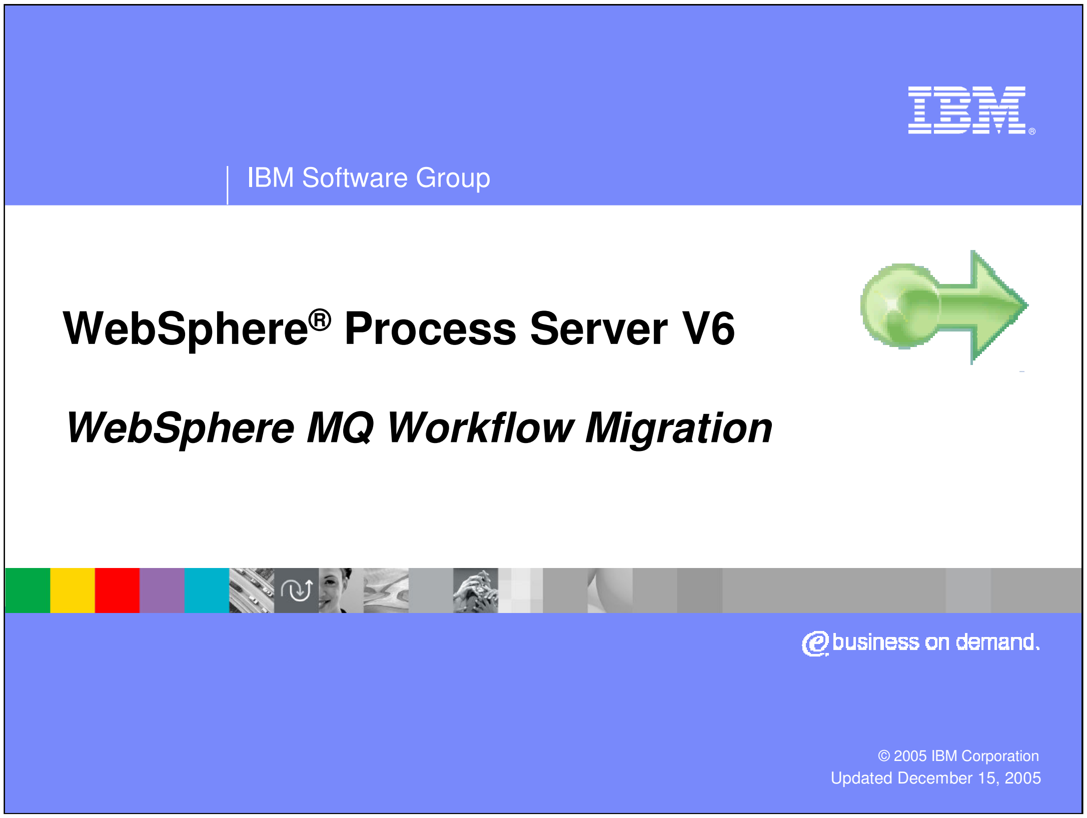


IBM Software Group

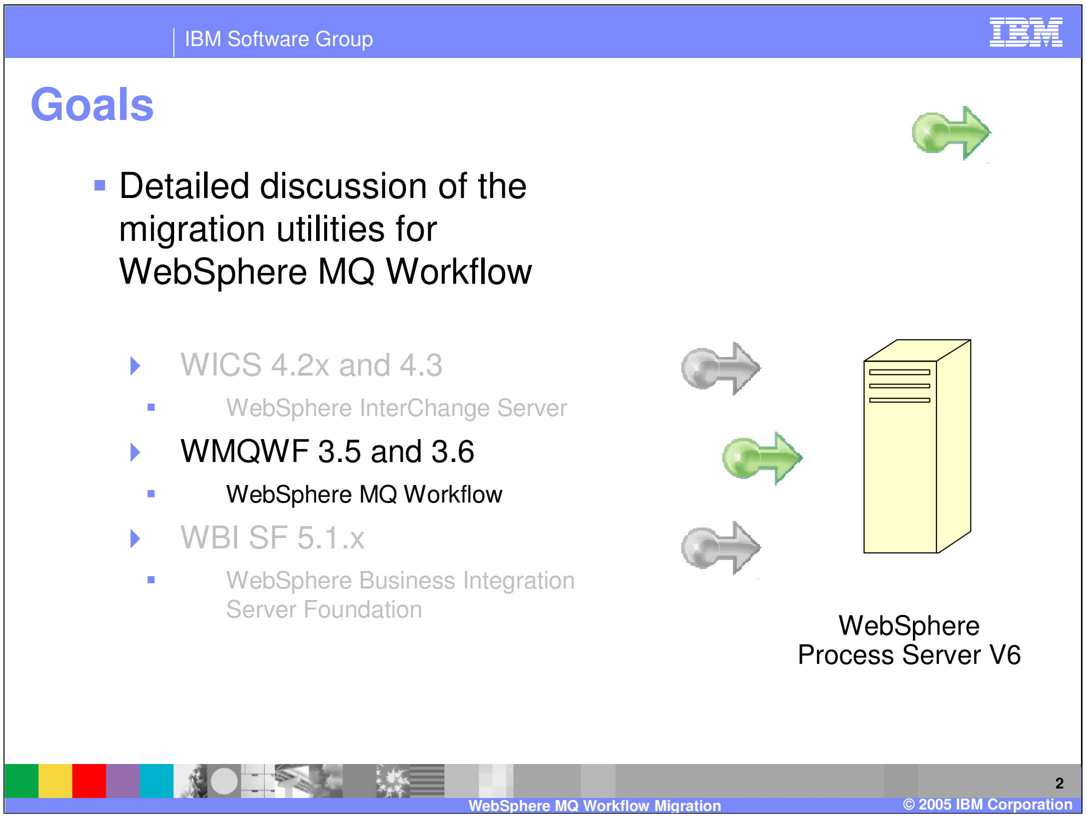
WebSphere® Process Server V6

WebSphere MQ Workflow Migration

business on demand.

Updated December 15, 2005

This presentation will cover migrating to WebSphere Process Server V6.0 from WebSphere MQ Workflow



IBM Software Group

Goals

- Detailed discussion of the migration utilities for WebSphere MQ Workflow
  - WICS 4.2x and 4.3
    - WebSphere InterChange Server
  - WMQWF 3.5 and 3.6
    - WebSphere MQ Workflow
  - WBI SF 5.1.x
    - WebSphere Business Integration Server Foundation

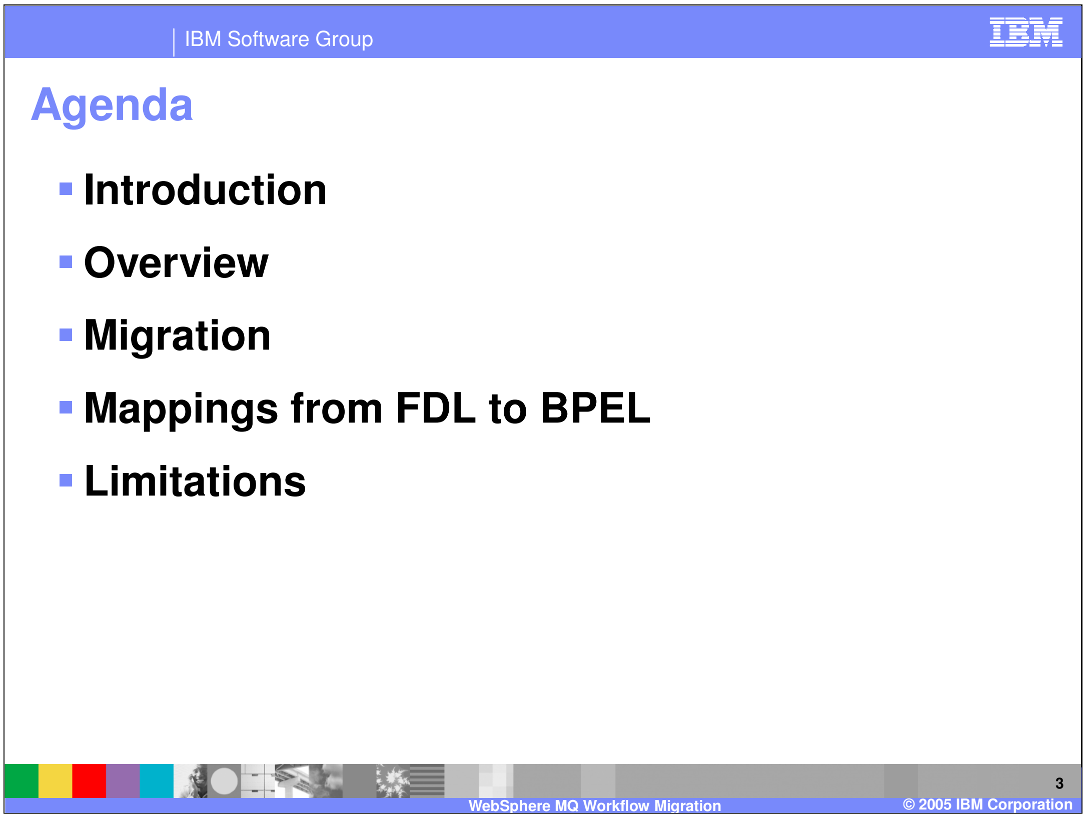
2

WebSphere MQ Workflow Migration

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 migrating from WebSphere MQ Workflow.



Agenda

- Introduction
- Overview
- Migration
- Mappings from FDL to BPEL
- Limitations

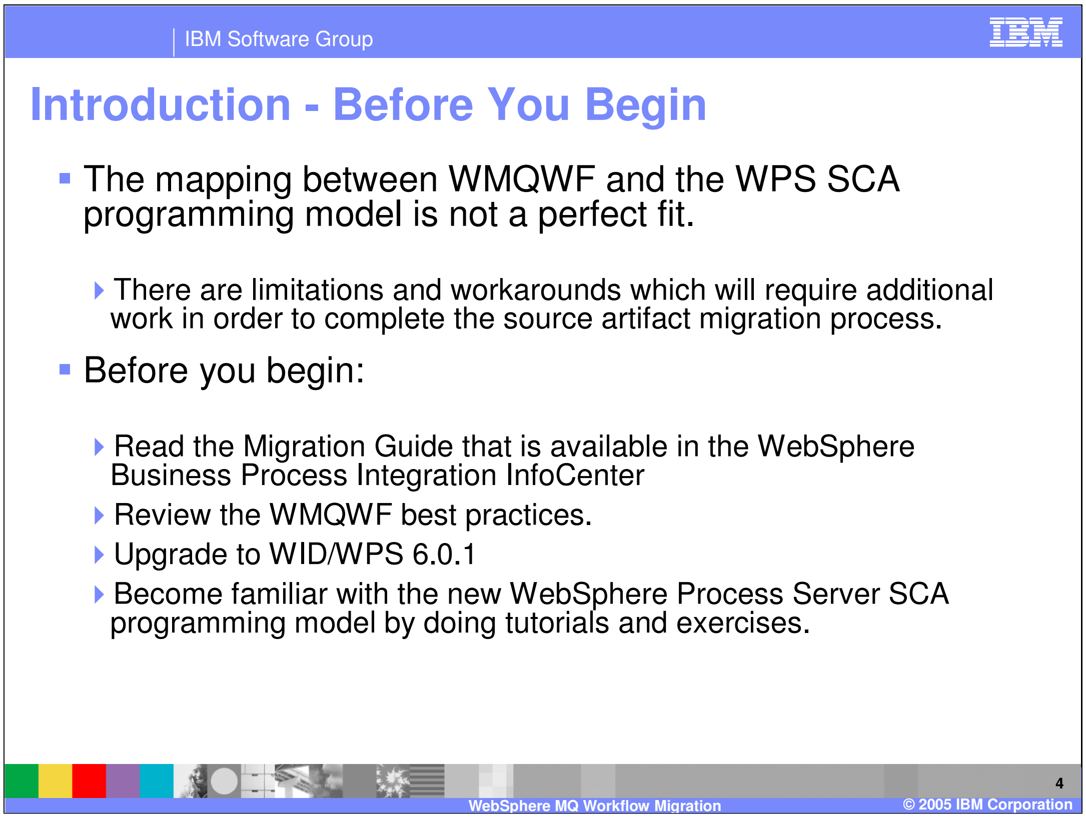
The agenda for this presentation is to focus on the steps involved in migrating from WebSphere MQ Workflow to WebSphere Process Server V6.

It begins with an introduction of background material and tasks that should be done before starting a migration effort.

Followed by a high level overview of the migration process.

Migration options and details will then be discussed with detailed information about the artifact mapping from WMQWF to WPS/SCA

Concluding with a section on the known limitations and how they impact the migration process.



IBM Software Group

Introduction - Before You Begin

- The mapping between WMQWF and the WPS SCA programming model is not a perfect fit.
  - There are limitations and workarounds which will require additional work in order to complete the source artifact migration process.
- Before you begin:
  - Read the Migration Guide that is available in the WebSphere Business Process Integration InfoCenter
  - Review the WMQWF best practices.
  - Upgrade to WID/WPS 6.0.1
  - Become familiar with the new WebSphere Process Server SCA programming model by doing tutorials and exercises.

4

WebSphere MQ Workflow Migration

© 2005 IBM Corporation

Before you begin a migration, you should become familiar with the known limitations identified at the end of this presentation and be prepared to perform some post migration tasks.

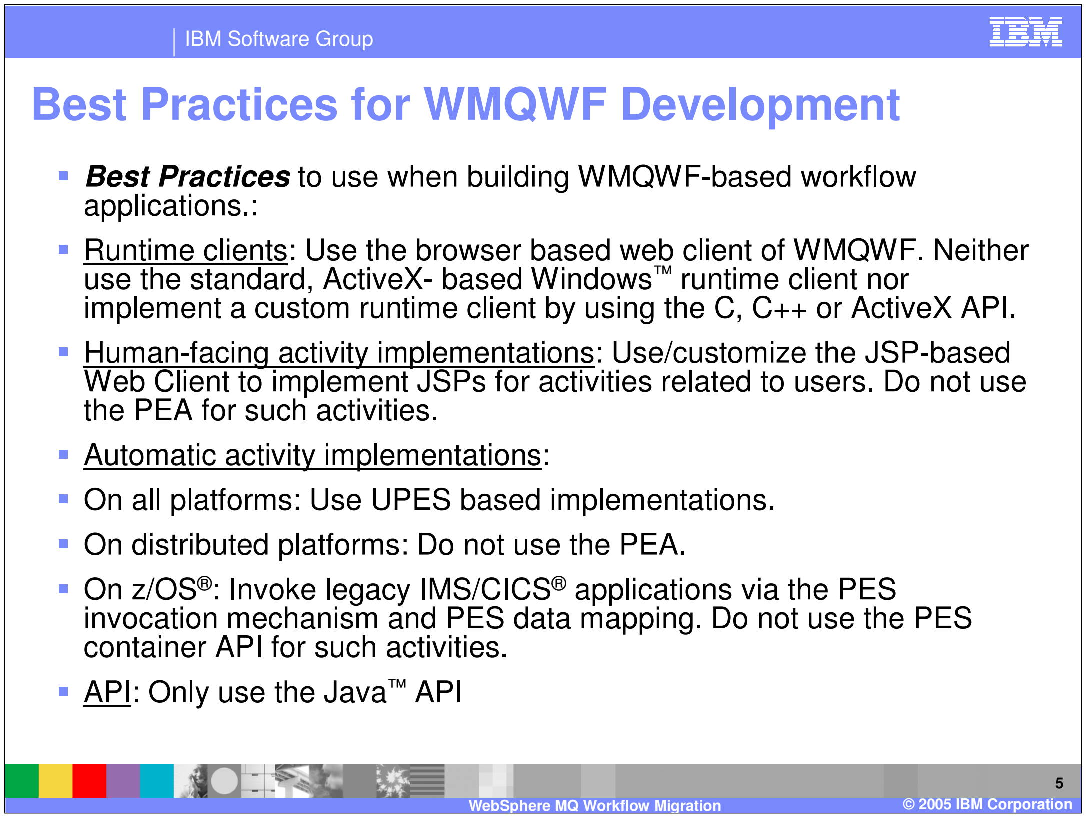
You should also:

Read the Migration Guide.

