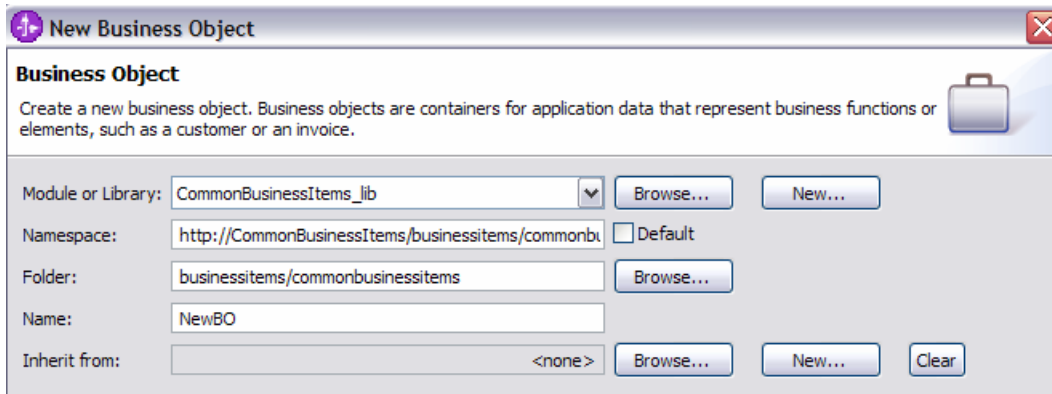
- ___ 16. Meanwhile the Integration Developer realizes that there is a need for a new business object (using implementation terminology) for the business process. It is something that is needed now in order to continue working with the current iteration so they create the new business object called **NewBO**.
- ___ a. In the CommonBusinessItems_lib module, create a new business object of type String in the BusinessItems\CommonBusinessItems folder. (Right mouse click on the folder and select *New* → *Business Object*).

Before proceeding, notice how the namespace is specified for the other business objects that are already there, the ones generated from the business item definitions in WebSphere Business Modeler. If the namespace is not the same, then when the business analyst adds the NewBO as a business item in the model, there will be refactoring issues when the model is synchronized for the next iteration.

- ___ b. Namespaces for business objects.

default used by WebSphere Integration Developer:
http://CommonBusinessItems_lib/BusinessItems/CommonBusinessItems
 default used by WebSphere Business Modeler:
<http://CommonBusinessItems/BusinessItems/CommonBusinessItems>

The default namespace used by Modeler cannot be changed, but the one used by WebSphere Integration Developer can.



- ___ c. The way the business object definitions are packaged into files is another area where the defaults between Modeler and WebSphere Integration Developer diverge.

Modeler will package all of the business objects into a single file while WebSphere Integration Developer puts each business object in a separate file.

default used by WebSphere Integration Developer: ...\\CommonBusinessItems\\NewBO.xsd
default used by Modeler: ...\\CommonBusinessItems\\CommonBusinessItems.xsd

This can be easily managed using the available tools.

- ___ d. (Optional) if you like you can now create the implementation for the NewTask in the Simple_impl module using a Java or BPEL implementation, your choice.

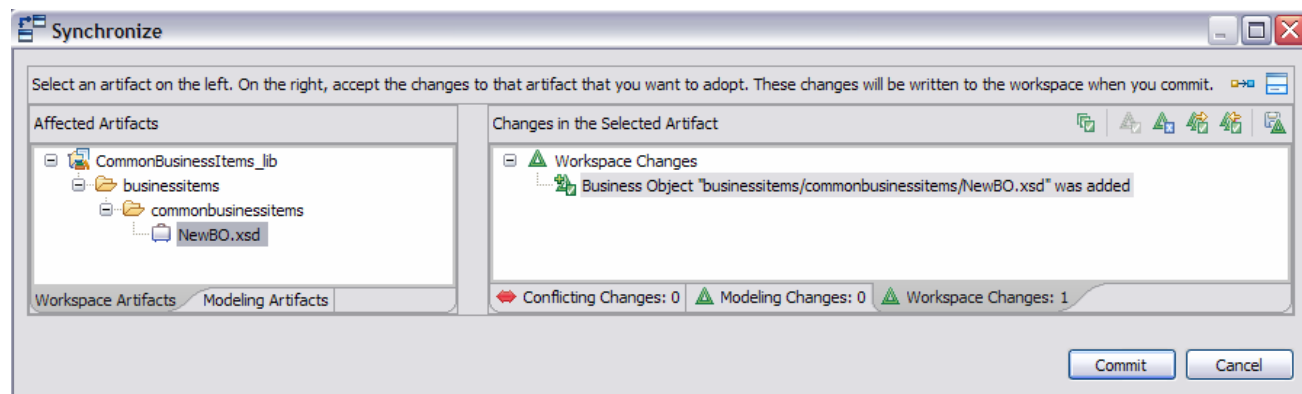
The changes you make in the **_impl** module are not reflected back to the business analyst because they are purely implementation details. They are beyond the scope of this exercise.

