



Business Process Management IBM Business Process Manager V7.5 IBM Integration Designer V7.5

Advanced integration service development process



© 2011 IBM Corporation

This presentation provides an overview of the Advanced Integration Service development process, which enables the incorporation of SCA services into process applications.

Goal and agenda

- Goal
 - Provide an overview of the Advanced Integration Service (AIS) development process
 - Focus on those aspects unique to development of an AIS
- Agenda
 - Basic requirements
 - Bottom up AIS development process
 - Top down AIS development process
- Prerequisite knowledge provided in these presentations
 - [Application development process](#)
 - [Process application authoring](#)
 - [Service Component Architecture development process](#)

The goal of this presentation is to provide you with an understanding of the Advanced Integration Service (AIS) development process. An advanced integration service is an SCA service that is called from a process application. The focus of the presentation is on those aspects unique to enabling the SCA service to be used as an advanced integration service and does not describe the more generic development process for the SCA service itself.

The presentation starts by defining the basic requirements that must be met in order for an SCA service to be used as an advanced integration service. Next the bottom up scenario is described, which takes an existing SCA service and uses it as an advanced integration service. Finally, the top down scenario is described, where the advanced integration service is defined first and then an SCA service is implemented to meet the definition.

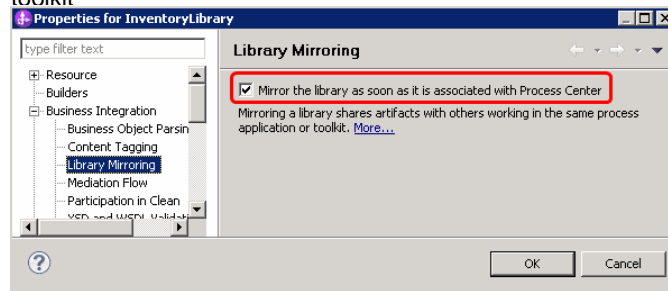
This presentation is more beneficial to you if you are already familiar with the other presentations addressing the development process in BPM Advanced. These include the presentations “Application Development Process”, “Process Application Authoring”, and “Service Component Architecture Development Process”. Links to these are provided on the slide.

Basic requirements

This section discusses the basic requirements for an SCA module to be used as an advanced integration service.

Basic requirements - Library

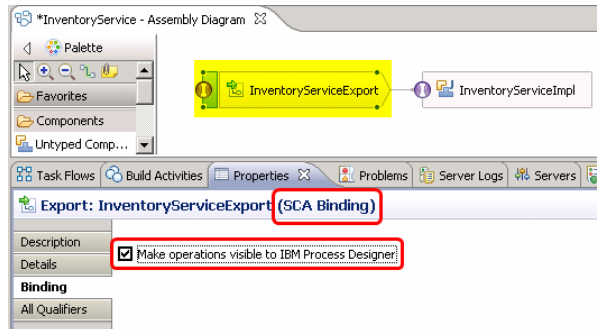
- Library
 - Must be mirrored if:
 - It contains the interface definition with operations for the AIS
 - It contains business object definitions that are shared with the process application or toolkit



An SCA service can have its interface and business object definitions contained in a library. In order for the interface and business object definitions to be available to the process application or toolkit which contains the AIS, the library must be mirrored. As is shown in the screen capture, mirroring is set in the properties dialog for the library under Business Integration, Library Mirroring.

Basic requirements – Export

- Export
 - Must have a default SCA binding
 - Operations must be visible to the Process Designer
 - Limited to one interface



SCA uses exports to publish an interface that allows an SCA component to be called by a requestor external to the SCA module. When the SCA module is to be used as an AIS, the export must be configured with only one interface. Also, it must use the default SCA binding type. Additionally, in the properties view for the export, in the Binding panel, the property to make operations visible in Process Designer must be selected, as is shown in the screen capture.

