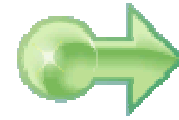




IBM Software Group

# WebSphere® Process Server V6



## *WebSphere Business Integration Server Foundation Migration*



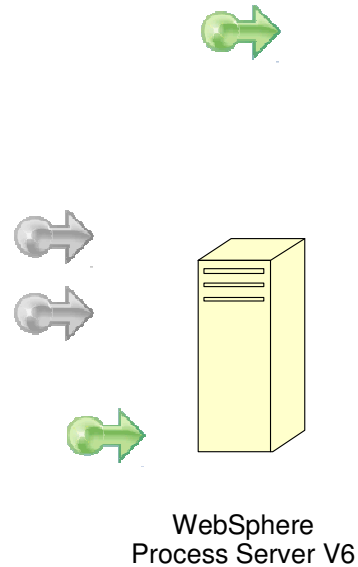
@business on demand.

© 2006 IBM Corporation  
Updated January 9, 2006

This presentation will cover migrating to WebSphere Process Server V6.0 from previous releases of WebSphere products.

## Goals

- Discuss details of the migration utilities for WebSphere Business Integration Server Foundation
  - ▶ WebSphere InterChange Server V4.2 and V4.3
  - ▶ WebSphere MQ Workflow V3.5 and V3.6
  - ▶ WebSphere Business Integration Server Foundation V5.1



WebSphere Process Server is the merger of 3 existing product lines.

- WebSphere InterChange Server
- WebSphere MQ Workflow
- WebSphere Business Integration Server Foundation

This presentation will discuss the details of the WebSphere Business Integration – Server Foundation migration

## Agenda

- **Introduction**
- Overview
- Java™ Snippet Migration
- Running the Migration Wizard
- Post Migration Tasks
- Limitations
- Summary
- References



The agenda for this presentation is to focus on the steps involved in migrating from WebSphere Business Integration Server Foundation to WebSphere Process Server V6.

## Introduction – Before You Begin

- This discussion will address *source artifact migration*.
- Migration from WebSphere Business Integration Server Foundation V5.1.1 to WebSphere Process Server V6 uses a utility to convert V5.1.1 BPEL artifacts to V6 artifacts.
- A WebSphere Business Integration Server Foundation V5.1.1 EAR with BPEL cannot be deployed to WebSphere Process Server V6
- A WebSphere Business Integration Server Foundation V5.1.1 EAR that uses Business Rule Beans or Extended Messaging can be deployed to WebSphere Process Server V6.
  - ▶ The components must be installed with the “custom install” option when installing WebSphere Process Server V6



Of the three migration paths to WebSphere Process Server V6, WebSphere Business Integration Server Foundation is the most similar to WebSphere Process Server V6. Although it is similar, there are significant differences which make it difficult to provide binary compatibility between the two. The EAR file created by WebSphere Business Integration Server Foundation V5.1.1 cannot be deployed and run in the WebSphere Process Server V6 runtime if the application uses BPEL. The application source artifacts must be converted to the new programming model.

The migration (conversion) utility is invoked from WebSphere Integration Developer using the import function when importing a WebSphere Studio Application Developer Integration Edition Service Project.

If the WebSphere Business Integration Server Foundation 5.1.1 EAR has no BPEL then it can run in WebSphere Process Server V6.

If the WebSphere Business Integration Server Foundation 5.1.1 application uses Business Rules Beans or Extended Messaging then you must install these components using the “custom install” option when installing WebSphere Process Server V6.

## Introduction - Before You Begin

- There are manual steps which will require additional work after the import in order to complete the source artifact migration process.
- Read the Migration Guide that is available in the WebSphere Business Process Integration Information Center
- Review the *“Migrating source artifacts to WebSphere Integration Developer from WebSphere Studio Application Developer Integration Edition”* in the WebSphere Integration Developer 6.0.1 Information Center.
  - ▶ *“Best Practices for Source Artifact Migration”*
- Upgrade to WebSphere Integration Developer / WebSphere Process Server 6.0.1
- Become familiar with the new WebSphere Process Server SCA programming model by doing tutorials and exercises.

5

Migration – WebSphere Business Integration Server Foundation V5.1.1

© 2006 IBM Corporation

Before you begin you should become familiar with the known limitations, which are listed at the end of this presentation.

You should also:

Read the Migration Guide.