___ 17. Synchronize the models again.

- ___ a. Synchronize with the same modeler file that you used previously, I02

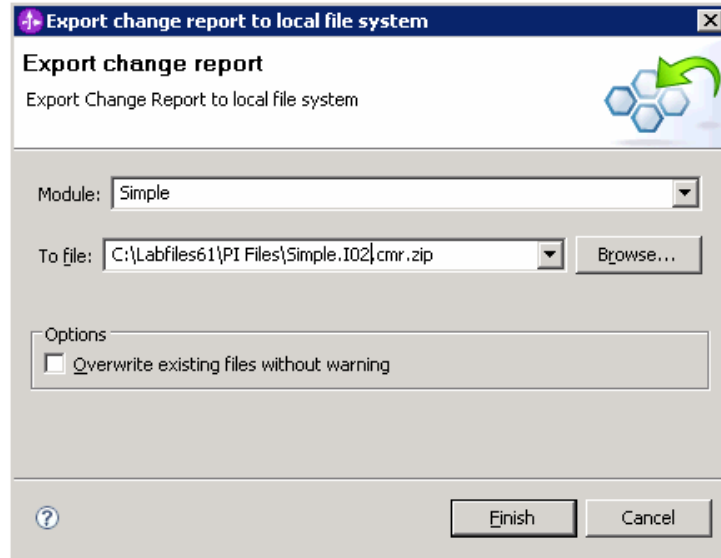
1) Smiple.Modeler.I02_2007-mm-ddThh.mm.ss.zip

- ___ b. The only change you should see is the NewBO that was added. It's automatically accepted by default.

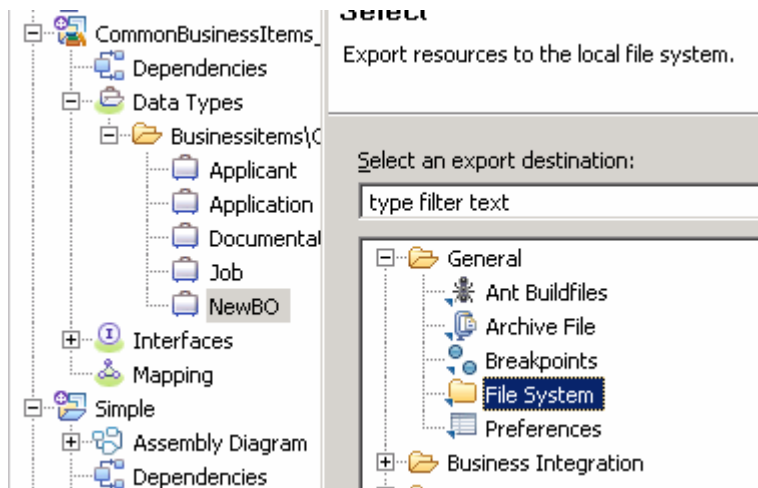


- ___ c. Select the **Commit** button.

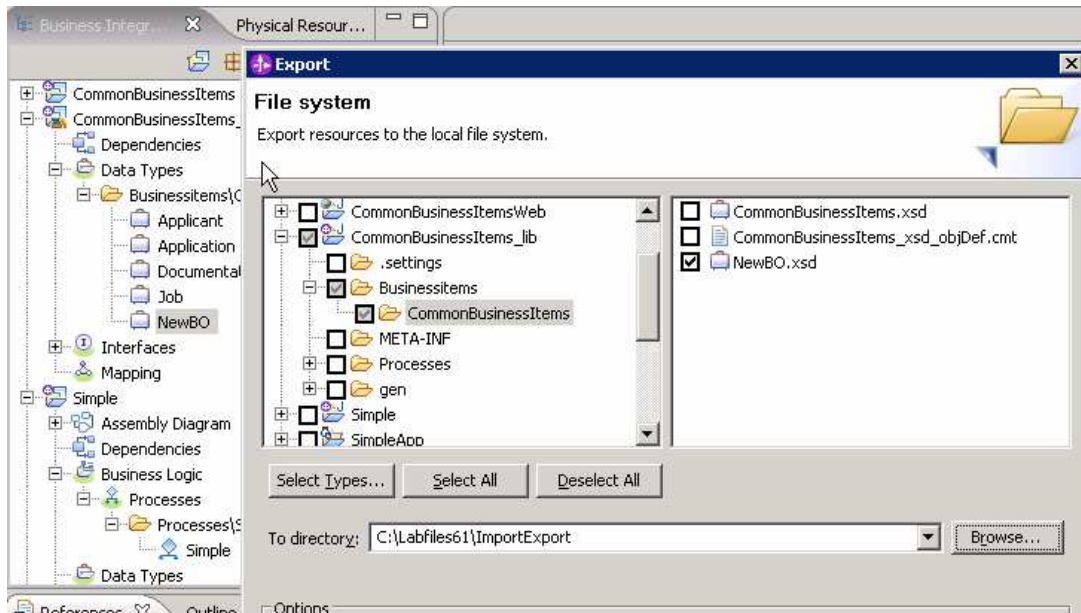
- ___ d. Export the change report and save it to the Simple.I02.cmr.zip file



- ___ 18. Based on previous experience, you know that the Business Analyst will want to know more details about the NewBO you just created so you go ahead and export it to the file system.
- ___ a. Right mouse click on the NewBO and select **Export...**



- ___ b. Make sure that only the NewBO is checked, set the directory as shown below