Review the Best Practice for WMQWF development

Upgrade to WID/WPS 6.0.1

Become familiar with the SCA programming model



Best Practices for WMQWF Development

- ***Best Practices*** to use when building WMQWF-based workflow applications.:
- Runtime clients: Use the browser based web client of WMQWF. Neither use the standard, ActiveX- based Windows™ runtime client nor implement a custom runtime client by using the C, C++ or ActiveX API.
- Human-facing activity implementations: Use/customize the JSP-based Web Client to implement JSPs for activities related to users. Do not use the PEA for such activities.
- Automatic activity implementations:
- On all platforms: Use UPES based implementations.
- On distributed platforms: Do not use the PEA.
- On z/OS®: Invoke legacy IMS/CICS® applications via the PES invocation mechanism and PES data mapping. Do not use the PES container API for such activities.
- API: Only use the Java™ API

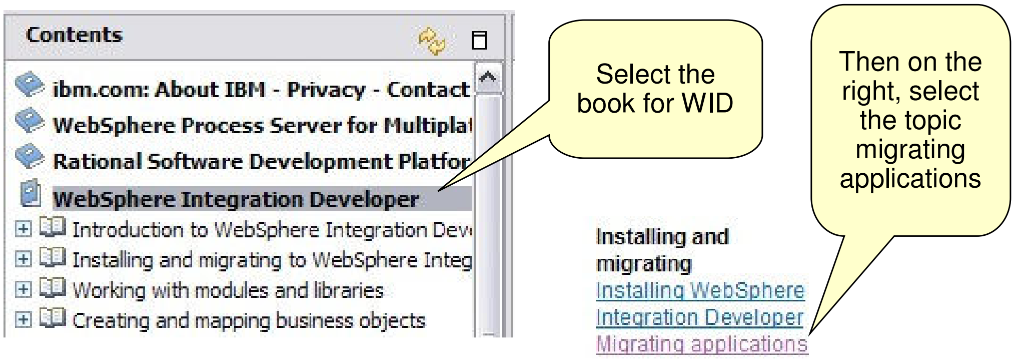
The best practices listed here have been published for current WMQWF development as guidelines for mitigating potential problems when migrating to WPS V6.

It is recommended that you take a moment to review these ‘best practices’.

Introduction - References

- Information Center

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



Contents

- ibm.com: About IBM - Privacy - Contact
- WebSphere Process Server for Multiplat
- Rational Software Development Platfor
- WebSphere Integration Developer
  - ⊞ Introduction to WebSphere Integration Deve
  - ⊞ Installing and migrating to WebSphere Integ
  - ⊞ Working with modules and libraries
  - ⊞ Creating and mapping business objects

Select the book for WID

Installing and migrating

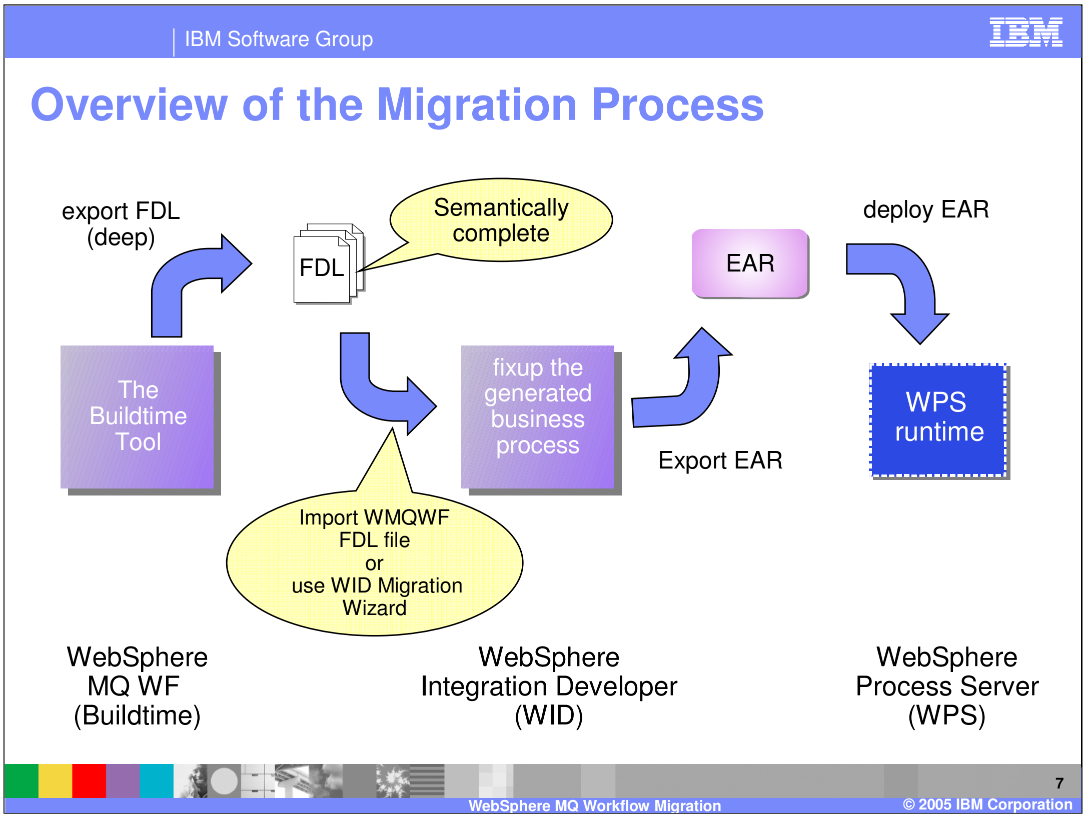
Installing WebSphere Integration Developer

Migrating applications

Then on the right, select the topic 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.



IBM Software Group

Overview of the Migration Process

export FDL (deep)

FDL

Semantically complete

The Buildtime Tool

Import WMQWF FDL file or use WID Migration Wizard

fixup the generated business process

EAR

Export EAR

deploy EAR

WPS runtime

WebSphere MQ WF (Buildtime)

WebSphere Integration Developer (WID)

WebSphere Process Server (WPS)

7

WebSphere MQ Workflow Migration

© 2005 IBM Corporation

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

export from the source system, import to the target system, edit, resolve errors and tune the migrated artifacts for BPEL / SCA and then deploy the application to the WPS V6 runtime.

Beginning with the Export:

•Export the FDL from the WMQWF environment using the WMQWF Buildtime Tool.

•This requires a semantically complete FDL
That is to say use, **Export DEEP** from the MQWF Buildtime

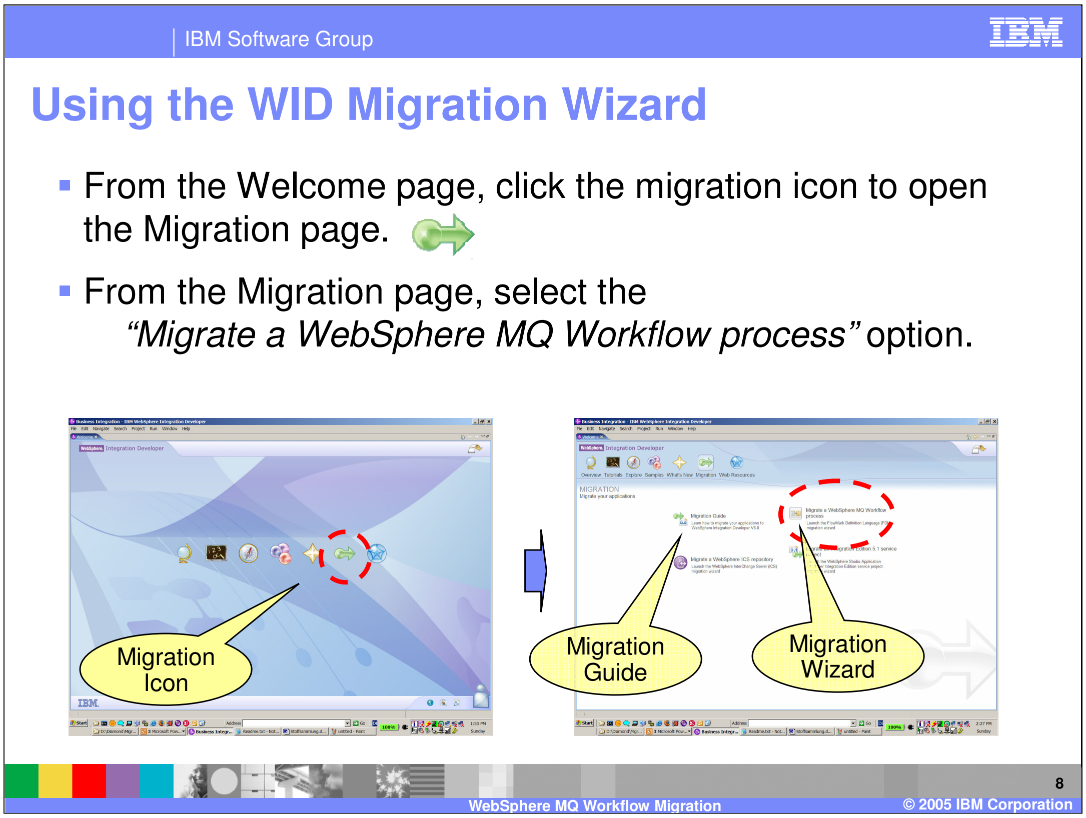
•Import the FDL into WebSphere Integration Developer V6 (WID) using the Migration Wizard or File Import

•In WID, fix the Business Processes to eliminate any errors based on the errors identified and the known limitations.

•Export the EAR and deploy it to the WebSphere Process Server V6 runtime (WPS)

The WID Migration Wizard will call the import utility under the covers which in turn will call the import utility, which will call the FDL2BPEL utility.

The FDL2BPEL utility will generate the new artifacts which will be loaded into the appropriate WPS modules.



Using the WID Migration Wizard

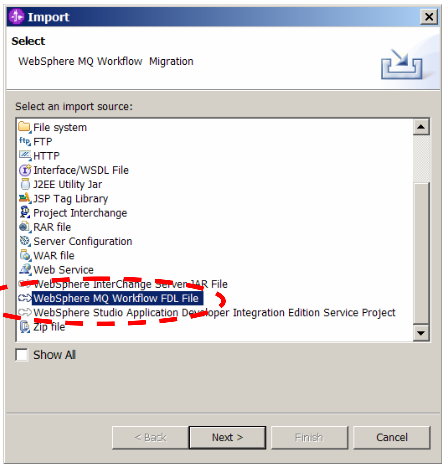
- From the Welcome page, click the migration icon to open the Migration page.
- From the Migration page, select the *“Migrate a WebSphere MQ Workflow process”* option.

Using the Migration Wizard that comes with WID V6 is the easiest way to get started. The Migration Wizard is launched from the Welcome page. To locate the Welcome screen in WebSphere Integration Developer, use the menu bar, **Help → Welcome**.

The Migration Guide as well as the WMQWF Migration Wizard are available from the Migration page.

Using Import

- You can also open the Migration wizard from the WID "*Business Integration Perspective*" by clicking:
  - > **File** > **Import** > **WebSphere MQ Workflow FDL file**



Import

Select

WebSphere MQ Workflow Migration

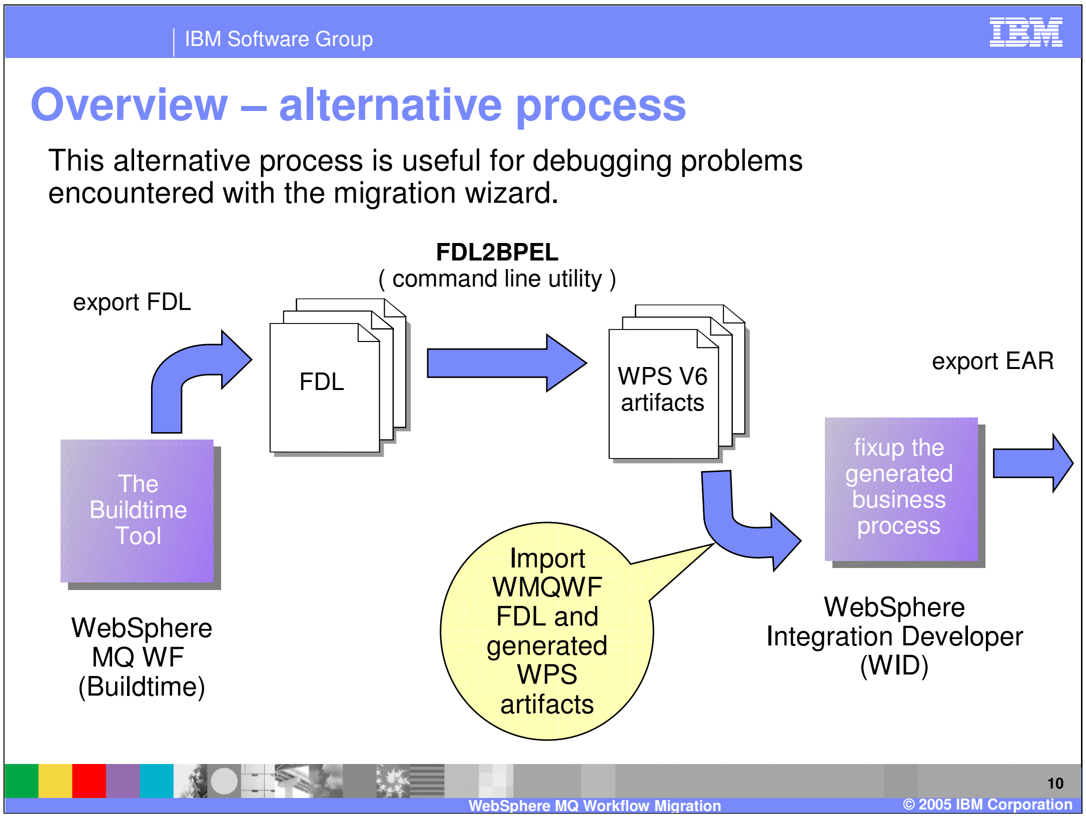
Select an import source:

File system
FTP
HTTP
Interface/WSDL File
J2EE Utility Jar
JSP Tag Library
Project Interchange
RAR file
Server Configuration
WAR file
Web Service
WebSphere InterChange Server JAR File
WebSphere MQ Workflow FDL File
WebSphere Studio Application Developer Integration Edition Service Project
Zip file

□ Show All

The Import function can also be used directly to invoke the Migration Wizard.