Basic requirements – Interfaces, operations and restrictions

- Interfaces and operations
 - Interface must use the document literal wrapped binding style
 - Operations cannot contain faults
 - Operation parameter names must be valid Java™ names
 - Each operation results in an AIS
- For a full list of restrictions see the Information Center topic entitled:
 - [Limitations when working with process applications and toolkits](#)
- Additional restrictions for XSD and WSDL are documented in a technote entitled:
 - [Some XML schema constructs are not supported in IBM Business Process Manager Advanced](#)

There are several capabilities of interfaces and business objects in SCA which are not compatible with their use as an advanced integration service. An interface used as an AIS must use the document literal wrapped binding style. The operations defined on the interface must only contain inputs and outputs and cannot contain any faults. The operation names themselves must be valid names for use with Java. When associated with a process application or toolkit, every operation defined in the interface of the export becomes an advanced integration service. For a more detailed list of the limitations and restrictions, see the Information Center topic entitled “Limitations when working with process applications and toolkits”. There is also a technote with additional information which is entitled “Some XML schema constructs are not supported in IBM Business Process Manager Advanced”. Links to both of these are contained in the slide.

Bottom up AIS development process

Now that the basic requirements and restrictions for an SCA service to be used as an AIS have been covered, this section looks at the development process for an AIS using the bottom up approach. With this approach, all of the work is done using the Integration Designer.

Overview of bottom up process

- Start with a fully functioning SCA service
 - Pre-existing service
 - New service that is used both as an SCA service and an AIS
 - New service specifically used as an AIS
 - In this case, the top down process is normally more appropriate
- Determine the process application or toolkit that will invoke the AIS
 - Typically a toolkit is more appropriate than a process application
 - Might be a pre-existing one or be newly created
 - Needs to be opened in the workspace
- Associate the SCA service with the Process Center, selecting the process app or toolkit
- Ensure requirements are correctly configured
 - Library mirroring
 - Export visible
- Publish to the Process Center
- Take a snapshot if AIS used from a toolkit

This slide presents an overview of the bottom up approach, serving as an introduction to the more detailed slides that follow.

The bottom up approach to developing an AIS starts with a fully functioning SCA service. There are a few different scenarios possible with the bottom up approach. The first, and probably the most common, is to start with a pre-existing service that is already in use, but now needs to be called as an AIS in addition to its existing usage. Another approach is when there is a requirement for a new service that can be called either as an AIS or as a regular SCA service. In this case, the new SCA service is developed and tested. It is then treated similar to the pre-existing service for development of its use as an AIS. The third possible scenario is the requirement for a new SCA service that will only be used as an AIS. However, in this case, there is no advantage in using the bottom up approach and the top down approach is preferable.

Once the SCA service has been identified, the next step is to determine if the AIS invocation is to be done by a process application directly or by a toolkit. In almost all cases, it is preferable to have the AIS in a toolkit so that it can be used by many process applications. The process application or toolkit must then be opened in the Integration Designer workspace that also contains the project for the SCA service. You then associate the project containing the SCA service with the Process Center, along with any required library projects. When this is done, the target process application or toolkit that is to contain the AIS must be identified.

The next step is to make sure that any library projects are configured to be mirrored and that export is configured to be visible. After this, publish the toolkit or process application to the Process Center. If the AIS is associated with a toolkit, you must take a snapshot of the toolkit before it can be used from any process applications.

The following slides look at each of these steps in more detail.

The SCA service

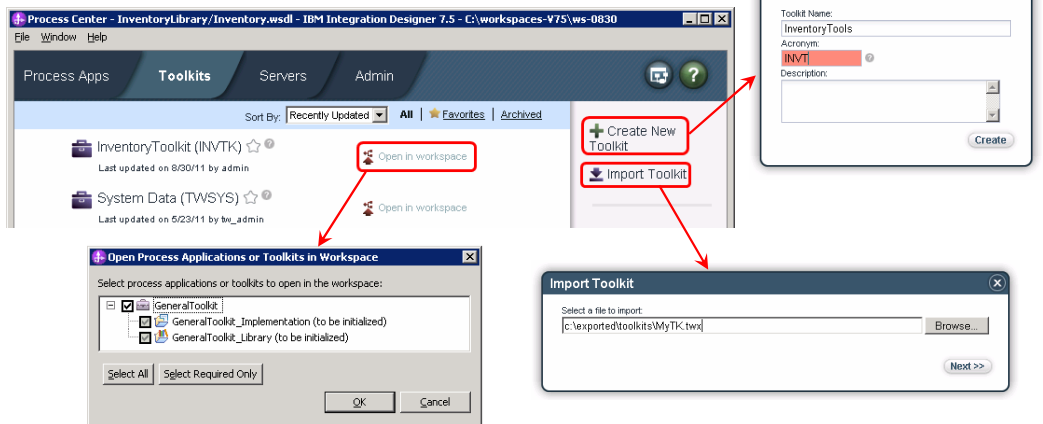
- Pre-existing service
 - Select mirroring in the properties of any libraries intended to be shared
 - Ensure there is an export with SCA default binding
 - Check the documented restrictions to ensure the SCA service can be used as an AIS
- New service
 - Follow development process documented in [Service Component Architecture development process](#) which is summarized here:
 - Define or import business object definitions and interfaces
 - Create modules containing components, exports and imports
 - Implement the components
 - Configure the export and import bindings
 - Test in the unit test environment
 - Perform same checks as pre-existing service