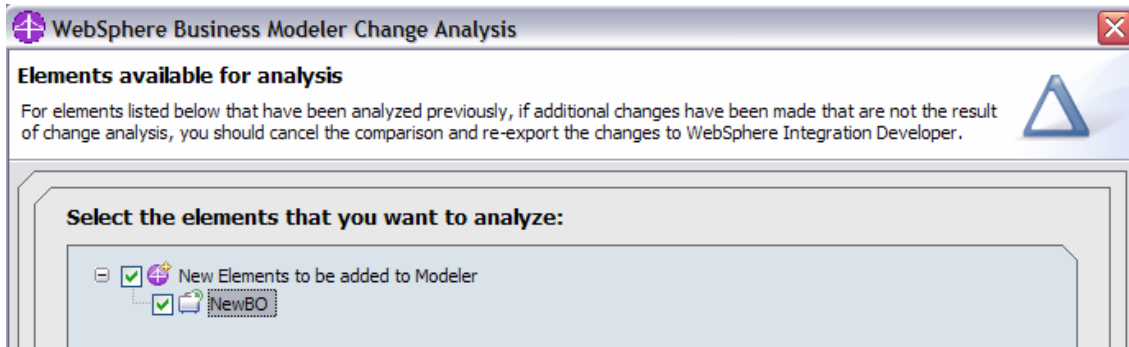


___ c. Select the **Finish** button.

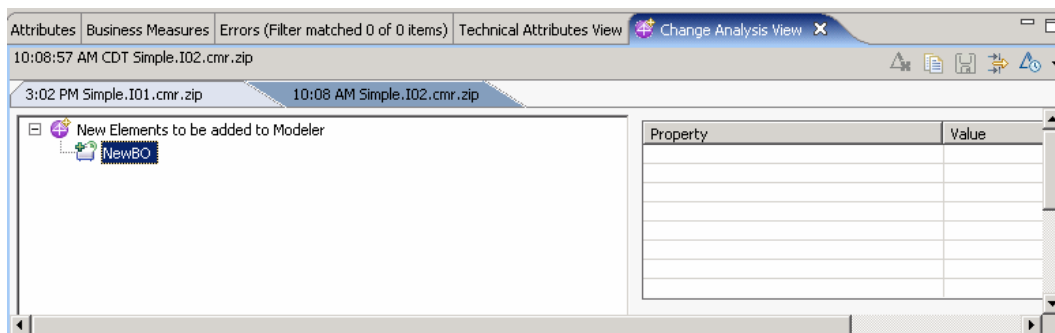
___ 19. This concludes the second iteration of the simple model synchronization exercise.

Part 3: Iteration 03

- ___ 20. Open WebSphere Business Modeler V6.1 on the Simple modeler workspace
- ___ 21. Analyze the Model Implementation Changes
 - ___ a. Analyze using the file, Simple.I02.cmr.zip

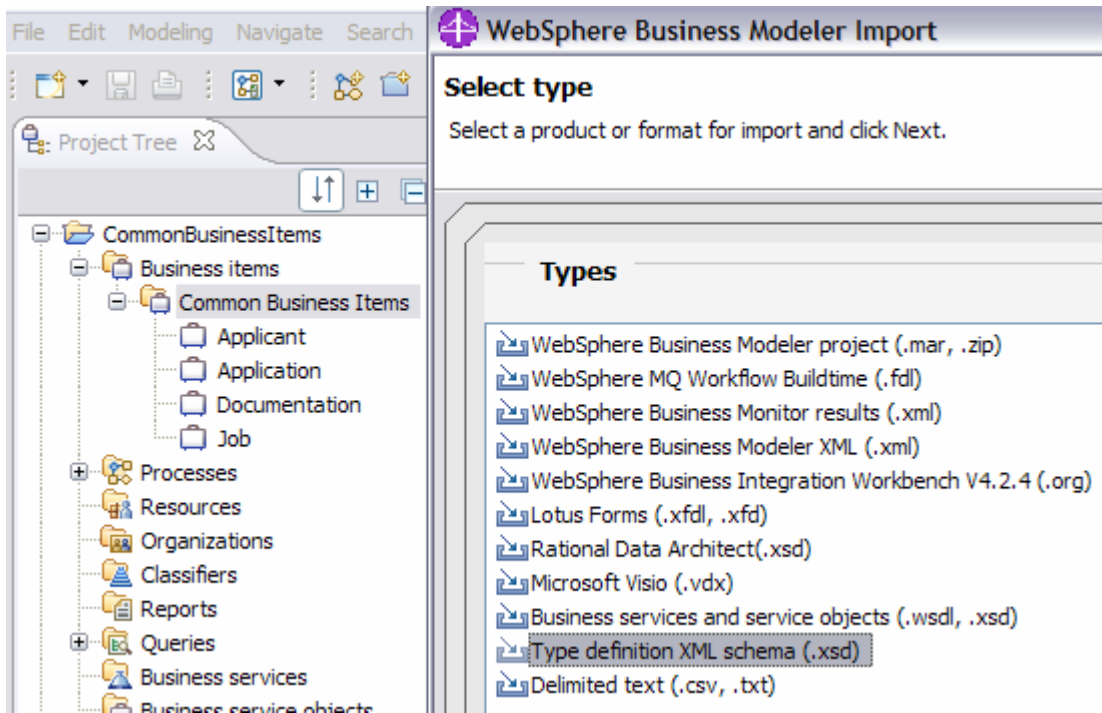


- ___ b. Select **Finish**.
- ___ c. As you can see the addition of the NewBO has been detected, but there is not really a lot of information about what this NewBO thing is. What are the attributes? What it used for, and so on?