WID V6 has a special import type for the WMQWF FDL file that will recognize the WMQWF artifacts and launch the Migration Wizard, which will convert them to the appropriate WPS / BPEL artifacts as they are imported.



IBM Software Group

Overview – alternative process

This alternative process is useful for debugging problems encountered with the migration wizard.

FDL2BPEL
(command line utility)

export FDL

FDL

WPS V6 artifacts

export EAR

The Buildtime Tool

fixup the generated business process

Import WMQWF FDL and generated WPS artifacts

WebSphere MQ WF (Buildtime)

WebSphere Integration Developer (WID)

10

WebSphere MQ Workflow Migration

© 2005 IBM Corporation

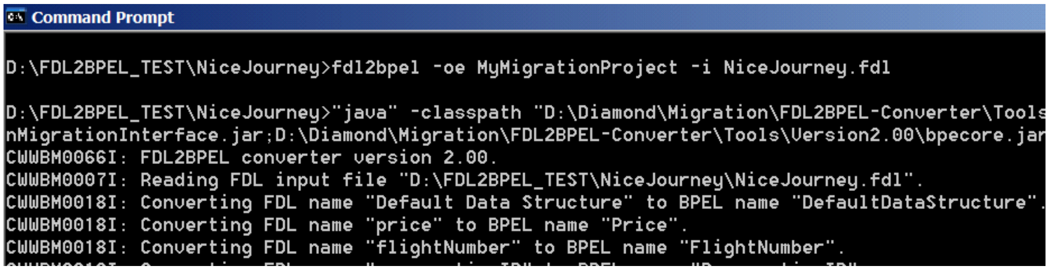
Alternatively, you can download the WMQWF support pak WA73, which contains additional documentation and the FDL2BPEL utility.

Using the FDL2BPEL utility directly from the command line with the FDL as input, the new source artifacts can be generated and then imported into the appropriate WPS V6 modules.

Introducing this extra step can be useful when problems are encountered using the WID migration utility, as the error reporting is more extensive with the command line tool.

Using FDL2BPEL: Generating Artifacts

- Download support pak WA73
  - Documentation and FDL2BPEL
- Run the FDL2BPEL utility from the command line, using the exported FDL file as input.



Command Prompt

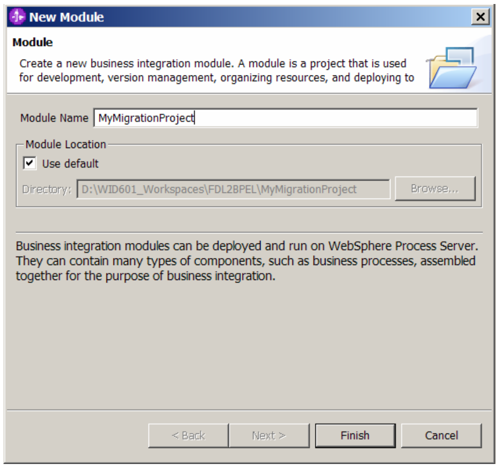
```
D:\FDL2BPEL_TEST\NiceJourney>fdl2bpel -oe MyMigrationProject -i NiceJourney.fdl

D:\FDL2BPEL_TEST\NiceJourney>"java" -classpath "D:\Diamond\Migration\FDL2BPEL-Converter\Tools
nMigrationInterface.jar;D:\Diamond\Migration\FDL2BPEL-Converter\Tools\Version2.00\bpecore.jar
CWWBM0066I: FDL2BPEL converter version 2.00.
CWWBM0007I: Reading FDL input file "D:\FDL2BPEL_TEST\NiceJourney\NiceJourney.fdl".
CWWBM0018I: Converting FDL name "Default Data Structure" to BPEL name "DefaultDataStructure".
CWWBM0018I: Converting FDL name "price" to BPEL name "Price".
CWWBM0018I: Converting FDL name "flightNumber" to BPEL name "FlightNumber".
```

The FDL2BPEL command line utility can be downloaded from the IBM Support site. The URL is available in the references section at the end of this presentation.

Using the FDL2BPEL command line utility provides more control and error reporting, which is beneficial when debugging migration problems.

Using FDL2BPEL: Create a New Module

- From the Business Integration Perspective, create a new module
  - **File → New →Project → Module**
  - make the module name the same as the value used for the command line parameter, **-oe**



Create a new business integration module. A module is a project that is used for development, version management, organizing resources, and deploying to

Module Name: MyMigrationProject

Module Location

☑ Use default

Directory: D:\WID601_Workspaces\FDL2BPEL\MyMigrationProject

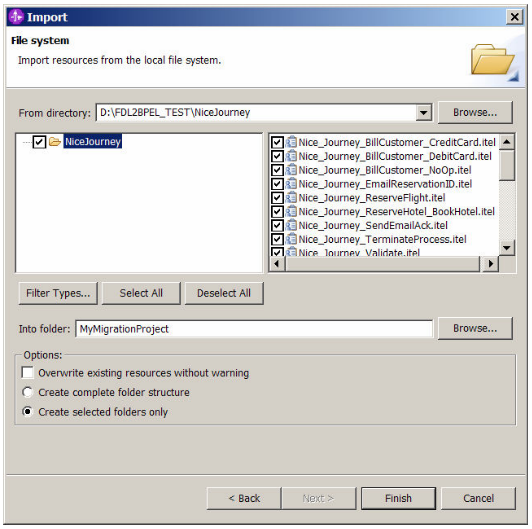
Business integration modules can be deployed and run on WebSphere Process Server. They can contain many types of components, such as business processes, assembled together for the purpose of business integration.

After the new artifacts have been generated using the FDL2BPEL utility, the destination module must be manually created in the target workspace.

Making the module name the same as the name used when generating the artifacts will reduce the amount of post migration rework.

Using FDL2BPEL: Import the Generated Artifacts

- From the Resources Perspective, select the project folder and import the generated files.
  - **File → Import → File System**



Import

File system

Import resources from the local file system.

From directory: D:\FDL2BPEL_TEST\NiceJourney

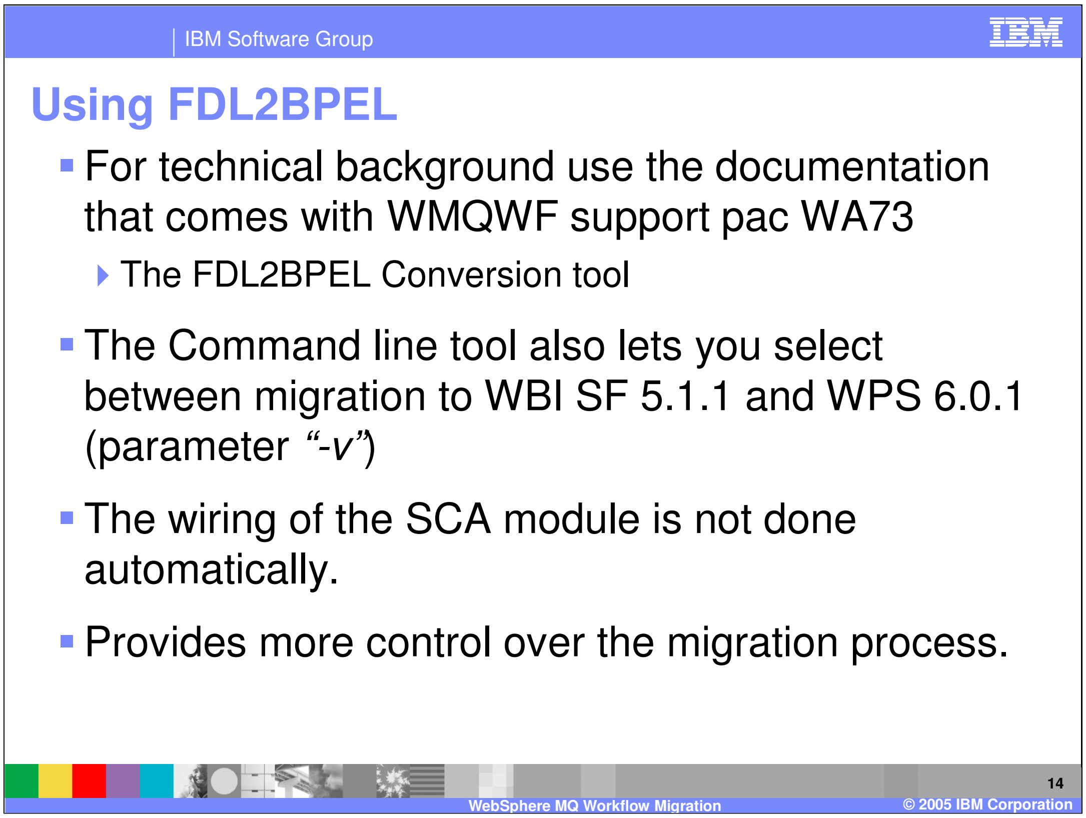
☑ NiceJourney

☑ Nice_Journey_BillCustomer_CreditCard.itel
☑ Nice_Journey_BillCustomer_DebitCard.itel
☑ Nice_Journey_BillCustomer_NoOp.itel
☑ Nice_Journey_EmailReservationID.itel
☑ Nice_Journey_ReserveFlight.itel
☑ Nice_Journey_ReserveHotel_BookHotel.itel
☑ Nice_Journey_SendEmailAck.itel
☑ Nice_Journey_TerminateProcess.itel
☑ Nice_Journey_Validate.itel

Into folder: MyMigrationProject

Options:
☐ Overwrite existing resources without warning
○ Create complete folder structure
● Create selected folders only

The next step is to import the generated artifacts into the newly created module and workspace.



IBM Software Group

Using FDL2BPEL

- For technical background use the documentation that comes with WMQWF support pac WA73
  - The FDL2BPEL Conversion tool
- The Command line tool also lets you select between migration to WBI SF 5.1.1 and WPS 6.0.1 (parameter *“-v”*)
- The wiring of the SCA module is not done automatically.
- Provides more control over the migration process.

14

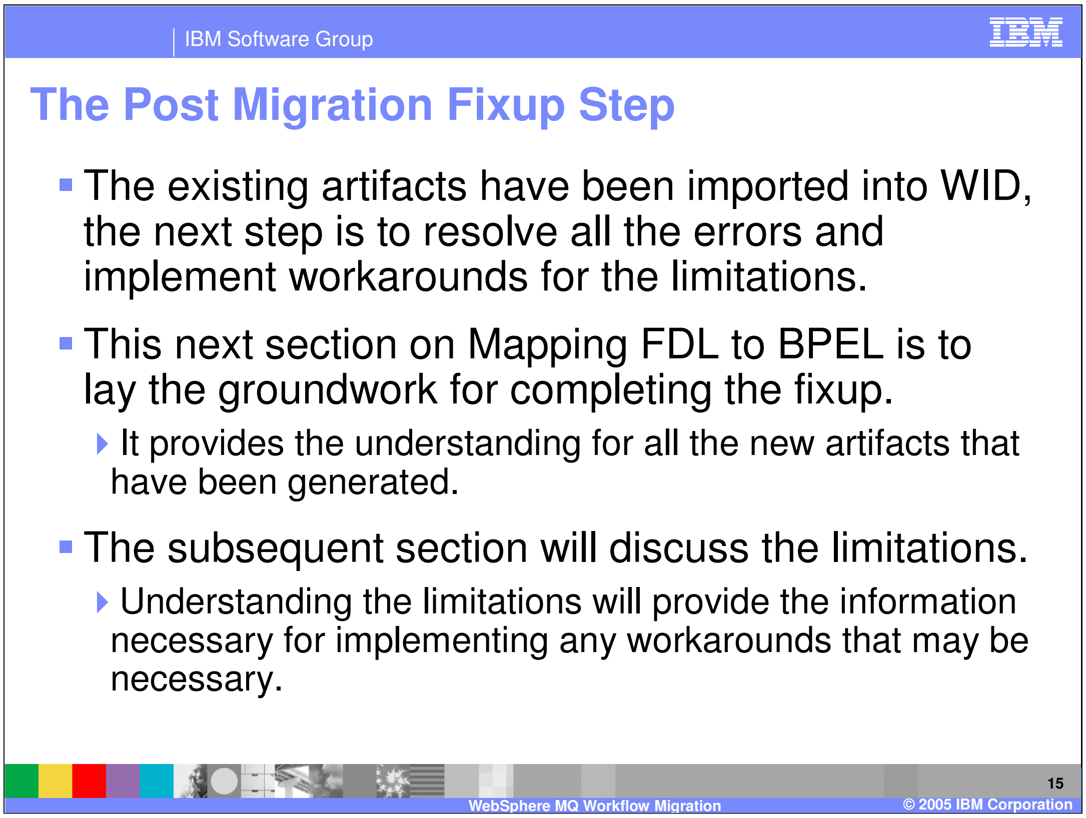
WebSphere MQ Workflow Migration

© 2005 IBM Corporation

The WMQWF support pac WA73 is provided to support WMQWF users and is from the perspective of WMQWF.

The documentation that is provided with the support pak is more complete than that available in WPS V6 Information Center and will provide the technical background that will be familiar to WMQWF users.

Note that the wiring of the module assembly will be part of the post migration work.