Before incorporating the SCA service into a process application or toolkit, you should make sure that the SCA service meets the requirements, which were discussed in the previous section.

If you have a pre-existing service, make sure that the libraries are configured as mirrored. Also, check the export that exposes the interface used as the AIS to ensure that it has the SCA default binding type configured. Note that at this point, the property making the operations visible in Process Designer is not available to be selected, and therefore this must be done later. Finally, check the documented restrictions for the interface, operations and business objects to ensure that none of the restrictions exist in the definition of the SCA service.

In the case of a new SCA service, the first thing to do is to implement the service. The process for this is documented in the presentation “Service Component Architecture development process”, a link to which is contained in this slide. Summarizing that process, the business objects and interface must be created or imported. Then the SCA module with imports, exports and components is created and the components are implemented. The import and export bindings are then configured and the service is tested using the unit test environment in Integration Designer. After all this is done, you should also perform the same steps as are defined for the pre-existing service.

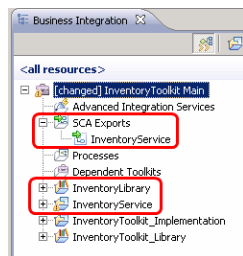
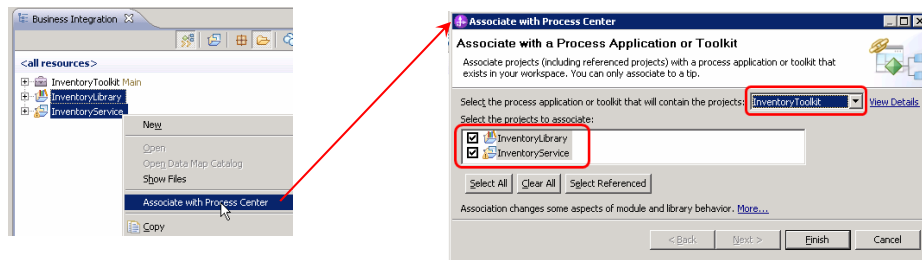
Adding a toolkit to the workspace



- Using the Process Center perspective you can:
 - Create or import a toolkit in the Process Center repository
 - Open in the workspace a toolkit in the Process Center repository

From this point on, the examples shown and discussed assume that the AIS is being used from a toolkit, rather than directly from a process application. The toolkit must be opened in the Integration Designer workspace that contains the project containing the SCA service. In the case where the toolkit is not already in the Process Center repository, you can either create a new one or import one from a file containing a toolkit that had been previously exported. Now that you have either an existing toolkit, new toolkit or imported toolkit in the Process Center repository, you select to have the toolkit opened in the workspace.

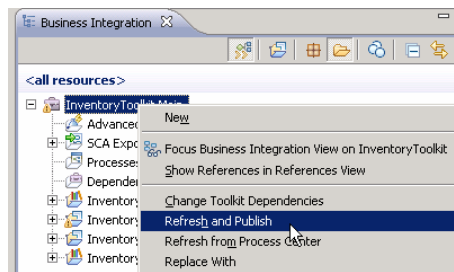
Associate SCA libraries and modules with a toolkit



- The toolkit must be open in the workspace
- Select Associate with Process Center for the libraries and modules
- Complete the dialog
 - Ensure correct library and module projects are selected
 - Select the target toolkit from the drop down
- Associated SCA libraries and modules now contained within the toolkit
- Ensure libraries are mirrored and exports are visible