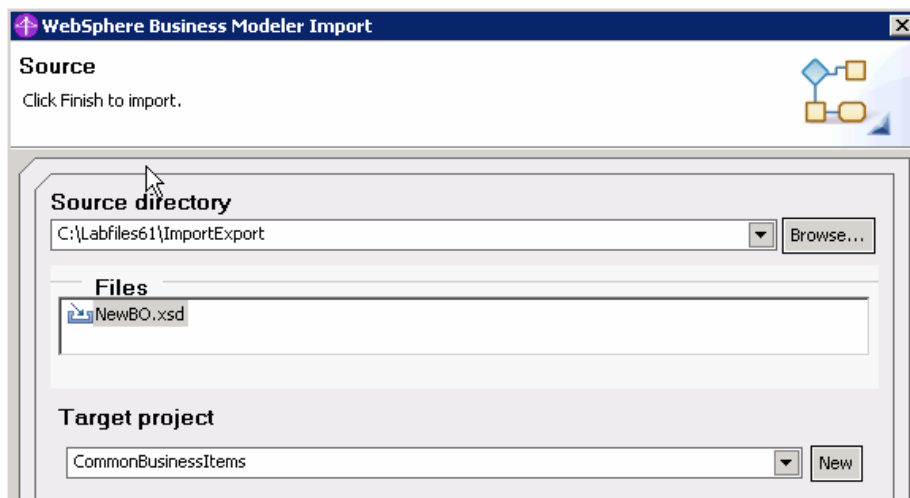
The best way to get the information is to export the business object from WebSphere Integration Developer and import it into WebSphere Business Modeler.

- ___ 22. Update the business model per the changes introduced by the Integration developer
 - ___ a. Import the NewBO by right mouse clicking on the **Common Business Items** folder and selecting **Import...**



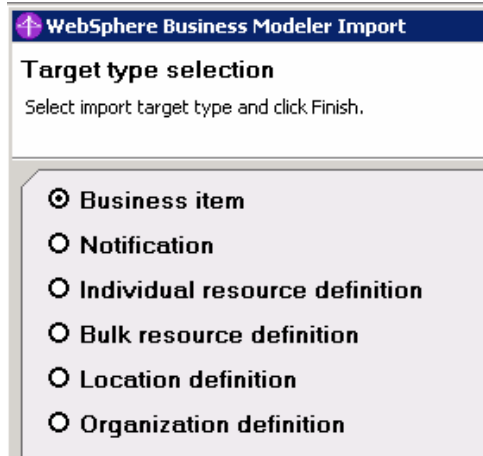
__ b. Use the **XML schema** import type.

__ c. Navigate to the Import\Export folder, select the NewBO.xsd



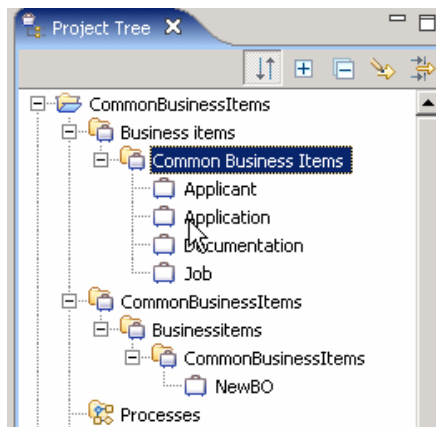
__ d. Notice the target project. Select the **Next** button.

__ e. Keep the default Target type as *Business Item*.

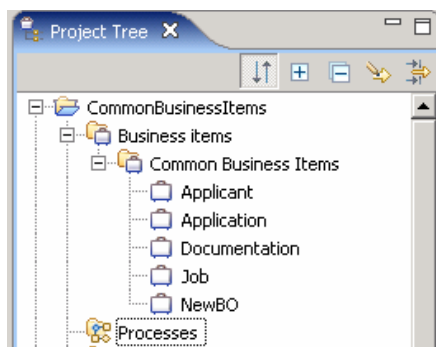


Select the **Finish** button.

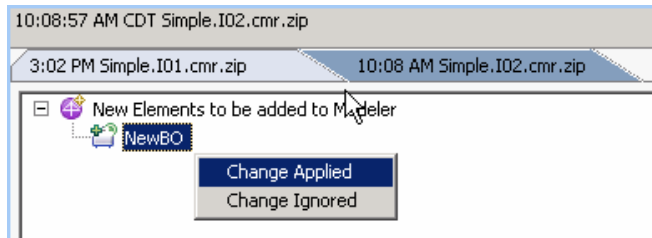
___ f. But wait, the NewBO went into a different folder.