The Post Migration Fixup Step

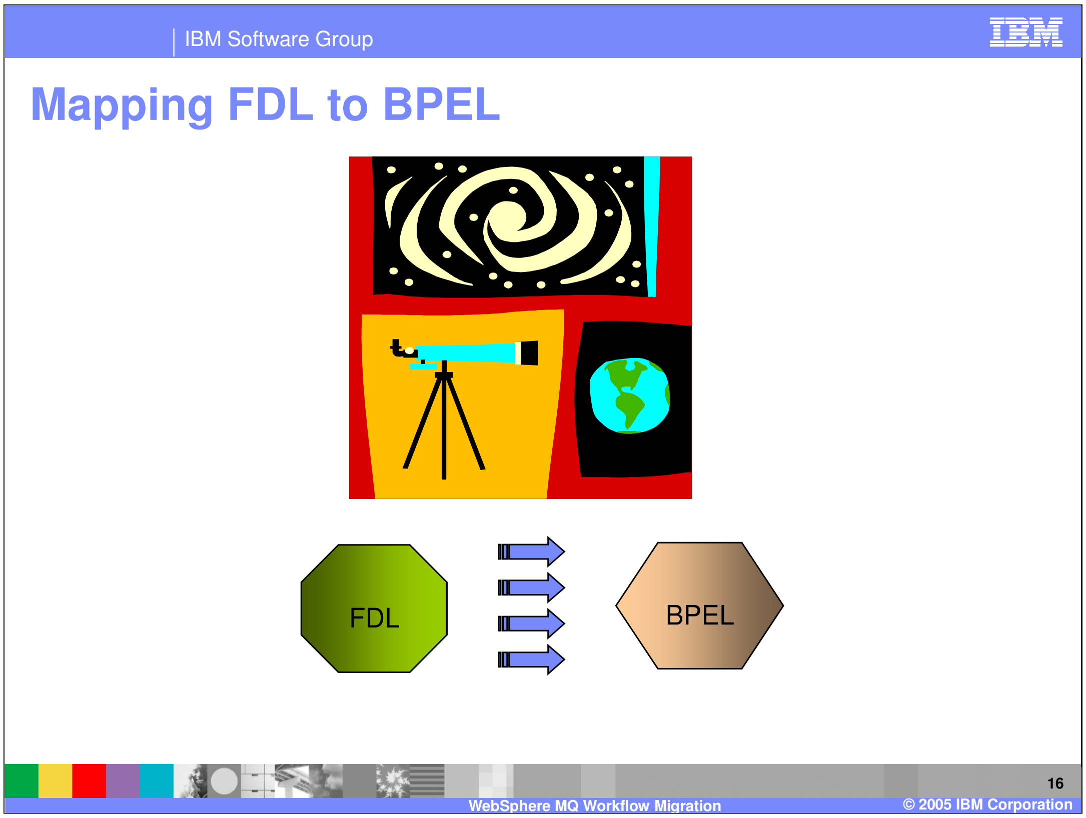
- The existing artifacts have been imported into WID, the next step is to resolve all the errors and implement workarounds for the limitations.
- This next section on Mapping FDL to BPEL is to lay the groundwork for completing the fixup.
  - It provides the understanding for all the new artifacts that have been generated.
- The subsequent section will discuss the limitations.
  - Understanding the limitations will provide the information necessary for implementing any workarounds that may be necessary.

At this point all of the WMQWF artifacts that can be converted, have been converted to WPS V6 BPEL / SCA artifacts.

However, not all artifacts can be converted and there will be errors to correct and workarounds to implement.

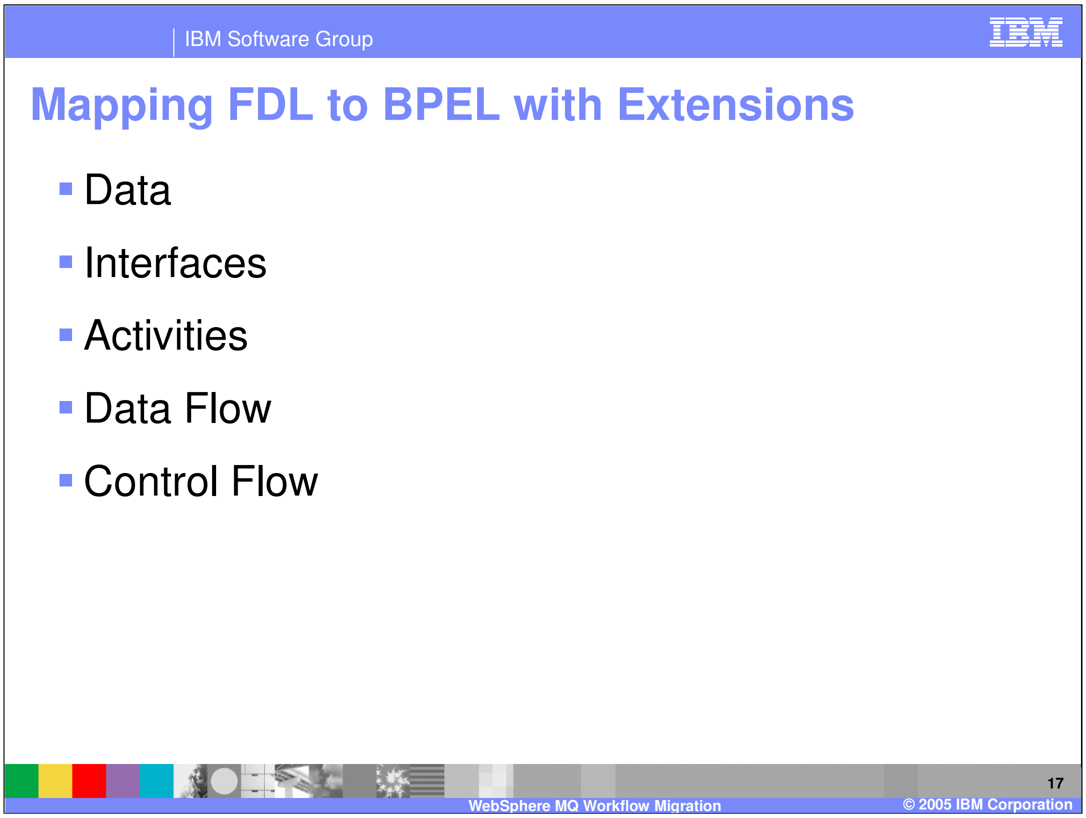
The remaining part of this presentation will focus on what does get mapped and then highlight the limitations.

This will provide the information needed to complete the migration so that the application can be deployed to the WPS V6 runtime.



Mapping FDL to BPEL

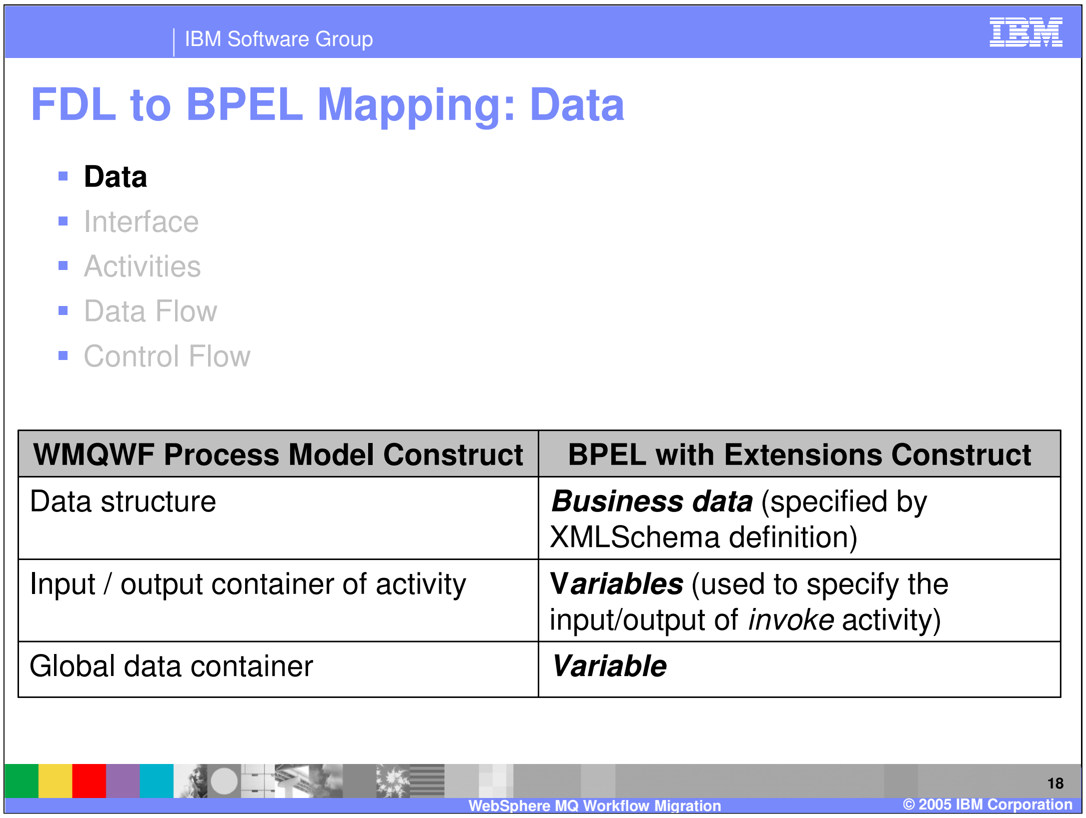
This section will cover the mapping of Flow Definition Language (FDL) to Business Process Execution Language (BPEL).



Mapping FDL to BPEL with Extensions

- Data
- Interfaces
- Activities
- Data Flow
- Control Flow

Each category listed here represents a kind of FDL artifact that must be considered when migrating from FDL to BPEL.



IBM Software Group

FDL to BPEL Mapping: Data

- **Data**
- Interface
- Activities
- Data Flow
- Control Flow

WMQWF Process Model Construct	BPEL with Extensions Construct
Data structure	***Business data*** (specified by XMLSchema definition)
Input / output container of activity	***Variables*** (used to specify the input/output of *invoke* activity)
Global data container	***Variable***

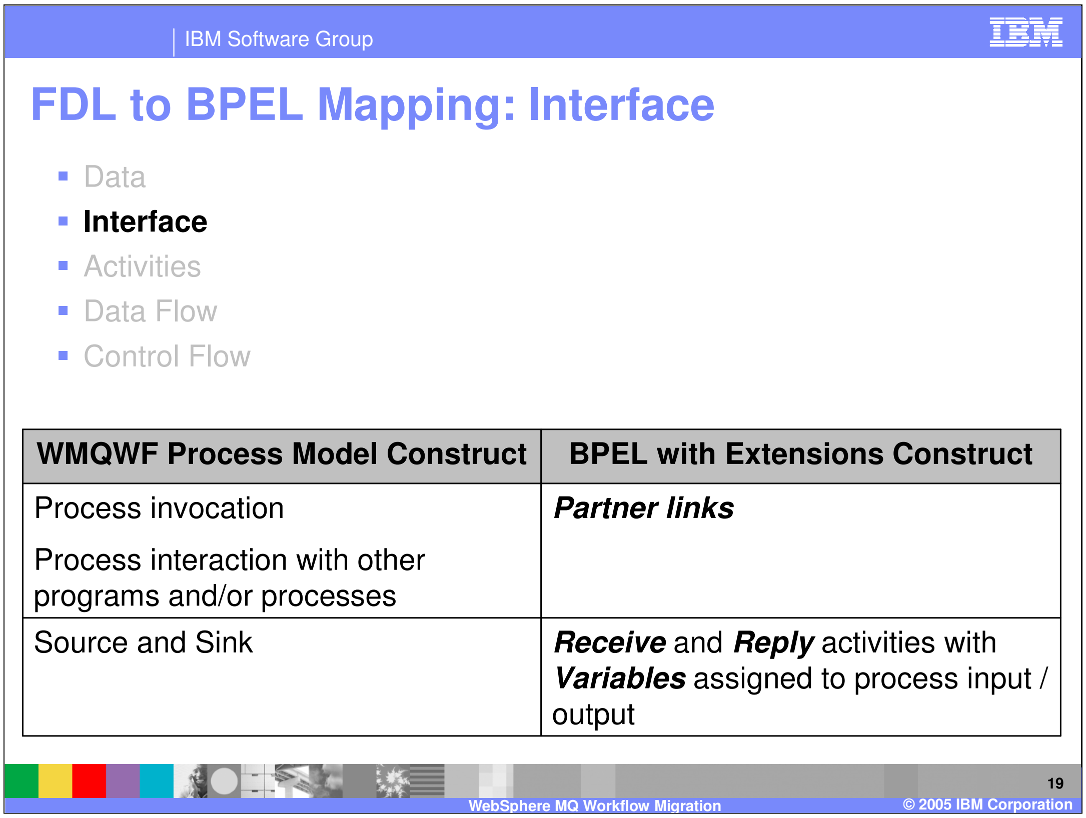
18

WebSphere MQ Workflow Migration

© 2005 IBM Corporation

In BPEL / SCA, data are represent by XML Schema and XSDs and variables are used to hold the data values and facilitate the movement of data from one activity to another.

The WMQWF data container can be mapped directly to the BPEL variables.



IBM Software Group

FDL to BPEL Mapping: Interface

- Data
- **Interface**
- Activities
- Data Flow
- Control Flow

WMQWF Process Model Construct	BPEL with Extensions Construct
Process invocation Process interaction with other programs and/or processes	***Partner links***
Source and Sink	***Receive*** and ***Reply*** activities with ***Variables*** assigned to process input / output

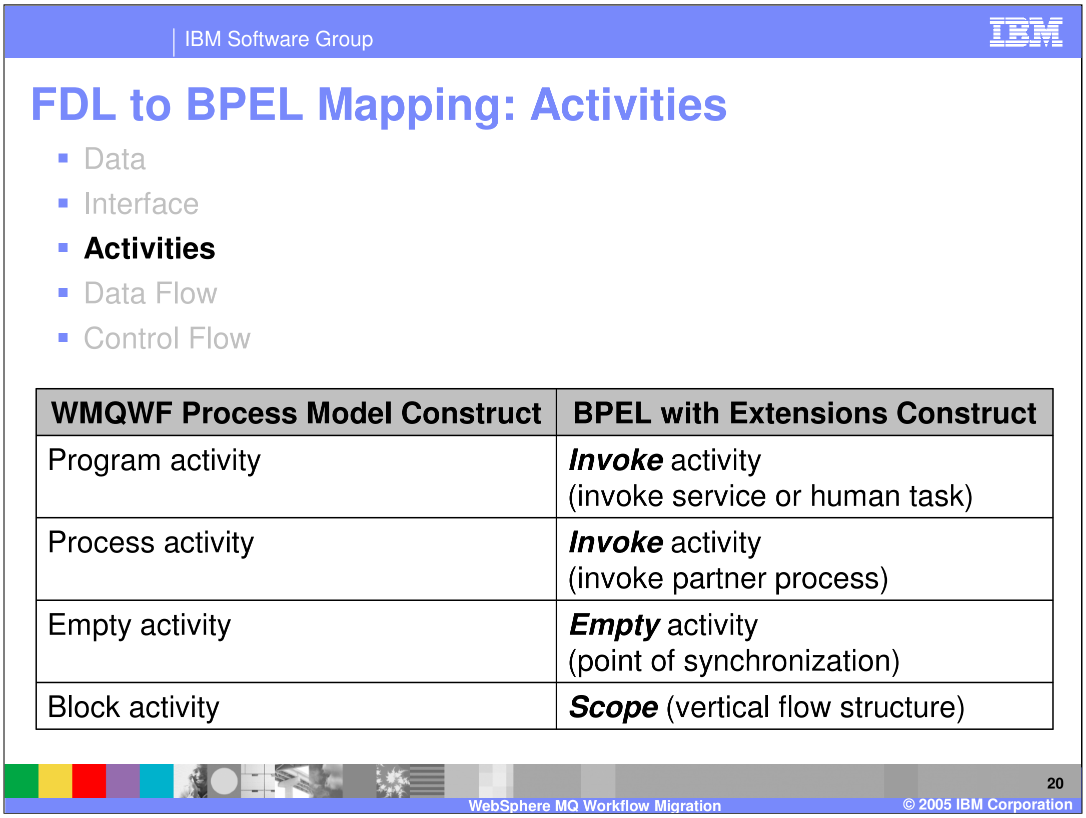
19

WebSphere MQ Workflow Migration

© 2005 IBM Corporation

With BPEL, the interfaces are defined using WSDL, which in turn defines interactions with other services in terms of Partner Links.