Review the Best Practices in the WebSphere Integration Developer V6.0.1 Information Center

Upgrade to WebSphere Integration Developer / WebSphere Process Server V6.0.1

Become familiar with the SCA programming model

## Agenda

- Introduction
- **Overview**
- Java Snippet Migration
- Running the Migration Wizard
- Post Migration Tasks
- Limitations
- Summary
- References



The Overview section will provide a high level perspective, the “Big Picture” of the migration process from WebSphere Business Integration Server Foundation, to WebSphere Integration Developer V6 , using the Migration Wizard.

## Overview - The Information Center

### Information Center

<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp>

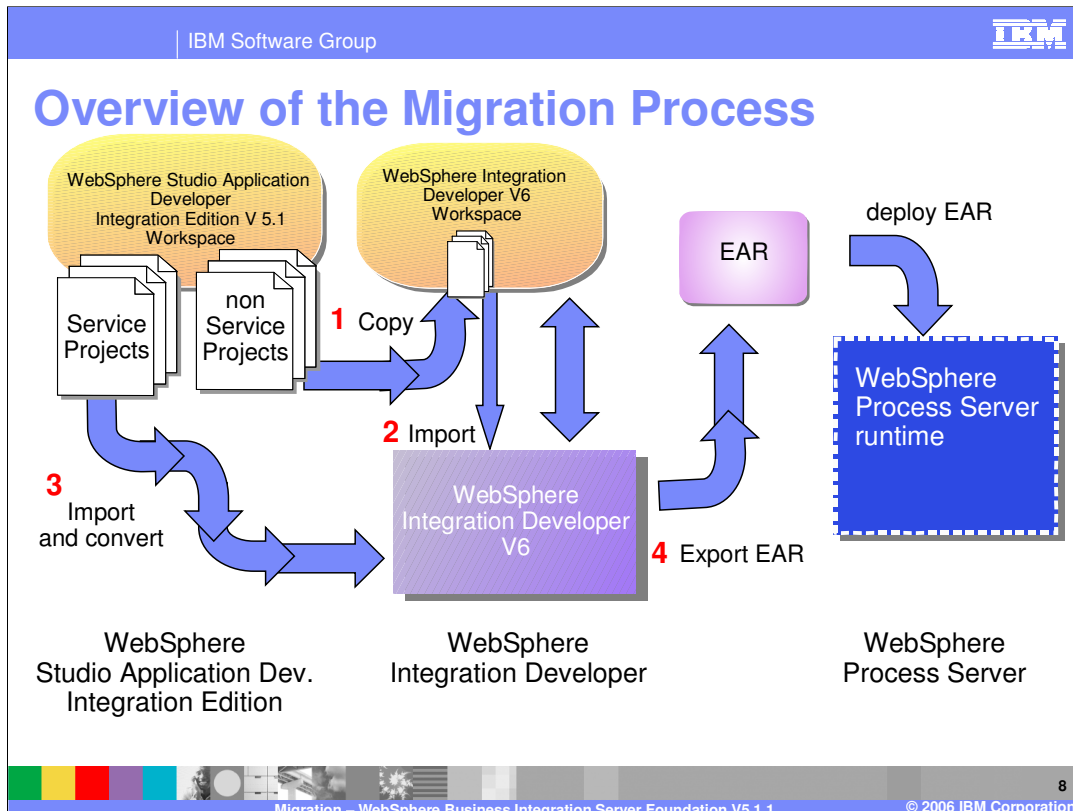
Select the book for WebSphere Integration Developer

Then on the right, select the topic migrating applications

Installing and migrating  
[Installing WebSphere](#)  
[Integration Developer](#)  
[Migrating applications](#)

The Information Center is an excellent place to get additional details about the migration process.

A PDF version of the migration guide is also available from the Welcome Page of WebSphere Integration Developer V6.



From a high level perspective, the migration process consists of the following steps:

- 1) Copy the WebSphere Studio Application Developer Integration Edition non-Service projects to a WebSphere Integration Server V6 Workspace
- 2) Import the non-Service projects into WebSphere Integration Developer V6
- 3) Import and convert the BPEL artifacts in the WebSphere Studio Application Developer Integration Edition Service projects into WebSphere Integration Developer V6 artifacts

Resolve the errors, wire up the module assembly and tune the migrated artifacts for BPEL / SCA, and then

- 4) deploy the application to the WebSphere Process Server V6 runtime.