Now that the toolkit is opened in the workspace, the SCA library and module projects can be associated with the toolkit. As seen in the screen captures, you start with the Business Integration view, select the library and module projects and then select to associate them with the Process Center from the pop-up menu. This opens a dialog in which you can ensure you have the right libraries and modules selected, and from a dropdown, select the toolkit with which they are to be associated. After selecting finish on the dialog, you should then ensure that the libraries are mirrored and that the export is configured to be visible to the Process Designer. The screen capture on the lower left of the slide shows the Business Integration view. Notice that the SCA module and library are now contained within the toolkit and that the export, which you marked as visible to Process Designer, is listed under SCA exports for the toolkit.

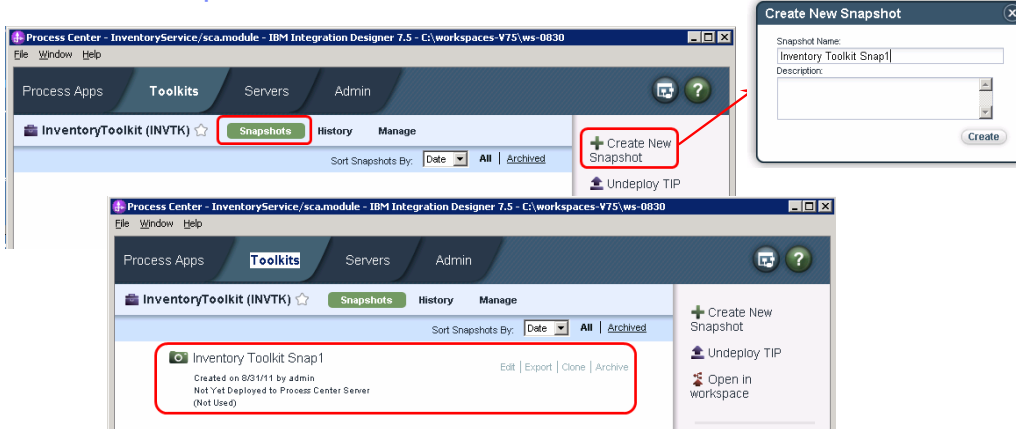
Publish to the Process Center



- Select Refresh and Publish for the toolkit
- Contents of the toolkit, including the SCA module and library, are published to Process Center repository
- An AIS is created in the repository for each operation on the visible interface of the export

Now that the SCA service is part of the toolkit, you should select refresh and publish from the pop-up menu for the toolkit. At this time, the contents of the toolkit, including the SCA module and library, is published to the Process Center repository. The publishing process also generates an AIS in the toolkit for every operation defined on the interface of the export that is defined as visible to the Process Designer.

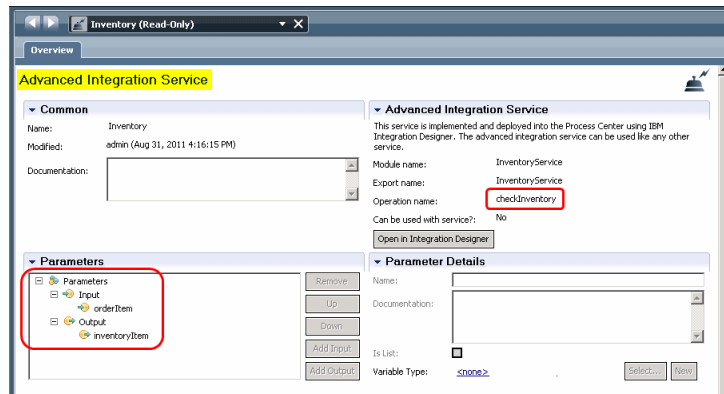
Create a snapshot



- From the Process Center perspective
 - Open the toolkit from the list of toolkits
 - Select Snapshots
 - Create the snapshot, giving it an appropriate name
 - The snapshot is now shown

A snapshot of the toolkit is required before an AIS in the toolkit can be called from a process application. From the Process Center perspective in Integration Designer, open the toolkit from the list of toolkits. Once the toolkit is opened, select Snapshots and then Create New Snapshot. In the dialog, provide an appropriate name for the snapshot. The screen capture in the lower portion of the slide now shows the snapshot.