With WMQWF, the external interface to the business process is called a **Source** and the external interface for the output is call the **Sink**. BPEL provides analogous constructs called **Receive** for input and **Reply** for output. Its very natural to map the Source to the Receive and the Sink to the Reply.

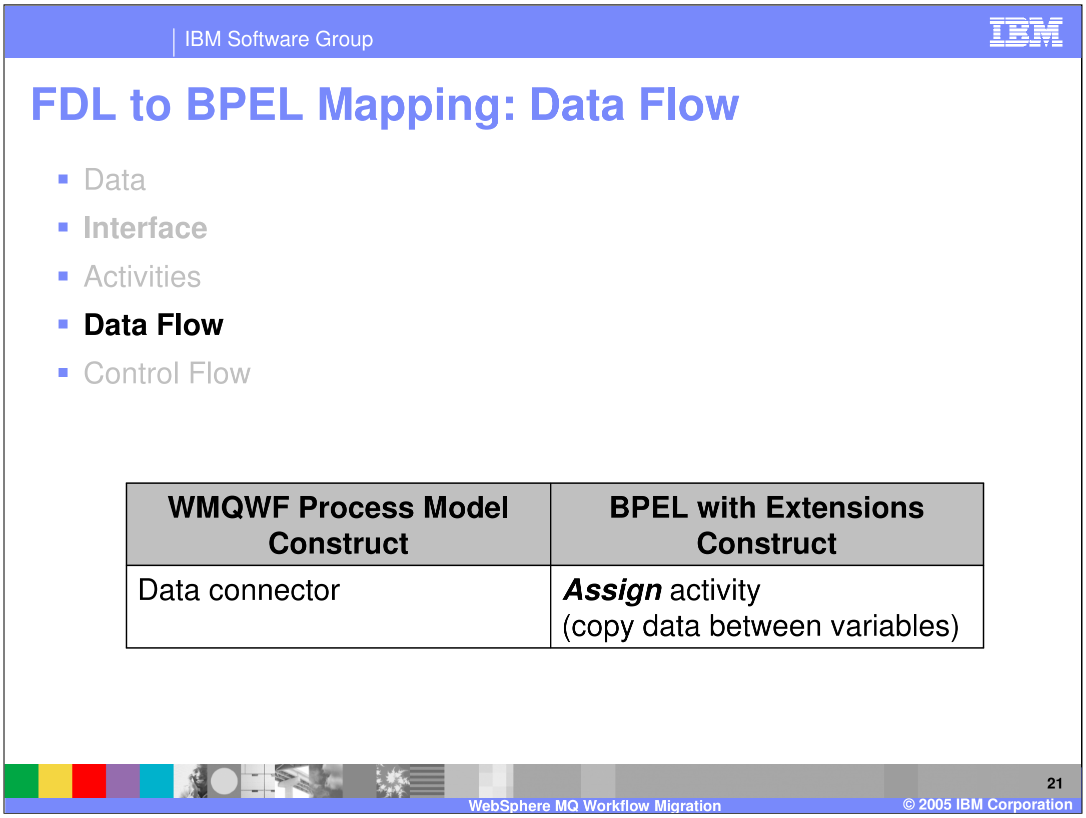


FDL to BPEL Mapping: Activities

- Data
- Interface
- **Activities**
- Data Flow
- Control Flow

WMQWF Process Model Construct	BPEL with Extensions Construct
Program activity	***Invoke*** activity (invoke service or human task)
Process activity	***Invoke*** activity (invoke partner process)
Empty activity	***Empty*** activity (point of synchronization)
Block activity	***Scope*** (vertical flow structure)

For each of the four basic activities in WMQWF there are corresponding constructs in BPEL. The mapping is relatively straightforward based on classification rules, which will be discussed in subsequent slides.

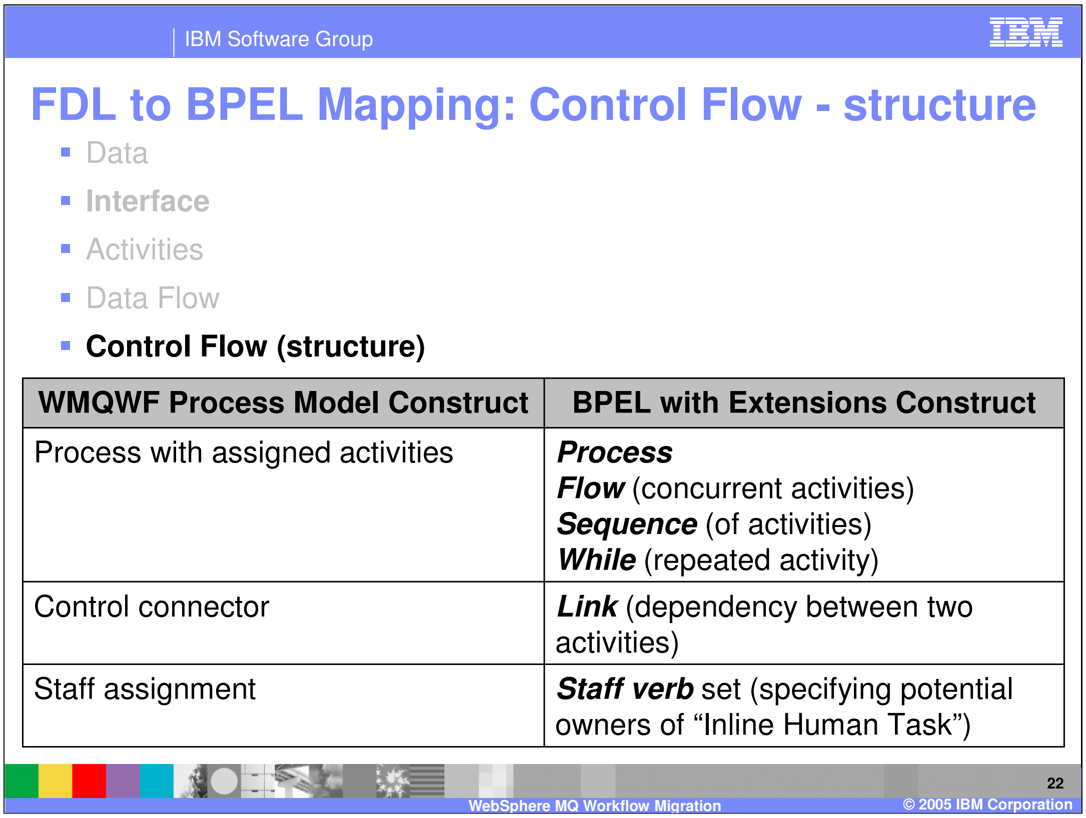


FDL to BPEL Mapping: Data Flow

- Data
- Interface
- Activities
- **Data Flow**
- Control Flow

WMQWF Process Model Construct	BPEL with Extensions Construct
Data connector	***Assign*** activity (copy data between variables)

Data flow is one of those constructs in WMQWF for which there is no analog in BPEL. The movement of data in BPEL is managed by assigning data to variables, which are then used by the activities. When a data connector is encountered in the WMQWF model, Assign activities are created and inserted into the BPEL model at the appropriate places. An example is provided later in this presentation.



IBM Software Group

FDL to BPEL Mapping: Control Flow - structure

- Data
- **Interface**
- Activities
- Data Flow
- **Control Flow (structure)**

WMQWF Process Model Construct	BPEL with Extensions Construct
Process with assigned activities	***Process*** ***Flow*** (concurrent activities) ***Sequence*** (of activities) ***While*** (repeated activity)
Control connector	***Link*** (dependency between two activities)
Staff assignment	***Staff verb*** set (specifying potential owners of “Inline Human Task”)

22

WebSphere MQ Workflow Migration

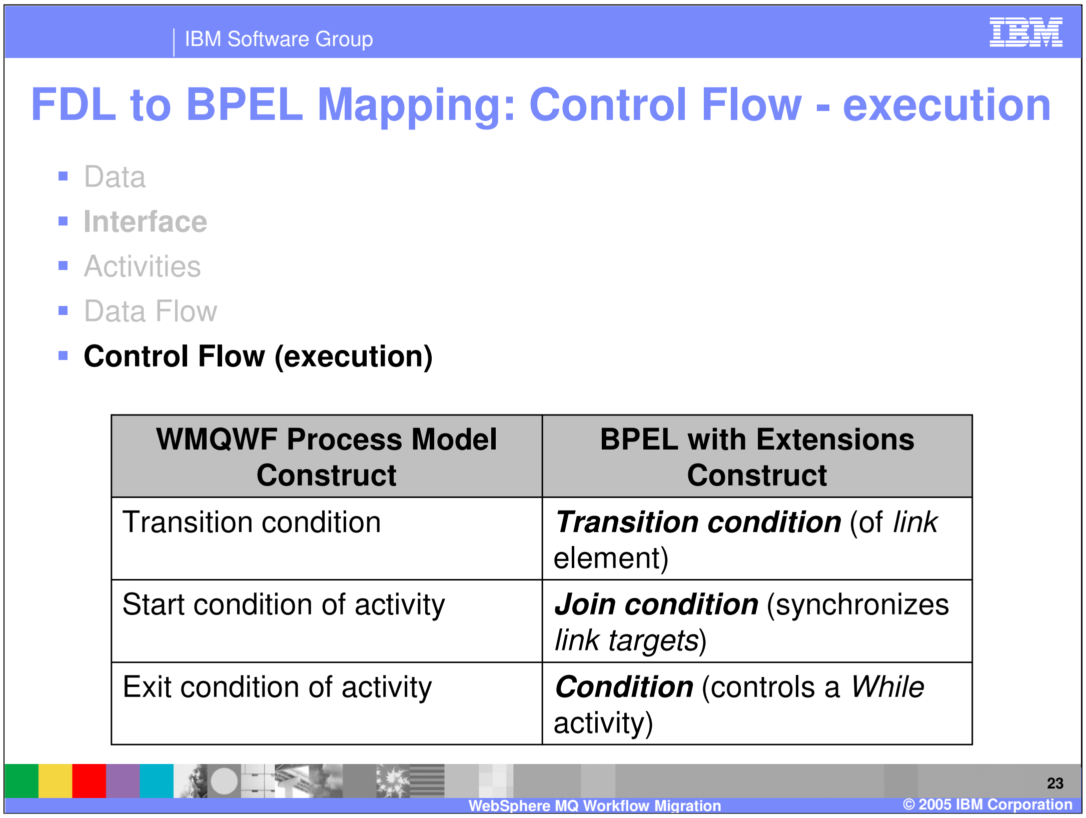
There are two aspects to control flow, one is structure and the other is execution. When considering the **structural** aspect of control flow in a WMQWF business process, there are elements such as assigned activities, connectors and staff assignments.

WMQWF provides only one basic kind of process and there are no ‘structuring’ elements like the ones available in BPEL. With BPEL, there is the “Flow”, which allows for grouping activities that will be run concurrently in a collapsible container with a distinct scope. There is also the “Sequence”, which provides a collapsible container for activities that will run sequentially.

The “while” construct is another feature that BPEL handles differently than WMQWF. The “while” construct provides a way of grouping activities that will be run repeatedly while waiting for some condition to be met. WMQWF has a more generic approach, using a condition on the activity.

Therefore, based on the properties of the WMQWF activity, a BPEL Flow, Sequence or While will be created.

The Link is the BPEL way of connecting two or more activities and the management of the Staff assignment is done in the Verb of the Human Task activity.

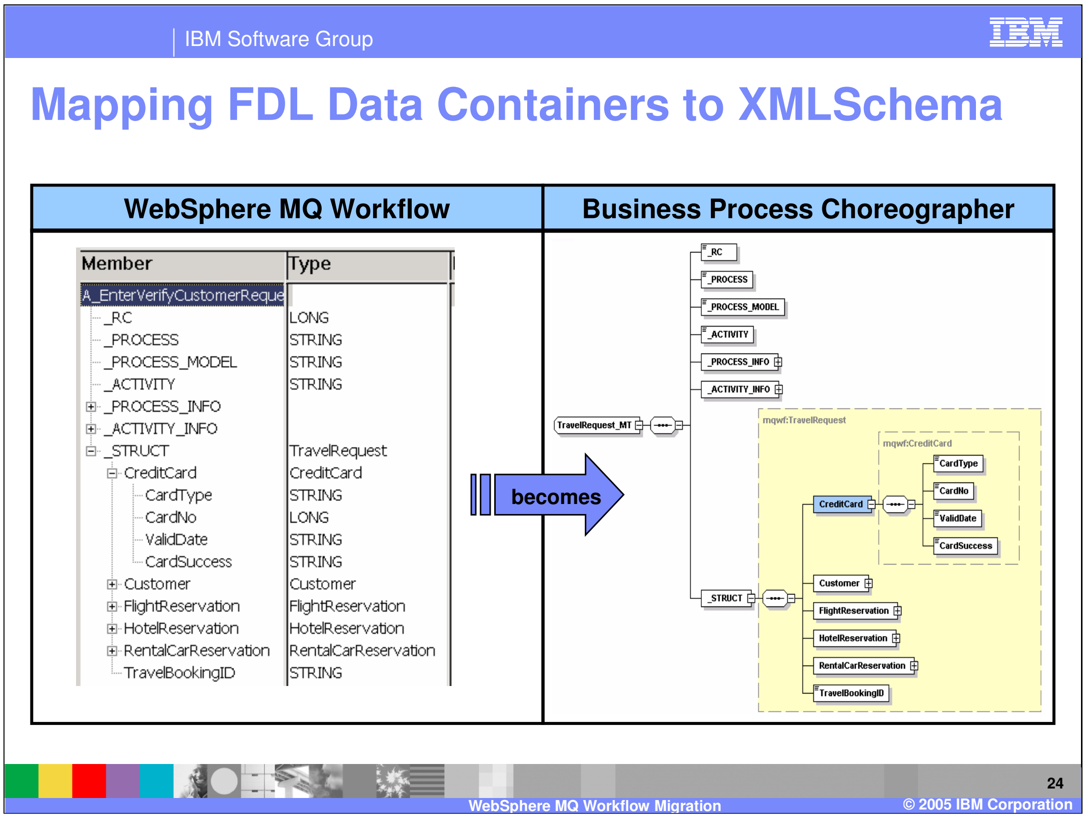


FDL to BPEL Mapping: Control Flow - execution

- Data
- Interface
- Activities
- Data Flow
- **Control Flow (execution)**

WMQWF Process Model Construct	BPEL with Extensions Construct
Transition condition	***Transition condition*** (of *link* element)
Start condition of activity	***Join condition*** (synchronizes *link targets*)
Exit condition of activity	***Condition*** (controls a *While* activity)

WMQWF uses conditions, which are checked to determine when to start, exit or transition from an activity for controlling the flow of the business process. There is a corresponding condition or construct for each of these in the BPEL model.