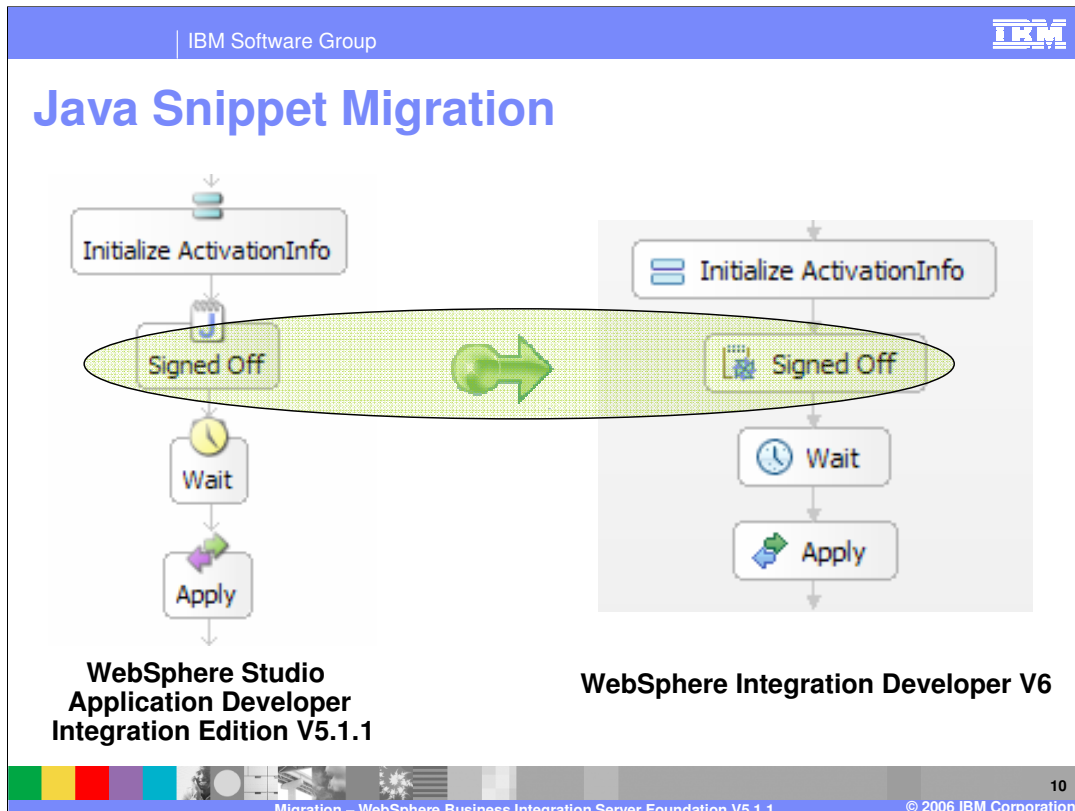


## Agenda

- Introduction
- Overview
- **Java Snippet Migration**
- Running the Migration Wizard
- Post Migration Tasks
- Limitations
- Summary
- References



Because Java Snippets provide access to the underlying programming model, when the programming model changes, the Java Snippets may also be affected. This next section will focus on the Java Snippets and things to watch for, and things to change before you run the Migration Wizard.



Java Snippets were introduced into the programming model to facilitate data management. With Java Snippets the Business Process Developer could use the Java programming language and the WSIF APIs to access and manipulate the data associated with the business process.

With WebSphere Process Server V6, the APIs used to interact with the data have changed, and as a result the Java Snippets will be affected. The automatic migration tool will convert the WSIF API calls to the DataObject API calls where possible and if specified, the original Java code will be preserved in comments.

This next section will discuss a few of the details related to Java Snippet Migration.

## Java Snippet Migration

- A new data model with WebSphere Integration Developer/ WebSphere Process Server V6
- Not all supported WSIF APIs can be directly migrated to WebSphere Integration Developer .
  - ▶ Not all Java snippets will be completely migrated.
  - ▶ Most of the standard snippet APIs will be converted
  - ▶ Use the option to *preserve the Java original Java snippet*.
- WSIF API calls are migrated to DataObject API calls where possible.
- Custom Java classes that accept WSIFMessage objects will need to be manually migrated to...
  - ▶ accept and return `commonj.sdo.DataObjects`



The new data model in WebSphere Integration Developer V6 is significantly different from WebSphere Studio Application Developer Integration Edition V5.1.1. Because of this difference, Java Snippets will require attention both before and after you run the migration wizard.

The Migration Wizard converts the WSIF API calls to the DataObject APIs where possible. Not all of the Java snippets will be completely converted.