The AIS is now available



- This is a screen capture of the new AIS in Process Designer
- Since there is a snapshot of the toolkit, it is available for use by process apps
- The operation name and the parameters reflect the interface and business object definitions used by the export and defined in the SCA library

This screen capture is from the Process Designer. It shows the properties of an advanced integration service that was generated when the toolkit containing the SCA service was published to the Process Center. Because there is a snapshot of this toolkit, this advanced integration service is available to be called from process applications. In the upper right you can see the operation name, which is the same name as the operation defined in the SCA interface. In the lower left you can see the input and output parameters which reflect the inputs and outputs defined for the operation in the SCA interface.

Top down AIS development process

The next section looks at the development process for an advanced integration service using the top down approach. Unlike the bottom up approach, which is done using the Integration Designer, this process starts by using the Process Designer and then transitions to using the Integration Designer.

Overview of top down process

- Start in Process Designer
 - Create a toolkit (or process application)
 - Define the data objects needed
 - Define the AIS implementation
 - Open the AIS implementation in Integration Designer
- In Integration Designer
 - Check the business objects and interface generated
 - Generate the implementation of the AIS, selecting the component type
 - Complete implementation of the component
 - Test in the local process server test environment
 - Refresh and publish to the Process Center
 - Create a snapshot of the toolkit to make it available for use

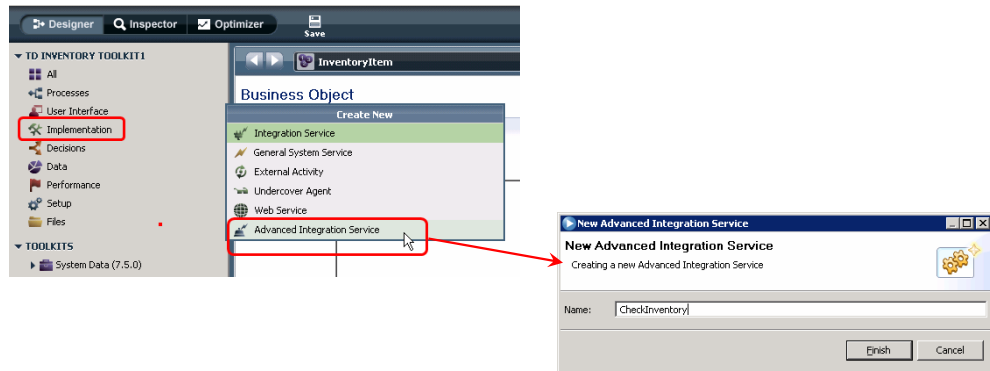
Before going into the details, this slide provides an overview of the advanced integration service top down development process. In the Process Designer you create a new toolkit or process application, or you can select an existing one. Note that in the slides that follow it is assumed this is being done with a toolkit, as that is the most common.

Next the data objects that will serve as the input and output parameters need to be defined, if appropriate ones do not already exist. The AIS is then created, identifying the data objects used for input and output. At this point the AIS is ready to be implemented, so you must open it in the Integration Designer. This causes the Integration Designer to generate an interface and business objects reflecting the inputs and outputs of the AIS. It is a good idea at this point to examine the interface and business objects so that you have an understanding of what was generated. The next step is to generate the SCA assembly for the implementation, specifying the SCA component type that is to be used to implement the logic. After completing the implementation, the SCA service can be tested in the local process server test environment. Once the testing is successful, the toolkit can then be published to the Process Center and a snapshot taken so the implementation is available for process applications to use.

The following slides look at these steps in more detail.

Create the AIS implementation

- As you normally do in Process Designer
 - Create a new toolkit
 - Define the data objects to be used by the AIS
- Create the new AIS



17

Advanced integration service development process

© 2011 IBM Corporation

To begin the process, use the Process Designer to create a new toolkit and to define the appropriate data objects for the AIS input and output. These steps are similar to how you normally do this in Process Designer for any toolkit. Then you create the AIS as is shown in these screen captures, In the toolkit, select Implementation and then Advanced Integration Service. This opens a dialog where you provide the name of the AIS.