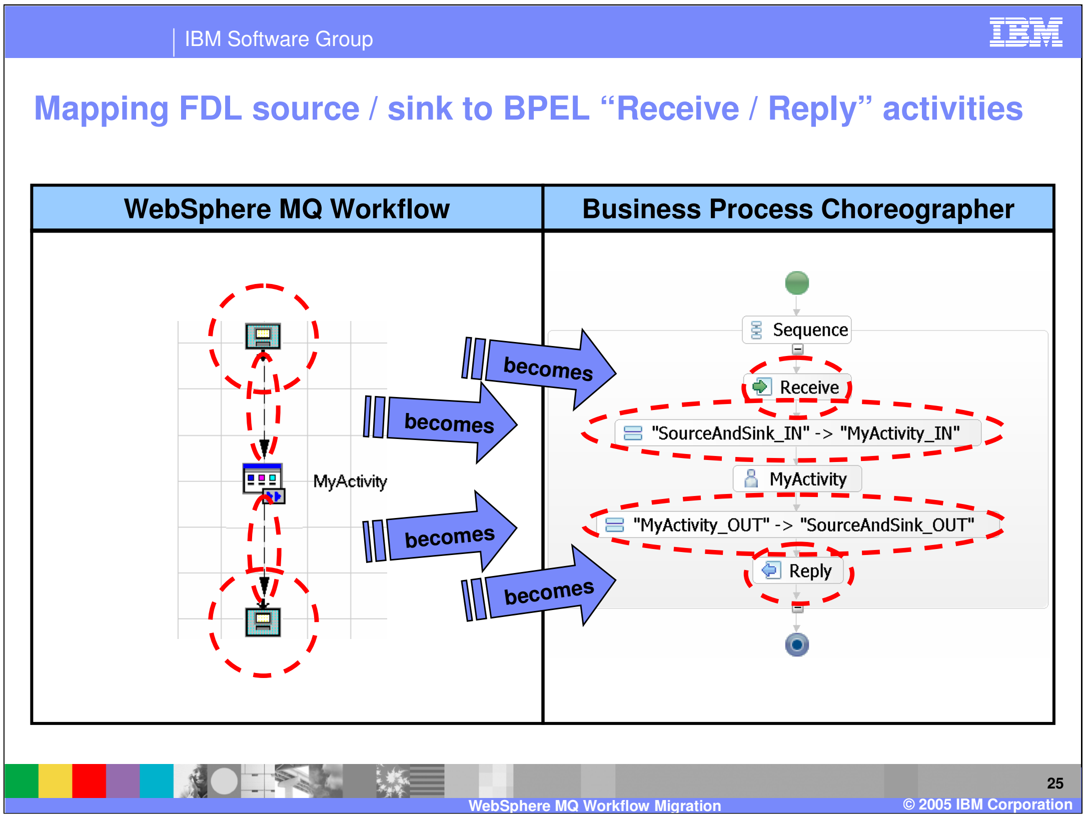


Mapping FDL Data Containers to XMLSchema

WebSphere MQ Workflow	Business Process Choreographer

Member	Type
A_EnterVerifyCustomerReque	
_RC	LONG
_PROCESS	STRING
_PROCESS_MODEL	STRING
_ACTIVITY	STRING
_PROCESS_INFO	
_ACTIVITY_INFO	
_STRUCT	TravelRequest
CreditCard	CreditCard
CardType	STRING
CardNo	LONG
ValidDate	STRING
CardSuccess	STRING
Customer	Customer
FlightReservation	FlightReservation
HotelReservation	HotelReservation
RentalCarReservation	RentalCarReservation
TravelBookingID	STRING

Mapping of the data structures from WMQWF to WPS/BPEL is very straightforward. In WMQWF the data are defined as structures and in WPS/BPEL data are defined using XML Schema and XSD. Both systems provide for complex nested structures, which makes the mapping intuitive.



IBM Software Group

Mapping FDL source / sink to BPEL “Receive / Reply” activities

WebSphere MQ Workflow	Business Process Choreographer

MyActivity
becomes
becomes
becomes
becomes
Sequence
Receive
"SourceAndSink_IN" -> "MyActivity_IN"
MyActivity
"MyActivity_OUT" -> "SourceAndSink_OUT"
Reply

25

WebSphere MQ Workflow Migration

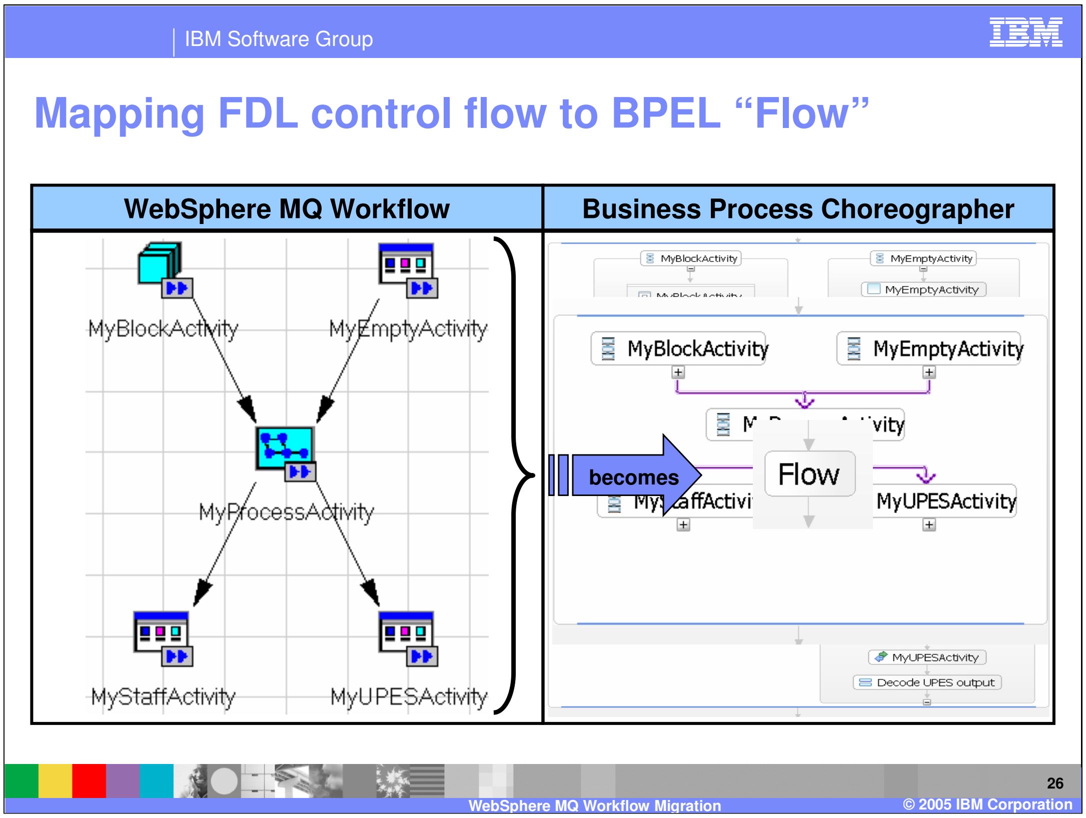
© 2005 IBM Corporation

The WMQWF concept of source and sink map nicely to the BPEL receive/reply activities. Because there is no explicit data flow in the BPEL model, the Assign activities are inserted in the BPEL model to facilitate the movement of data into and out of the activity.

The graphic above illustrates:

1) How the WMQWF Source and Sink activities map to the BPEL Receive and Reply activities.

2) How the WMQWF data flow is mapped to the BPEL Assign activity.



IBM Software Group

Mapping FDL control flow to BPEL “Flow”

WebSphere MQ Workflow	Business Process Choreographer

26

WebSphere MQ Workflow Migration

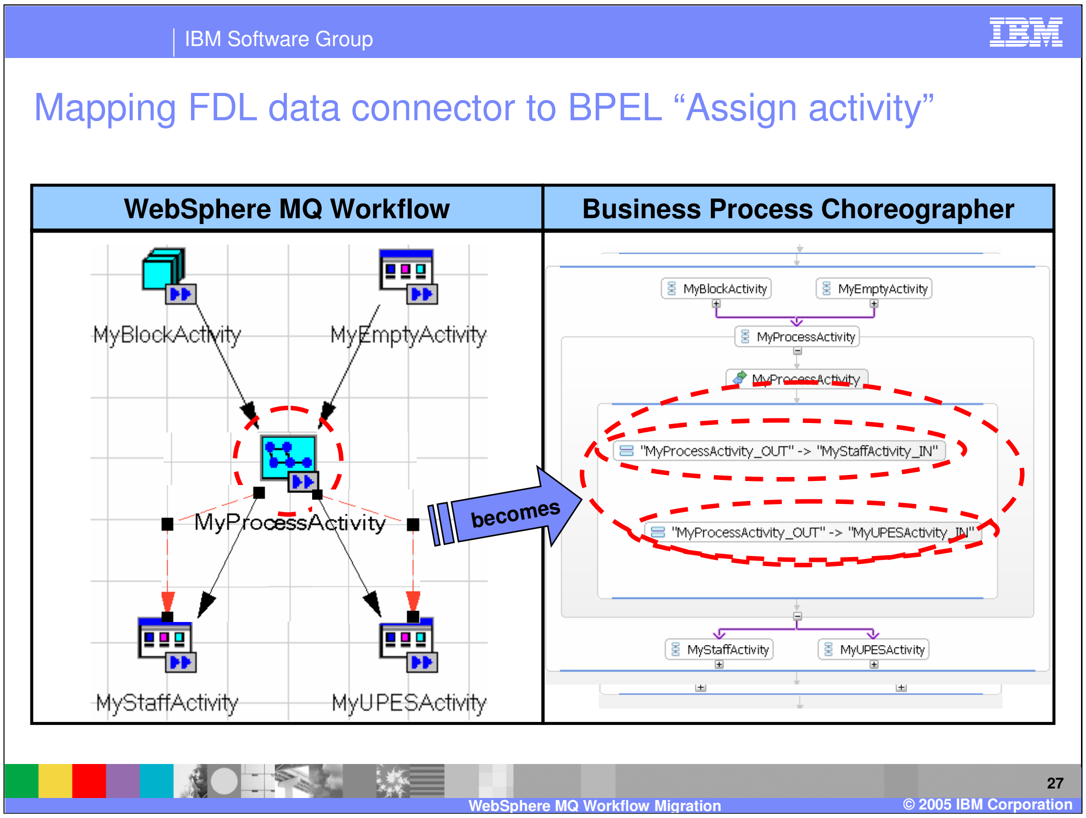
© 2005 IBM Corporation

The generalized WMQWF on the left will be used to illustrate the WMQWF to BPEL mappings. With WMQWF, the concept of multiple concurrent threads is inherent in the model. The multiple paths converge on an activity, which is responsible for synchronizing and distributing the data. There is no special kind of activity.

In BPEL, concurrent execution is modeled with a Flow activity, which is special kind of activity designed to manage and organize the concurrent paths. The contents of the flow can be hidden by collapsing the icon (-) and viewed by expanding it again (+)

The graphic illustrates:

1) The mapping of the entire WMQWF model to a BPEL Flow
2) Expanding the BPEL Flow activity you can see the boundary of the flow is defined by the horizontal blue bars at the top and the bottom. The purple lines represent the links from one activity to the next, the concurrent paths of execution. Notice too, that each activity is also encapsulated by the BPEL Sequence activity, which is another organizing structure.
3) The expansion of the Sequence activities, revealing the details of the activities.



Mapping FDL data connector to BPEL “Assign activity”

In this example, the “MyProcessActivity” has data that must be distributed to 2 different outputs. In WMQWF the data connectors provide the mechanism for achieving this. In BPEL there are no data connectors, so the solution is to insert a BPEL Flow in the BPEL Sequence called “MyProcessActivity” directly after the Invoke activity. The data assignments are then made in the context of the Flow, one for each possible output (data connector).

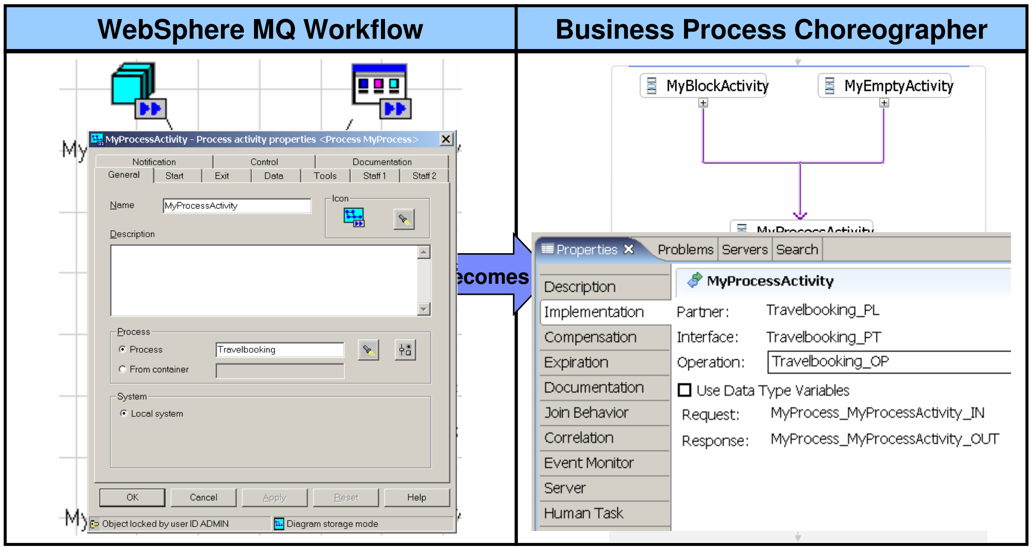
The red circles highlight the area of interest in the graphic shown here, which illustrates:

1) The data connectors on the WMQWF side and the insertion of the sequence on the BPEL side.