A feature to *preserve the original BPEL Java snippets in the comments* is provided. Using this feature, the logic of the Java snippet can be maintained, providing a guide for the post migration activities.

When developing Java Snippets it is often convenient to create additional Java utility classes that accept and return WSIFMessage objects as parameters. The Migration Wizard does not automatically convert these classes; you must update them manually after you run the Migration Wizard.

## Java Snippet Migration

- WSIF elements that are not automatically migrated
  - ▶ WSIFMessage metadata APIs
  - ▶ EndpointReference/EndpointReferenceType APIs
- Issues with XSD type and WSDL definition names that are known to be problematic.
  - ▶ Complex types with duplicate names
    - Two different definitions with identically qualified names
      - Target namespace URI and local part
    - Different namespaces and but identical local names
  - ▶ Complex types with local name in java.lang.package.
    - Example: String

When migrating the Java Snippets there are known limitations which you should address before you run the Migration Wizard.

Java statements that use the WSIFMessage metadata APIs and the EndpointReferences are not automatically converted. You must convert them manually after you run the Migration Wizard.

When dealing with the local variable names in the Java Snippets, there is the potential for conflicting XSD/WSDL definitions, because WebSphere Integration Developer allows **at most one** unique definition of a type (as identified by its qualified name) in the classpath, whereas WebSphere Studio Application Developer Integration Edition was not as strict.

If there are conflicts, the Java Snippet may not be converted correctly. Also, if the name of the local variable is the same as the name of a class defined in the java.lang.package, such as **String**, the type may not be converted correctly.

When migrating business processes with Java snippets, review the names of the local variables used in the Java Snippets and identify whether any of these problematic conditions exist. If possible, make the modifications in WebSphere Studio Application Developer Integration Edition, altering the V5.1 source artifacts, before you run the Migration Wizard.

## Java Snippet Migration – BPEL Variables

- BPEL variables cannot have the same name as the local variables
  - ▶ Before running the Migration Wizard, change the names of the local variables that are the same as the BPEL variables.
- Read-only and read-write BPEL variables
  - ▶ WebSphere Studio Application Developer IE V5.1.1 example:
    - Read-write  
`getVariableAsWSIFMessage("MyInputVariable", true)`
    - Read-only  
`getVariableAsWSIFMessage("MyInputVariable", false)`
  - ▶ WebSphere Integration Developer V6 example:
    - Uses comments on a per snippet basis
    - `// @bpe.readOnlyVariables names="variableA,variableB"`
    - `// @bpe.readWriteVariables names="variableA,variableB"`

In WebSphere Business Integration Server Foundation V5.1.1 the BPEL variables were accessed indirectly with the APIs. In WebSphere Process Server V6 it is now possible to access the BPEL variables by name from the Java Snippet, introducing the possibility of a naming conflict. You should change the names of the local variables in the Java Snippets before you run the Migration Wizard.

With WebSphere Studio Application Developer Integration Edition V5.1. and WebSphere Business Integration Server Foundation 5.1.1 there is the capability of defining how the BPEL variable is to be accessed on each access, as signified by the value of the Boolean flag in the parameter list for the get methods.

With WebSphere Integration Developer V6 and WebSphere Process Server V6 the access pattern for the BPEL variables is determined once for the entire Java Snippet. The access pattern is declared in a comment as show in the slide.

During the migration process, when the boolean access qualifiers are encountered a comment is generated in the Java Snippet. If conflicting patterns are encountered, for example, read-write and read-only within the same Java Snippet, then a warning message is issued and the access is set to read-write.

After you run the Migration Wizard, check for this kind of warning message and ensure that the access method is correct for the application being migrated.

## Java Migration – Multi-valued Types

- WebSphere Studio Application Developer IE V5.1.1
  - ▶ Arrays are used to represent Multi-valued data types
- WebSphere Integration Developer V6
  - ▶ Lists are used to represent Multi-valued data types
- Conversion is managed by the Migration Wizard with the following exception.
  - ▶ When the property type of the array is a Java primitive
    - Calls that get and set the entire array cannot be mapped to the List programming model



Multi-valued data types in WebSphere Integration Developer V6 are now implemented as Lists rather than Arrays. The Migration Wizard will manage the conversion of the Java Snippet code to the List implementation except where the values in the array are Java primitives (int, long, short, byte, char, float, double and boolean)

In this case, manual migration will be required, adding a loop to wrap or unwrap the primitives to or from their corresponding Java wrapper class (Integer, Long, Short, Byte, Character, Float, Double and Boolean).