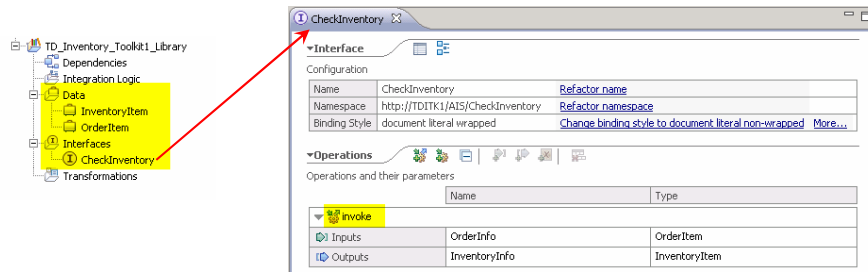
Define the parameters and open in Integration Designer

- Define input and output parameters
- Save the AIS
- Open in Integration Designer

The screenshot displays the 'Advanced Integration Service' configuration panel. The 'Parameters' section shows 'OrderInfo' as an input parameter and 'InventoryInfo' as an output parameter. The 'Parameter Details' for 'OrderInfo' are visible. A red box highlights the 'Open in Integration Designer' button, which is linked to a dialog box titled 'Open Process Applications or Toolkits in Workspace'. This dialog lists 'TD_Inventory_Toolkit1' and its sub-components for initialization. The bottom right shows the 'Business Integration' workspace view, where the 'InventoryToolkit1' and 'TD_Inventory_Toolkit1' are listed under 'resources', and the 'CheckInventory' service is shown as 'unimplemented'.

You are then presented with the panel for defining the AIS implementation. The screen capture in the upper right of the slide shows the portion of the panel used to define the inputs and outputs. Looking at the entire panel in the lower left of the slide, you can see that the module, export and operation names are not yet filled in. This is because the SCA module used to implement the AIS has not been created yet. Selecting the Open in Integration Designer button presents a dialog showing the two projects to be created, the implementation module and the library. The screen capture at the lower right shows the business integration view in Integration Designer. Notice that the toolkit contains the implementation module and library. It also shows the advanced integration service. Note that the AIS is shown to be unimplemented, meaning that the SCA assembly has not been generated yet.

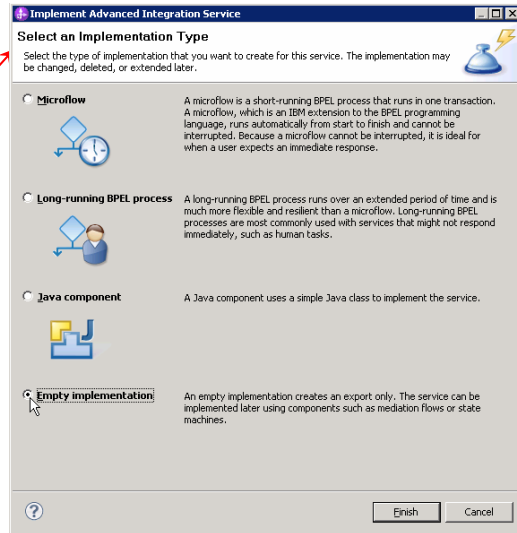
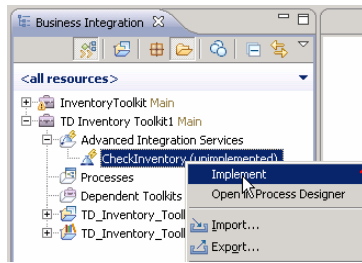
Check what was generated for you



- Data objects are created for the input and output parameters
- An interface is generated with an “invoke” operation

Before proceeding with the implementation, it is a good idea to examine what has already been generated. In the library you will see the business objects needed for the AIS input and output. An interface has also been generated. The interface has the same name that the AIS was named in the Process Designer. The interface has one operation called invoke, and the inputs and outputs use the appropriate business objects.