2) The mapping of the data connectors to the Assign activities in the BPEL Sequence.

Mapping FDL process activity to BPEL “*invoke activity*”

WebSphere MQ Workflow	Business Process Choreographer



MyProcessActivity - Process activity properties <Process MyProcess>

Name: MyProcessActivity

Process
● Process: Travelbooking
○ From container

System
● Local system

Object locked by user ID ADMIN | Diagram storage mode

MyProcessActivity

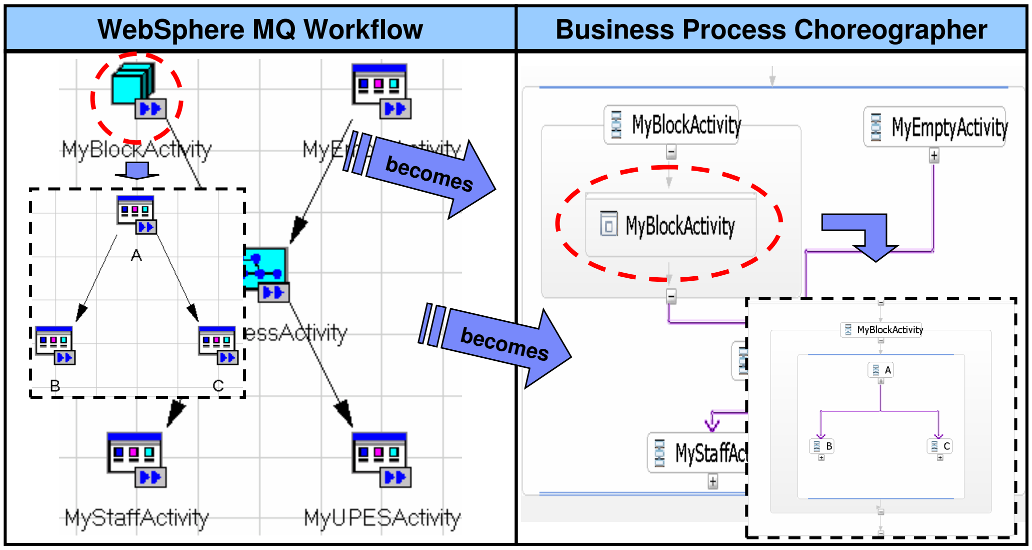
Partner: Travelbooking_PL
Interface: Travelbooking_PT
Operation: Travelbooking_OP
□ Use Data Type Variables
Request: MyProcess_MyProcessActivity_IN
Response: MyProcess_MyProcessActivity_OUT

Mapping the WMQWF activity to the BPEL Invoke activity is straightforward.

Remember that with WMQWF, the kind, or behavior of an activity is defined by the properties. The property panels for both the WMQWF and the BPEL are shown to illustrate how the details are mapped from one to the other. Notice that the name of the WMQWF activity, which is a process, is mapped to an operation on a Web Service in the BPEL implementation.

IBM Software Group

Mapping FDL block to BPEL “*Scope*”



29

WebSphere MQ Workflow Migration

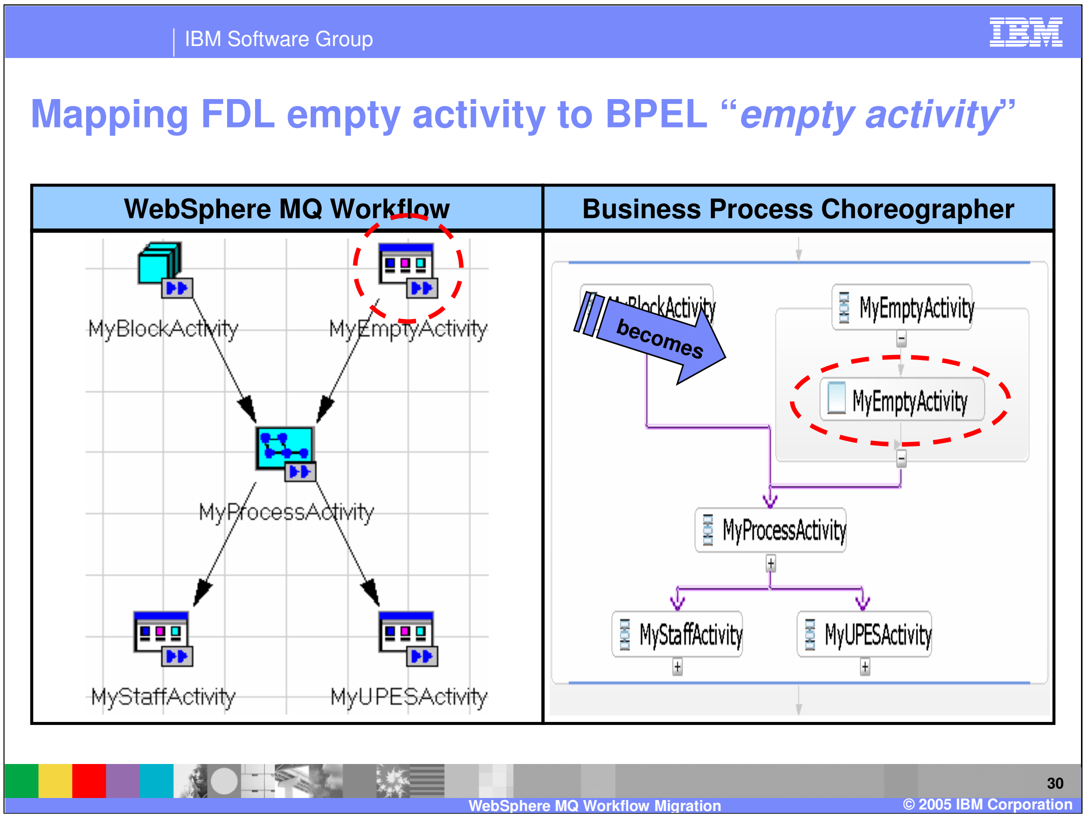
© 2005 IBM Corporation

The concept of Block and Scope are identical. They both provide a mechanism to create a local boundary, which is used to create a local context for isolating local variables and associating units of work and error handling constructs. They are also used to organize and hide the complexity.

In the graphic shown:

1) The red circles highlight the area of interest.

2) The WMQWF Block and BPEL scope are expanded to reveal the details.



IBM Software Group

Mapping FDL empty activity to BPEL "*empty activity*"

WebSphere MQ Workflow	Business Process Choreographer

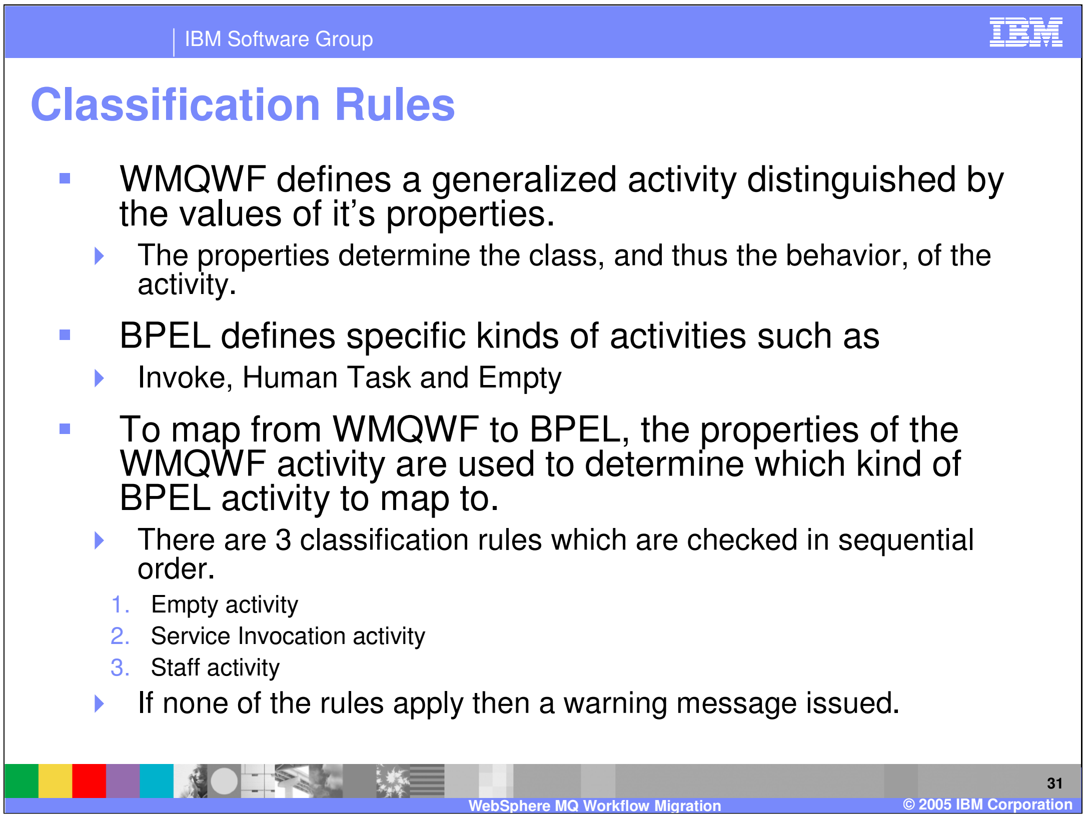
30

WebSphere MQ Workflow Migration

There is a direct mapping of the Empty Activity from WMQWF to BPEL. To determine if the WMQWF Activity is truly an Empty activity there are properties that must be interrogated first. The application of the classification rules will be discussed in the next few slides.

In the graphic shown:

1) The red circles highlight the areas of interest.

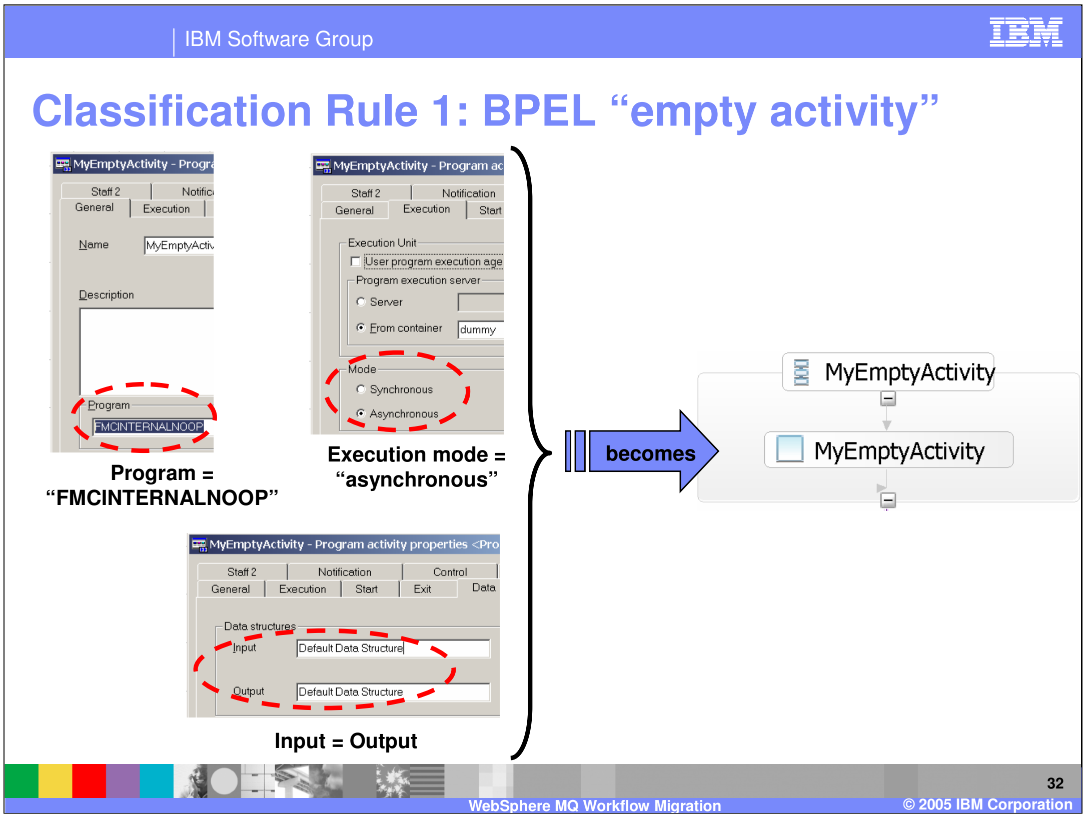


Classification Rules

- WMQWF defines a generalized activity distinguished by the values of it’s properties.
  - The properties determine the class, and thus the behavior, of the activity.
- BPEL defines specific kinds of activities such as
  - Invoke, Human Task and Empty
- To map from WMQWF to BPEL, the properties of the WMQWF activity are used to determine which kind of BPEL activity to map to.
  - There are 3 classification rules which are checked in sequential order.
    1. Empty activity
    2. Service Invocation activity
    3. Staff activity
  - If none of the rules apply then a warning message issued.

The WMQWF activities are defined using properties whereas with BPEL there distinct kinds of activities, such as Human Task (Staff), Empty or Service Invocation. To determine which kind of BPEL activity to map the WMQWF activity to, the properties must be inspected and the classification rules applied.