## Java Snippet Migration – Object Instantiation

- WebSphere Studio Application Developer Integration Edition V5.1.1
  - ▶ Complex types can be instantiated using the default no-argument constructor
- WebSphere Integration Developer V6
  - ▶ Complex types are instantiated using the Factory pattern.
    - An instance of the containing type can be used to instantiate sub-types.
- The Migration Wizard will attempt to make the conversion with the following caveat.
  - ▶ The order of the original instantiations must follow the parent-then-child pattern.



The mechanism for instantiating complex types has changed with the introduction of the new data model. When instantiating and initializing complex nested types in a Java Snippet the Migration Wizard will be able to make the conversion correctly if the original instantiations follow the parent-then-child pattern.

## Java Snippet Migration – Multi-Part Messages

- BPEL Variables typed by a WSDL Message
  - ▶ If there is a single message part
    - There is a one to one relationship between the message and the part.
    - The BPEL variable that is accessible in the Java Snippet will represent the message part.
      - Not the overall message
  - ▶ If multi-part messages are used then there is no direct relationship and the data object that gets created represents the message.
    - There is an extra level of indirection
    - The generated data object is more generic and requires additional programming to extract the parts from the message.
      - **commonj.sdo.DataObject**

In WebSphere Studio Application Developer Integration Edition V5.1.1 the BPEL variables are typed using WSDL messages and it is possible to use multi-part messages when defining these variables. Be aware that when multi-part messages are used for the BPEL variables in Java Snippets that the data type that the message gets converted to will be a representation of the message and not a direct representation of the data that is associated with the variable. There is an additional level of indirection that must be navigated by the programmer to get to the data values, using the SDO APIs. The WSDL message is converted to a generic SDO DataObject, of type **commonj.sdo.DataObject**.



## Agenda

- Introduction
- Overview
- Java Snippet Migration
- **Running the Migration Wizard**
- Post Migration Tasks
- Limitations
- Summary
- References

Now that the “Big Picture” has been presented and nuances of Java Snippet migration have been covered, the next section will discuss the details of running the migration wizard.

## WBI SF Migration Wizard

- The migration wizard does the following:
  - ▶ Creates a new Business Integration Module
  - ▶ Migrates the service project's classpath entries to the new module.
  - ▶ Copies all the WebSphere Business Integration Server Foundation source artifacts to the new module
  - ▶ Migrates the BPEL extensions in the WSDL files
  - ▶ Creates an SCA component for each .bpel process
  - ▶ Generates a monitoring *.mon* file for each BPEL process to preserve the default monitoring behavior from WebSphere Business Integration Server Foundation ( if necessary)



This slide outlines the steps performed by the WebSphere Business Integration Server Foundation migration wizard.

The Migration Wizard will...

1. Create a new business integration module based on the name provided by the user.
2. Migrate the classpath entries
3. Copy all the WebSphere Business Integration Server Foundation artifacts and convert the BPEL artifacts.
4. Create an SCA component for each business process.
5. Generate the monitoring files (.mon) based on the state of the 'business relevant' property in WebSphere Studio Application Developer Integration Edition V5.1.1.

If the 'business relevant' property is set to a non-default value, then the monitoring file is generated with a record indicating that the non-default behavior is being used and needs to be preserved.

## Preparing to Run the Migration Wizard

- Turn off '*automatic build*' in WebSphere Integration DeveloperV6
  - ▶ Project → Build Automatically
- Use a new WebSphere Integration DeveloperV6 workspace
- Copy the non-Service projects to the new WebSphere Integration Developer V6 workspace.
  - ▶ Copy the files to the workspace folder in the file system directory.

Before importing any artifacts, turn off the autobuild feature.

The migration process creates many new artifacts and if the autobuild feature is on, a build will be started before all the artifacts have been created. As new artifacts are created, new builds are also started. The result is a degradation in the overall build performance.

Disabling the “Build Automatically” option in WebSphere Integration Developer will speed up the migration process.

Do not open your existing WebSphere Business Integration Server Foundation V5.1.1 workspace with WebSphere Integration Developer V6. This is not supported. Open a new WebSphere Integration Developer V6 workspace and follow the procedures described in this presentation.

When importing existing files and projects into WebSphere Integration Developer, the location of the files remain unchanged, therefore it is necessary to copy the non-Service projects to the directory comprising the new WebSphere Process Server V6 workspace before doing the import. This is not necessary for the Service projects because they will be converted to new artifacts when they are imported.