___ g. This is easily managed at this point; drag the NewBO to the correct folder and then delete the extraneous folder.

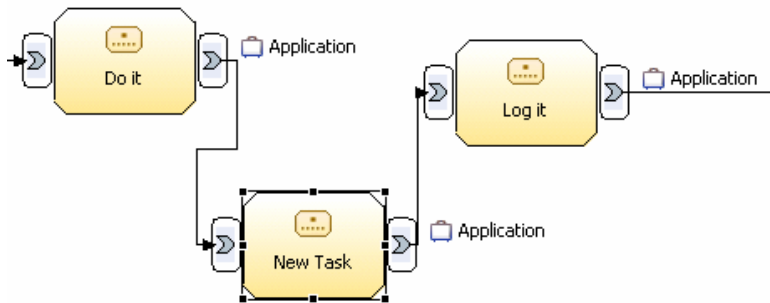


__ h. Accept the change in the *Change Analysis* view.



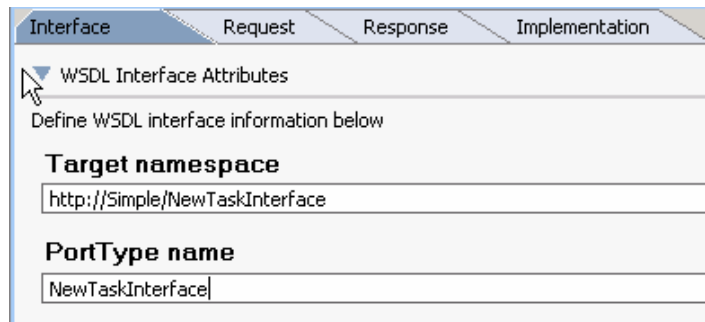
___ 23. Fill out the technical attributes for the New Task per the XYZ guidelines

__ a. Open the SimpleProcess1 business process and select the new task.



__ b. Select the *Technical Attributes View* tab in the pane below and complete the information for the Interface, Request, Response and Implementation pages as shown below.

1) Interface



2) Request

Interface Request **Response** Implementation

▼ Operation type

Select the type of operation

Request / 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 invoke activity information

Activity display name

Activity name

3) Response

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
Output Criterion	Input Criterion	Response

▼ WSDL Attributes

Define the WSDL interface information

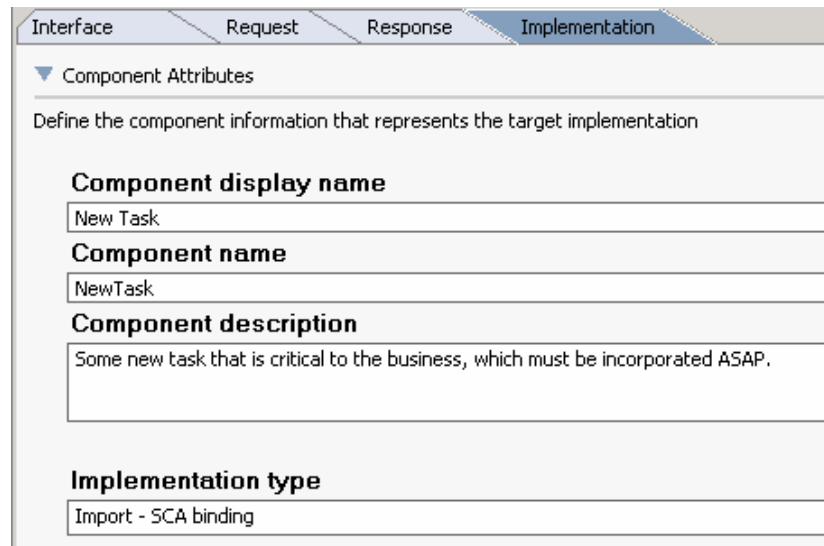
▼ WSDL Message Details

Define the WSDL message that represents the output of an operation

Message name

Part name

4) Implementation



Interface Request Response **Implementation**

▼ Component Attributes

Define the component information that represents the target implementation

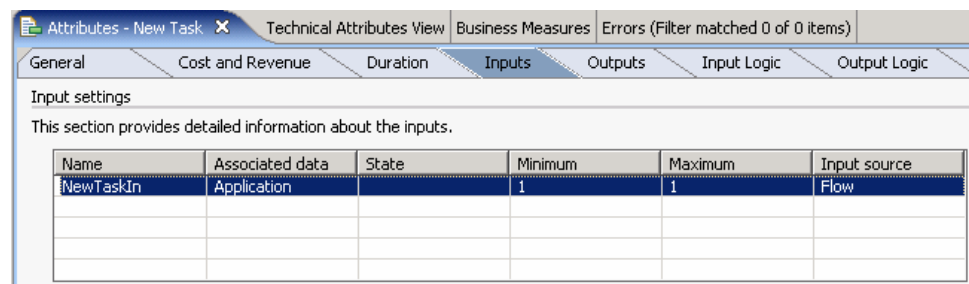
Component display name
New Task

Component name
NewTask

Component description
Some new task that is critical to the business, which must be incorporated ASAP.

Implementation type
Import - SCA binding

5) The guidelines also require that the inputs and outputs to the tasks be set.



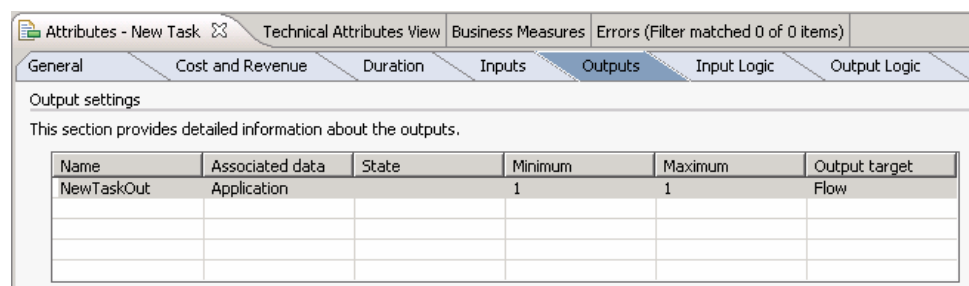
Attributes - New Task x Technical Attributes View Business Measures Errors (Filter matched 0 of 0 items)

General Cost and Revenue Duration **Inputs** Outputs Input Logic Output Logic

Input settings

This section provides detailed information about the inputs.