Generate the AIS implementation skeleton

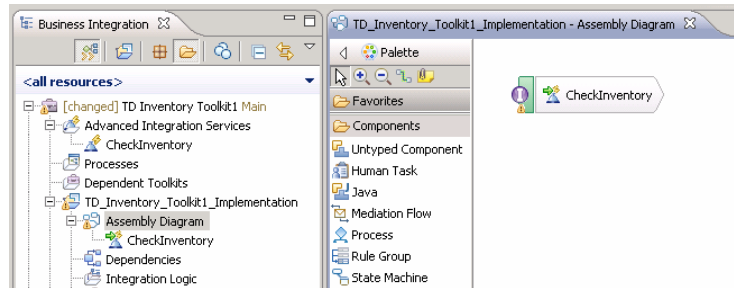


- Integration Designer generates the SCA assembly diagram, including an export visible to Process Designer
- The SCA component is optionally generated
 - BPEL microflow
 - BPEL long-running process
 - Java component
- For any other component type, specify empty implementation and only the export is generated

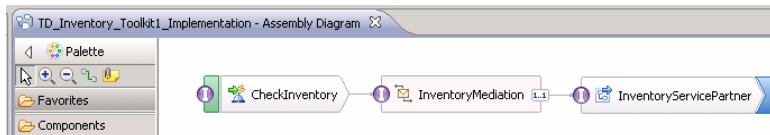
Now you are ready to have Integration Designer generate the assembly diagram used to contain the implementation of the SCA service. You do this by selecting **Implement** from the pop-up menu for the AIS, which opens the dialog shown in the slide. In the dialog you have the ability to select the SCA component type to generate, but only if you want it to be a BPEL microflow, a BPEL long running process or a Java component. If you plan to implement using a different component type, such as a meditation flow, you select **Empty implementation** and the component is not generated for you in the assembly diagram. In all cases, an export is generated that supports the interface of the AIS, and the export has already been configured to be visible to the Process Designer.

The SCA assembly diagram

- Generated assembly, in this case without a component



- After implementation of a mediation flow and a call to another service



21

Advanced integration service development process

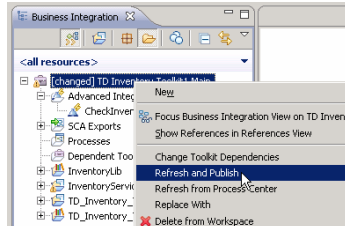
© 2011 IBM Corporation

You then implement the SCA component as you normally do for any SCA service implementation. In these screen captures, the one at the top shows the assembly that gets generated when empty implementation is selected, and therefore only contains the export. In the bottom screen capture, you see a completed assembly diagram showing that the implementation in this case is a mediation which makes a call to a web service.

At this point you might ask: If the target service is a web service, why was an AIS implemented in Process Designer rather than a call directly to the web service? There are many reasons why this might be the case. One is that the interface to the web service contained more data than was required for use in process applications. Using the AIS allowed the interface to process applications to contain the exact data content required and the SCA mediation is used to map between that interface and that of the target web service.

Complete the implementation

- Unit test the service using the local process server test environment
- Refresh and publish the toolkit to the Process Center



- Test the AIS using the Inspector
- Create a snapshot of the toolkit
- The AIS is now ready to be used by process apps

Once the implementation of the SCA service is complete, you should test the SCA service in the local process server test environment, independent if it's used as an AIS. This allows you to debug and refine the implementation as you normally do for any SCA service. Once the testing is complete, you refresh and publish the toolkit to the Process Center. Another round of testing can now be done by calling the AIS from the Inspector in Process Designer. Once that testing is complete, create a snapshot of the toolkit and the AIS is now ready for use by process applications.

Summary

- Basic requirements
- Bottom up AIS development process
- Top down AIS development process

The goal of this presentation was to provide you with an understanding of the Advanced Integration Service development process. This was done by defining the basic requirements that must be met in order for an SCA service to be used as an advanced integration service. Next the bottom up scenario was described, which takes an existing SCA service and uses it as an advanced integration service. Finally, the top down scenario was described, where the advanced integration service is defined and an SCA service is then implemented to meet the definition.

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 email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_BPMv75_AIS_DevProcess.ppt

This module is also available in PDF format at: [../BPMv75_AIS_DevProcess.pdf](..../BPMv75_AIS_DevProcess.pdf)

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



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) 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>

Java, and all Java-based trademarks and logos are trademarks of Oracle and/or its affiliates.

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

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 2011. All rights reserved.