## Preparing to Run the Migration Wizard (cont.)

- Shared WSDL and XSD
  - ▶ WebSphere Process Server V6 has the concept of a shared library which can be used for the common WSDL and XSD definitions.
  - ▶ In WebSphere Studio Application Developer IE the common WSDL and XSD may or may not have been factored out to a shared project.
    - If this has not been done, do this before attempting the migration
    - Create the Library module in WebSphere Integration Developer and then copy these files to the Library project folder.

With the advent of the Shared Library in WebSphere Integration Developer, there is a special case that needs to be considered. When there are common WSDL and XSD files that are to be shared across one or more Service projects, a Shared Library needs to be used. In this case the folder in the directory structure must be created first by creating the shared library in the Business Perspective of WebSphere Integration Developer. Once the folder has been created then the common WSDL and XSD files from the WebSphere Studio Application Developer Integration Edition workspace can be copied to this folder and imported into the Shared Library from that location.

If there are WSDL and XSD files that are common to one or more Service projects and they have not been factored out to a shared Java project in WebSphere Studio Application Developer Integration Edition, then you should factor and test before doing the migration.

## Running the Migration Wizard

- Import the non-Service projects
  - ▶ From the WebSphere Process Server V6 workspace
  - ▶ File → Import → Existing Project into Workspace
- Create a Business Integration (BI) Library project to hold the artifacts that are shared by the modules
  - ▶ Using WebSphere Integration Developer, create the BI Library project
  - ▶ Copy all shared artifacts from the V5.1 common java project into this new Library project folder and refresh/build the project in WebSphere Integration Developer .
- Import the Service projects
  - ▶ From the WebSphere Studio Application Developer IE workspace
  - ▶ File → Import → WebSphere Studio Application Developer Integration Edition
    - **Preserve the original BPEL Java snippets in the comments**

21

Migration – WebSphere Business Integration Server Foundation V5.1.1

© 2006 IBM Corporation

These steps are necessary and the order is important because the Migration Wizard only converts the BPEL artifacts in the Service projects and all the dependent projects and classes need to be in place before the conversions are done.

The Shared Library is a new construct that is available for managing the inter-service-project build path dependencies. The common files can be placed in a Shared Library Module and then the module dependency editor is used to define the dependency with the service projects.

Note that the files in the Shared Library are copied into the EAR of the dependent module when the J2EE EARs are created. The files are “shared by copy” at runtime.

After all the dependent projects and files have been loaded into the workspace, the service projects can then be imported, converting the BPEL artifacts along the way, using the Migration Wizard.

The Migration Wizard is invoked by using the **WebSphere Studio Application Developer Integration Edition** import type.

On the second panel of the Migration Wizard there is an option to *preserve the original BPEL Java snippets in the comments*. It is selected by default.

## Agenda

- Introduction
- Overview
- Java Snippet Migration
- Running the Migration Wizard
- **Post Migration Tasks**
- Limitations
- Summary
- References

Running the Migration Wizard is not the end. You still have work to do, errors to resolve, and wiring to complete.

## Post Migration Tasks

- Clean ( and build )
  - ▶ May turn *Autobuild* back on after the clean
- Understand and resolve all the errors.
- Wire up the Assembly Modules based on the interface/binding requirements of the application.
  - ▶ Create the imports and exports with the appropriate bindings and wire them to the business processes.
  - ▶ The Export and Import definitions will depend on the 'kind-of' transport/protocol you used for your inbound and outbound (partner links) services of your WebSphere Business Integration Server Foundation business process.
    - EJB
    - IBM Web Service (SOAP/JMS)
    - IBM Web Service (SOAP/HTTP)
    - Apache Web Service (SOAP/HTTP)
    - JMS
    - Additional outbound types include, Java Service, Transformer Service and J2C Service
- Deploy and test

Understanding and resolving all the errors can be a big task. It will depend on the application being migrated and the degree to which the known limitations can be addressed before beginning the migration process. Refer to the section on Java Snippet migration and the section on Limitations for guidance when performing this task.

With WebSphere Business Integration Server Foundation V5.1.1 there are five different ways to define the inbound and outbound partner links. The bindings are specified at the time the deployment code is generated. In the new SCA programming model the association with the partner references and the kind of binding to use, is managed in the Assembly Module with the Imports and Exports. Since there are multiple ways to migrate some of the WebSphere Studio Application Developer Integration Edition service types, the Wizard does not attempt to guess what is needed, and leaves this up to you to complete. Therefore, you need a good understanding of the WebSphere Process Server/SCA programming model before you attempt to migrate source artifacts.