Name	Associated data	State	Minimum	Maximum	Input source
NewTaskIn	Application		1	1	Flow



Attributes - New Task x Technical Attributes View Business Measures Errors (Filter matched 0 of 0 items)

General Cost and Revenue Duration Inputs **Outputs** Input Logic Output Logic

Output settings

This section provides detailed information about the outputs.

Name	Associated data	State	Minimum	Maximum	Output target
NewTaskOut	Application		1	1	Flow

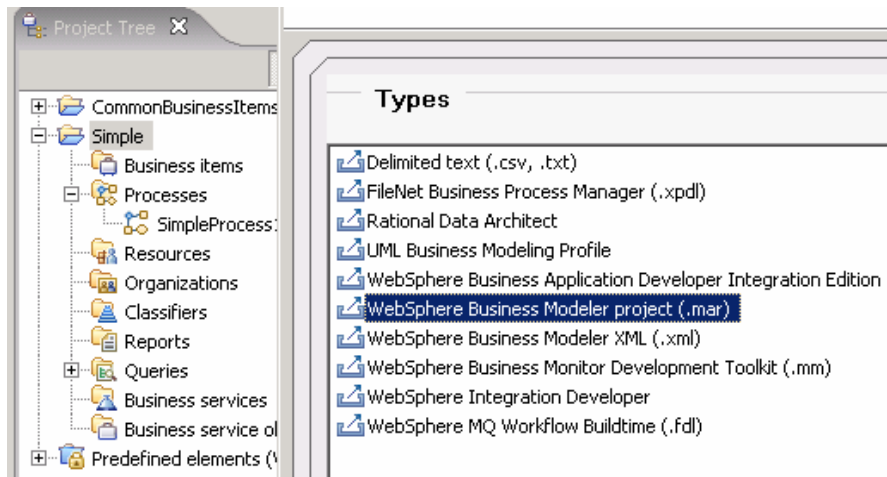
6) No changes required here.

7) Save your changes using **Ctrl-s**.