## Agenda

- Introduction
- Overview
- Java Snippet Migration
- Running the Migration Wizard
- Post Migration Tasks
- **Limitations**
- Summary
- References

The limitations are discussed in this next section as a guide for resolving the errors and developing any workarounds that may be necessary.



## Limitations - General

- The Migration Wizard works with Service projects
  - ▶ Don't attempt to migrate an entire workspace at once.
- The Migration Wizard migrates BPEL source artifacts, not application binaries.
- SCA JMS Data Binding
  - ▶ There is no access to the custom JMS header properties with the standard JMS data binding.
    - A custom data binding is required for the SCA services to get access to the customer JMS header properties.

When planning for migrating from WebSphere Studio Application Developer Integration Edition and WebSphere Business Integration Server Foundations V 5.1.1 to WebSphere Integration Developer and WebSphere Process Server V6, keep in mind that the migration wizard operates on the BPEL artifacts in the service project and is not designed to migrate the entire workspace, nor does it migrate the application binaries, the EAR files.

The Extended Messaging support has also been deprecated and the standard JMS data binding does not provide access to custom JMS header properties. For the SCA services to use the JMS header properties, a custom data binding must be written.

## Limitations – Deprecated Features

- Extended Messaging is deprecated
- Business Rule Beans have been deprecated
  - ▶ Runtime support requires additional steps during the installation of WebSphere Process Server V6. There is an optional check box during the installation process
  - ▶ There is no authoring support for the old Business Rule Beans
    - To compile and test components which use Business Rule Beans in the WebSphere Integration Developer V6 test server, see the WebSphere Integration Service V6 documentation.

There is an option during WebSphere Process Server V6 installation to install support for the deprecated Business Rule Beans. If you select that option, the Business Rule Beans will run “as-is” in WebSphere Process Server V6.0.

There is no authoring support for the old Business Rule Beans. However, to compile the old Business Rule Bean artifacts in WebSphere Integration Developer V6, follow the WebSphere Integration Developer documentation to install the deprecated features on top of the embedded WebSphere Process Server 6.0 test server and then manually add the deprecated jar files to the project classpath as external jars.

## Limitations – Programming Model

- COBOL or C byte array
  - ▶ The **SDO version 1 specification** does not provide access to the COBOL or C byte array
  - ▶ This will impact applications using IMS multi-segments.
- Serialization of COBOL redefines or C unions
  - ▶ The **SDO version 1 specification** for serialization does not support COBOL redefines or C unions.
- Doc/literal wrapped WSDL style is the default with the SCA programming model
  - ▶ Does not support method overloading.
- Native support for arrays is limited
  - ▶ For creating an Array, define a WSDL interface with elements whose “maxOccurs” is greater than one.

Note the limitations with the **SDO V1 specification** in the area of COBOL and C.

When you redesign your source artifacts according to the SCA programming model, note that the **document/literal wrapped WSDL style** (which is the default style for new artifacts created using the WebSphere Integration Developer tools) **does not support method overloading**. The other WSDL styles are still supported, so you should use a WSDL style/encoding other than “document/literal wrapped” for these cases.

**Native support for arrays is limited.** In order to invoke an external service that exposes a WSDL interface with soapenc:Array types, you will need to create a WSDL interface that defines an element whose “maxOccurs” attribute is greater than one (this is the recommended approach for designing an array type).

## Limitations - BPEL

- **Multiple Replies**
  - ▶ If there are multiple replies for a particular operation in a business process and there are different client settings for any of the replies for this operation, only the first client setting will be used for migration.
    - If this condition occurs, the Migration Wizard will issue a warning.
- **BPEL Java snippets**
  - ▶ WSIFMessage metadata APIs
  - ▶ EndpointReference/EndpointReferenceType APIs
  - ▶ Complex types with duplicate names
  - ▶ Complex types with local names identical to Java classes in the java.lang package
  - ▶ BPEL variables accessed as read-only and read-write in the same snippet.
  - ▶ Many-valued primitive properties in complex types
  - ▶ Instantiation of generated classes representing complex types



Note that multiple replies for a given operation is a BPEL feature that was supported in WebSphere Studio Application Developer Integration Edition and WebSphere Business Integration Server Foundation V5.1.1 but is no longer supported in WebSphere Integration Developer and WebSphere Process Server V6. If you are migrating an application that makes use of this feature, then you will need to re-write the application.

The list of limitations related to Java Snippet migration is included here for reference. Each item has already been covered in detail, in the section on Java Snippet migration.