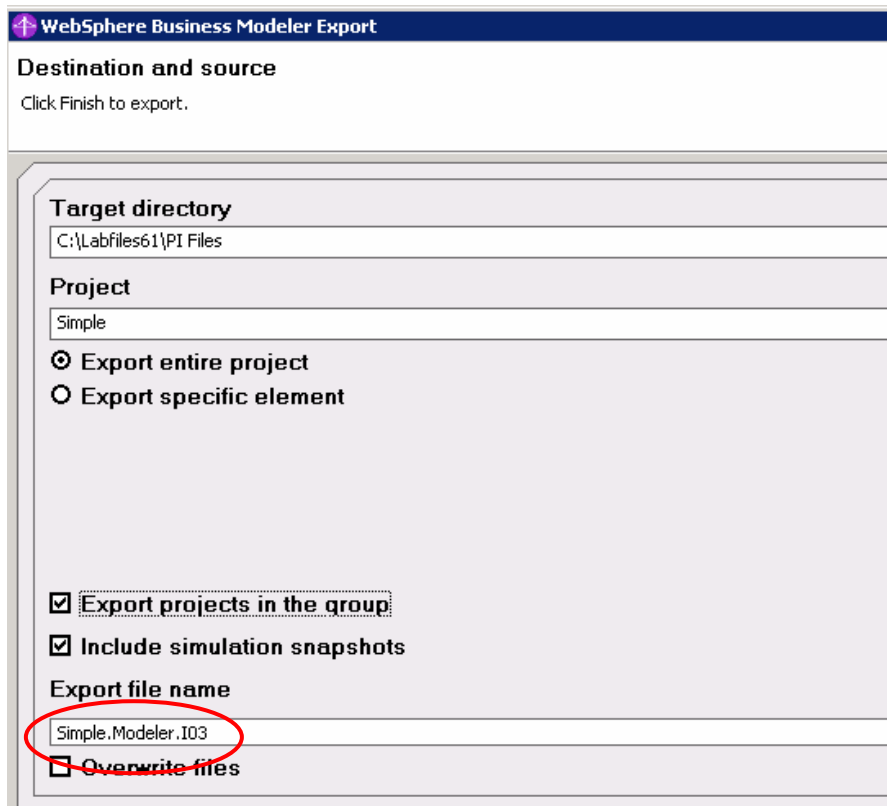
___ 24. Export the next iteration, I03

___ a. Export the modeler project as a model archive file (mar).

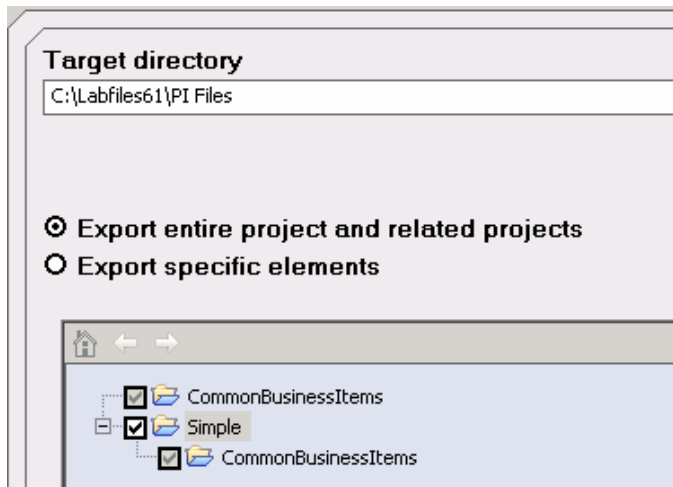
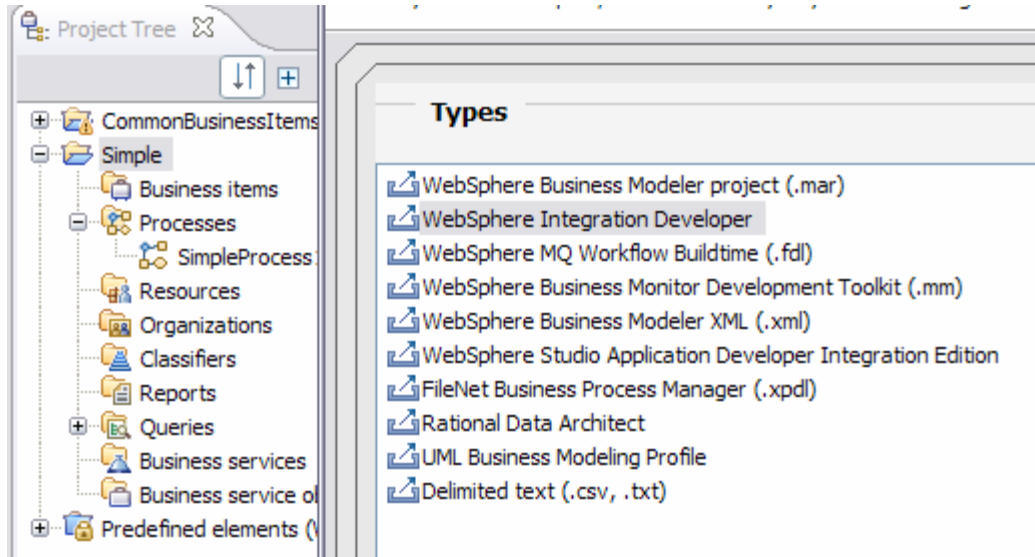
Possible best practice: Export the Modeler project as a model archive (.mar) at the end of each iteration, doing this will provide you with an archive that matches the project interchange file given to the integration developer. This will be a good restarting point should you need it.

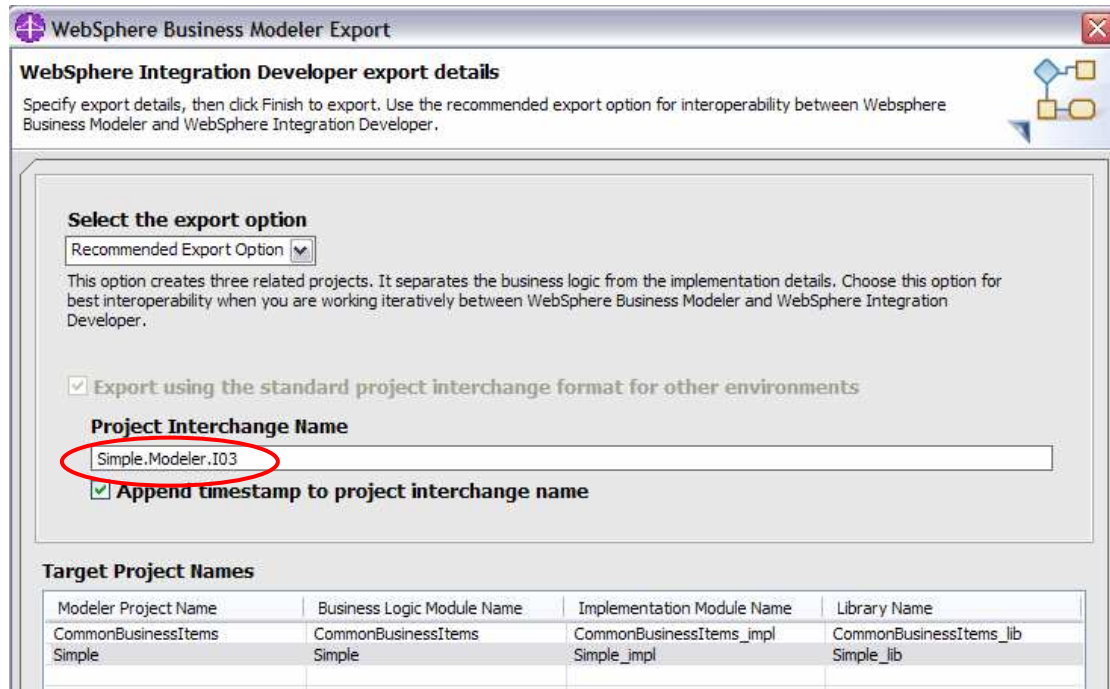


Export the entire project and any projects in the group.
Use the file name shown below.