## Agenda

- Introduction
- Overview
- Java Snippet Migration
- Running the Migration Wizard
- Post Migration Tasks
- Limitations
- **Summary**
- References

The next section summarizes this presentation.

## Summary – prepare, migrate and fixup

- Review existing WebSphere Studio Application Developer Integration Edition V5.1.1 application and make changes based on the known limitations
  - ▶ Extra focus on Java Snippets
  - ▶ Factor common WSDLs and XSDs to a Java Utility Library.
- Turn off '*automatic build*'
- Use a new workspace
- Copy all *non-service* projects to the new workspace
- Open WebSphere Integration Developer on the new workspace
- Import the *non-service* project
  - ▶ File -> Import -> Existing Project into Workspace
  - ▶ Fix the classpath to add the JRE and WebSphere Process Server libraries
- Import all the service projects from a location outside the new workspace, using the Migration Wizard.
- Ensure that all the .wsdl and .xsd files referenced by the .bpel files are accessible in the new workspace

In summary,

Review the existing application and make whatever changes are necessary based on the known limitations. Pay particular attention to the Java Snippets and the common WSDL and XSD files. If there are common definitions used by several different service projects, then factor them out and place them in a Java Utility project which will map to a WebSphere Process Server V6 Library Module.

Start WebSphere Integration Developer in a new workspace and turn off the autobuild feature.

Copy the non-service projects to the new workspace and import them to the new workspace.

Use the Migration Wizard to import the Service Projects with the BPEL artifacts to be migrated.

And finally, address any errors and test.

## Agenda

- Introduction
- Overview
- Java Snippet Migration
- Running the Migration Wizard
- Post Migration Tasks
- Limitations
- Summary
- **References**

## Deprecated Features

- Information Center
  - There are many deprecated features to be aware of which are documented in the Information Center.

<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp>

The screenshot shows the IBM Information Center interface. On the left, a 'Contents' sidebar lists the following items:

- ibm.com: About IBM - Privacy - Contact
- WebSphere Process Server for Multiplatforms, Version 6.0**
  - Release notes
  - Product overview
  - Installing
  - Migrating to WebSphere Process Server

A yellow callout bubble points to the book icon next to 'WebSphere Process Server for Multiplatforms, Version 6.0' with the text 'Select this book'. To the right of the main content, there is a vertical list of links: 'Installing and migrating', 'Installing', 'Migrating to WebSphere Process Server'. A second yellow callout bubble points to the 'Installing and migrating' link with the text 'Then on the right, select this topic migration topic'. The footer of the page contains the text 'Migration - WebSphere Business Integration Server Foundation V5.1.1' and '© 2006 IBM Corporation'.

Deprecated features can be found in the Information Center.



## Resources

- Source Artifact Migration Tutorial – DeveloperWorks
  - ▶ [http://www.ibm.com/developerworks/websphere/library/techarticles/0509\\_ iyengar/0509\\_ iyengar.html](http://www.ibm.com/developerworks/websphere/library/techarticles/0509_ iyengar/0509_ iyengar.html)
- WebSphere Process Server/ WebSphere Integration Developer 6.0.0 Info Center – WebSphere Business Integration Server Foundation Migration section
  - ▶ <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wbit.help.migration.ui.doc/topics/tsrcartifacts.html>
- WebSphere Business Integration Server Foundation/WebSphere Studio Application Developer IE 5.1 Information Center
  - ▶ <http://publib.boulder.ibm.com/infocenter/adiehelp/index.jsp>
- WebSphere Process Server V6.0
  - ▶ <http://www.ibm.com/software/integration/wps/>
- WebSphere Integration Developer V6.0
  - ▶ <http://www.ibm.com/software/integration/wid/>
- WebSphere Business Integration Server Foundation V5.1
  - ▶ <http://www.ibm.com/software/integration/wbisf/>
- WebSphere Studio Application Developer Integration Edition V5.1
  - ▶ <http://www.ibm.com/software/integration/wsadie/>
- BPEL4WS 1.1 Specification (WebSphere Business Integration Server Foundation supported BPEL spec)
  - ▶ <http://www.ibm.com/developerworks/webservices/library/specification/ws-bpel/>
- WS-BPEL 2.0 Specification Draft (WebSphere Process Server supports a subset)
  - ▶ <http://www.oasis-open.org/committees/download.php/10347/wsbpel-specification-draft-120204.htm>

## Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)/business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

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

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2005,2006. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.