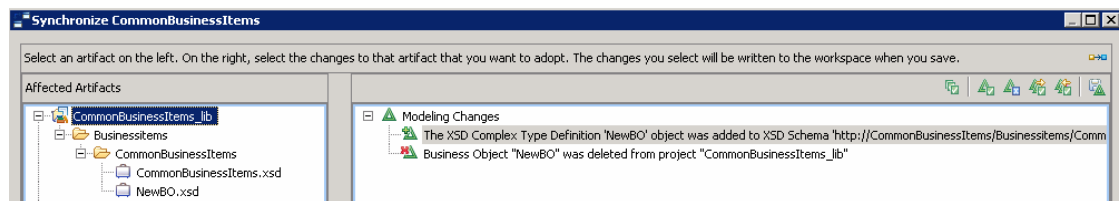


- __ b. Select **Finish**.
- __ c. Export the project interchange file for synchronization with implementation model.



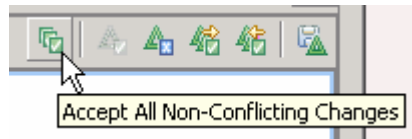


- ___ 25. This concludes the modeling portion of iteration I03.
- ___ 26. Open WebSphere Integration Developer V6.1 on the SimpleImpl workspace
- ___ 27. Synchronize with the modeler project interchange file, Simple.Modeler.I03_2007-mm-ddThh.mm.ss.zip
- ___ a. Because there were changes to interface and namespace names, there might be some changes that you were not expecting. Study them carefully.
 - ___ b. You might be tempted to reject the change regarding the deletion of the NewBO.xsd from the CommonBusinessItems_lib module, but do not reject the change.



- 1) In WebSphere Integration Developer the uniqueness of business objects is determined by file names and not the namespace. If this item is not deleted then there will be two definitions.
- 2) If there were references to the old definition then the Integration Developer will need to fix them up after it has been deleted.

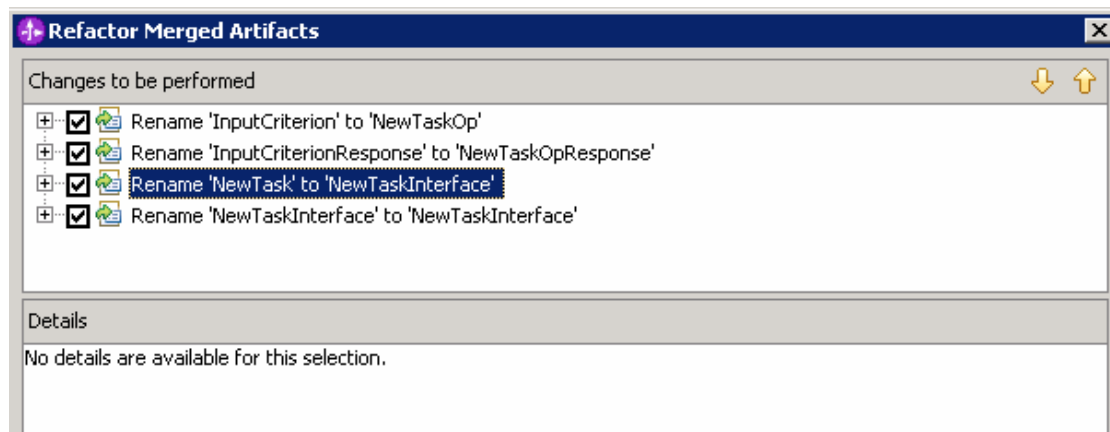
3) Accept all non-conflicting changes.



Hint: If there are build errors immediately after the synchronization, run clean from the Project menu item in the menu bar, *Project* → *clean*.

___ c. Select the **Commit** button.

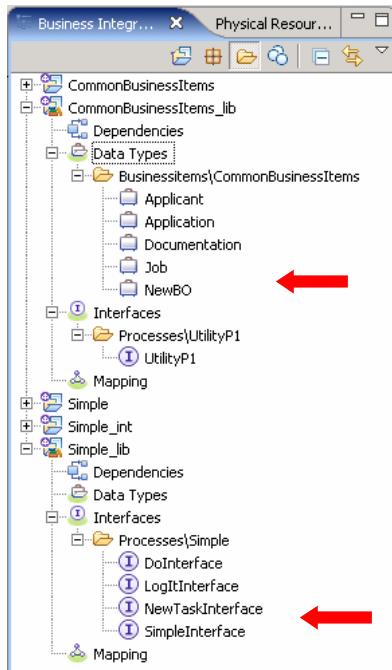
___ d. Browse/review the items in the *Refactor Merged Artifacts* dialog. Make sure you understand what the changes are. The *Refactor Merged Artifacts* dialog is displayed because there were changes to artifact names and interfaces.



___ e. Select the **OK** button.

___ 28. Inspect the changes that were made to the workspace as a result of the synchronization.

___ a. The NewBO should be in the same folder as the other business items that came from WebSphere Business Modeler and with the correct namespace and the interface name should be **NewTaskInterface**.



___ b. If you have not already deleted the interface for the UtilityP1 process, you can do so now. Since WebSphere Business Modeler does not know about interfaces it does not get reported back to the business analyst.

___ c. There should be no errors or warnings in the *Problems* view.

____ 29. This concludes the third and final iteration, the simple tour of the new model synchronization features available with WebSphere Business Modeler and WebSphere Integration Developer V6.1.

What you did in this exercise

You went through three iterations of the model → implement cycle, making changes in both the business model and the implementation models. With each iteration, the models were synchronized using the new model synchronization features of WebSphere Business Modeler V6.1 and WebSphere Integration Developer V6.1.

Solution instructions

There is no solution for this exercise.