Classification Rule 1: BPEL “empty activity”



Name MyEmptyActiv

Description

Program FMCINTERNALNOOP

Program = “FMCINTERNALNOOP”

Execution Unit

☐ User program execution age

Program execution server

○ Server

● From container dummy

Mode

○ Synchronous

● Asynchronous

Execution mode = “asynchronous”

Data structures

Input Default Data Structure

Output Default Data Structure

Input = Output

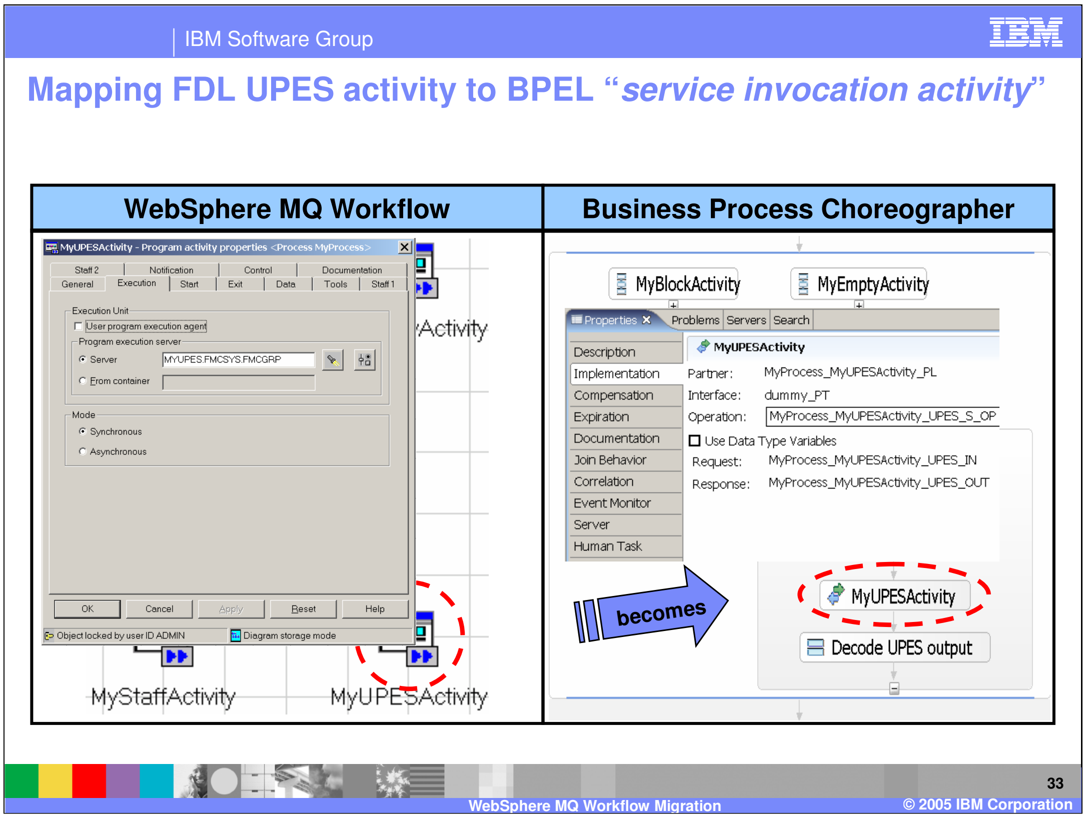
Classification Rule 1:

This is the first rule that is checked.

The migration utility will inspect the properties associated with the “Program Activity” and if it is defined to be a no-operation (NOOP) type of activity, which is asynchronous, and there is no change in the data, the activity is mapped to the Empty BPEL activity.

All three conditions must be met.

In other words, If “FMINTERNALNOOP” and “asynchronous” and “Input=Output” then Activity is of type “Empty”



Mapping FDL UPES activity to BPEL “*service invocation activity*”

WebSphere MQ Workflow	Business Process Choreographer

MyUPESActivity - Program activity properties <Process MyProcess>

Staff 2 | Notification | Control | Documentation

General | Execution | Start | Exit | Data | Tools | Staff 1

Execution Unit

□ User program execution agent

Program execution server

● Server MYUPES.FMCSYS.FMCGRP

○ From container

Mode

● Synchronous

○ Asynchronous

MyUPESActivity

Partner: MyProcess_MyUPESActivity_PL

Interface: dummy_PT

Operation: MyProcess_MyUPESActivity_UPES_S_OP

□ Use Data Type Variables

Request: MyProcess_MyUPESActivity_UPES_IN

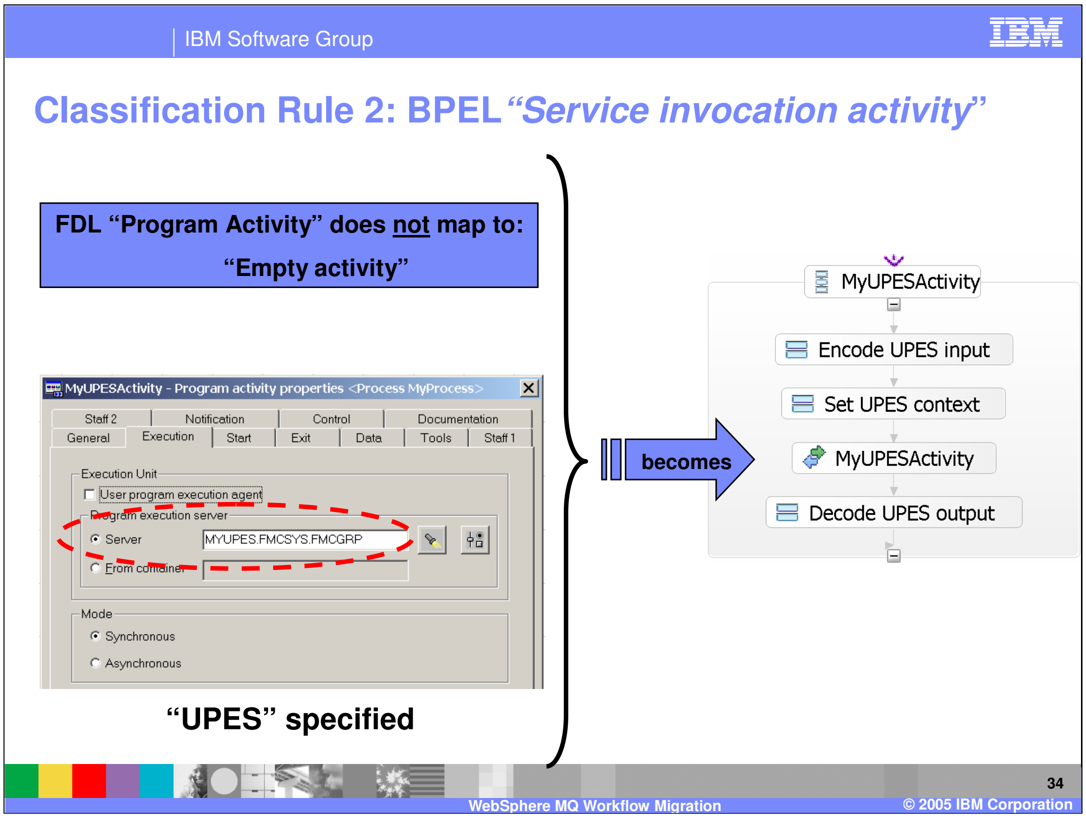
Response: MyProcess_MyUPESActivity_UPES_OUT

The WMQWF UPES activity is mapped to BPEL Invoke activity. The program executed by the UPES becomes the implementation for a Web Service invocation from BPEL.

The Migration Wizard will create the corresponding Invoke activity for the UPES but there is no support for creating the Web Service implementation. There is no UPES compatibility layer provided at this time.

In the graphic shown:

1) The red circles highlight the areas of interest
2) The details of the activity definitions are displayed for comparison.



IBM Software Group

Classification Rule 2: BPEL *"Service invocation activity"*

FDL "Program Activity" does not map to: "Empty activity"

MyUPESActivity - Program activity properties <Process MyProcess>

Staff 2 | Notification | Control | Documentation
General | Execution | Start | Exit | Data | Tools | Staff 1

Execution Unit

☐ User program execution agent

Program execution server

● Server MYUPES.FMCSYS.FMCGRP

○ From container

Mode

● Synchronous

○ Asynchronous

"UPES" specified

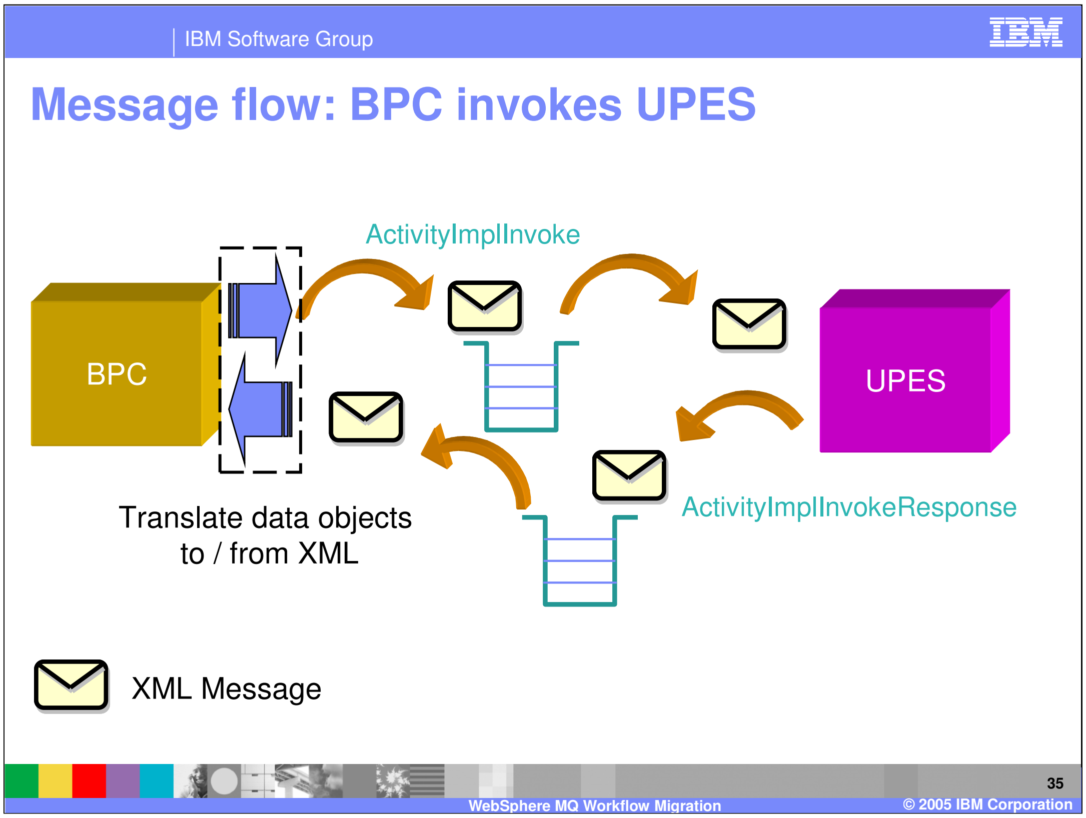
WebSphere MQ Workflow Migration

34

© 2005 IBM Corporation

Classification Rule 2:

If the execution mode of the WMQWF "Program Activity" is configured to be a server, then the activity is mapped to a BPEL Invoke Activity.



Message flow: BPC invokes UPES

ActivityImplInvoke

BPC

UPES

Translate data objects
to / from XML

ActivityImplInvokeResponse

XML Message

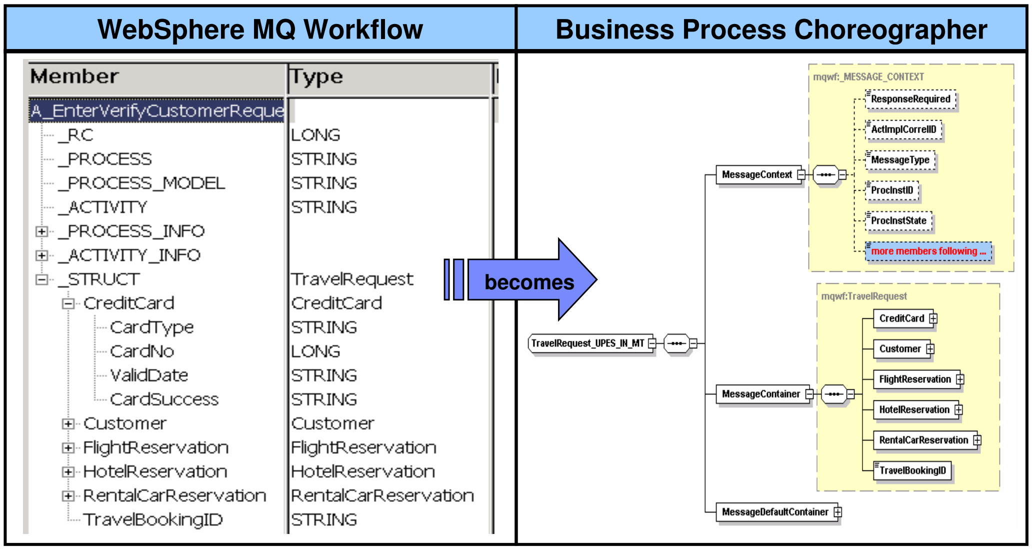
This illustrates the message flow involved with invoking a UPES from Business Process Choreographer.

The implementation of MyUPESActivity (BPEL) is responsible for marshalling the message and calling the UPES.

In the WPS V6.0.1 Information Center, this is referred to as the UPES compatibility layer. The UPES compatibility layer is not provided with WMQWF and must be developed by integration developers.

Adding UPES message types to XMLSchema definitions

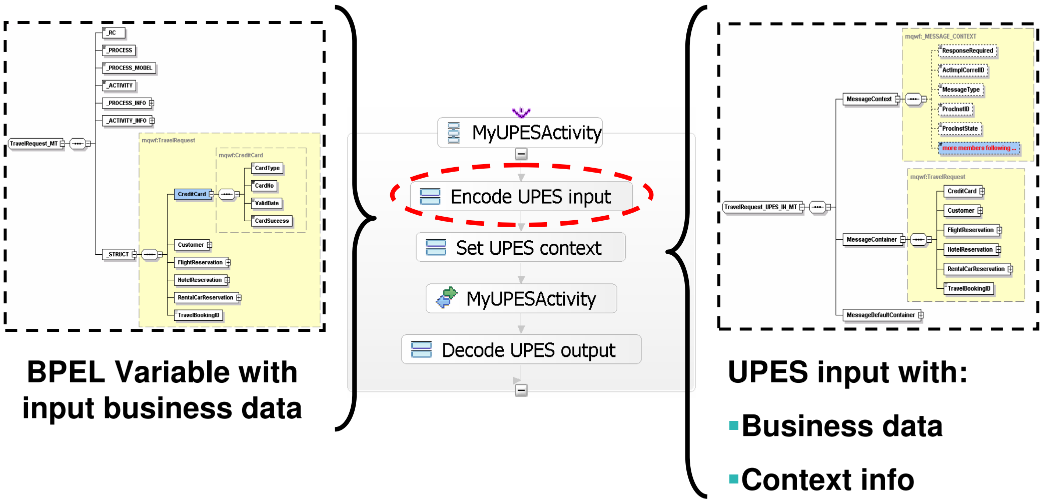
WebSphere MQ Workflow	Business Process Choreographer



Member	Type
A_EnterVerifyCustomerReque	
_RC	LONG
_PROCESS	STRING
_PROCESS_MODEL	STRING
_ACTIVITY	STRING
_PROCESS_INFO	
_ACTIVITY_INFO	
_STRUCT	TravelRequest
CreditCard	CreditCard
CardType	STRING
CardNo	LONG
ValidDate	STRING
CardSuccess	STRING
Customer	Customer
FlightReservation	FlightReservation
HotelReservation	HotelReservation
RentalCarReservation	RentalCarReservation
TravelBookingID	STRING

The UPES message types will be converted to XML Schema

Encoding UPES input

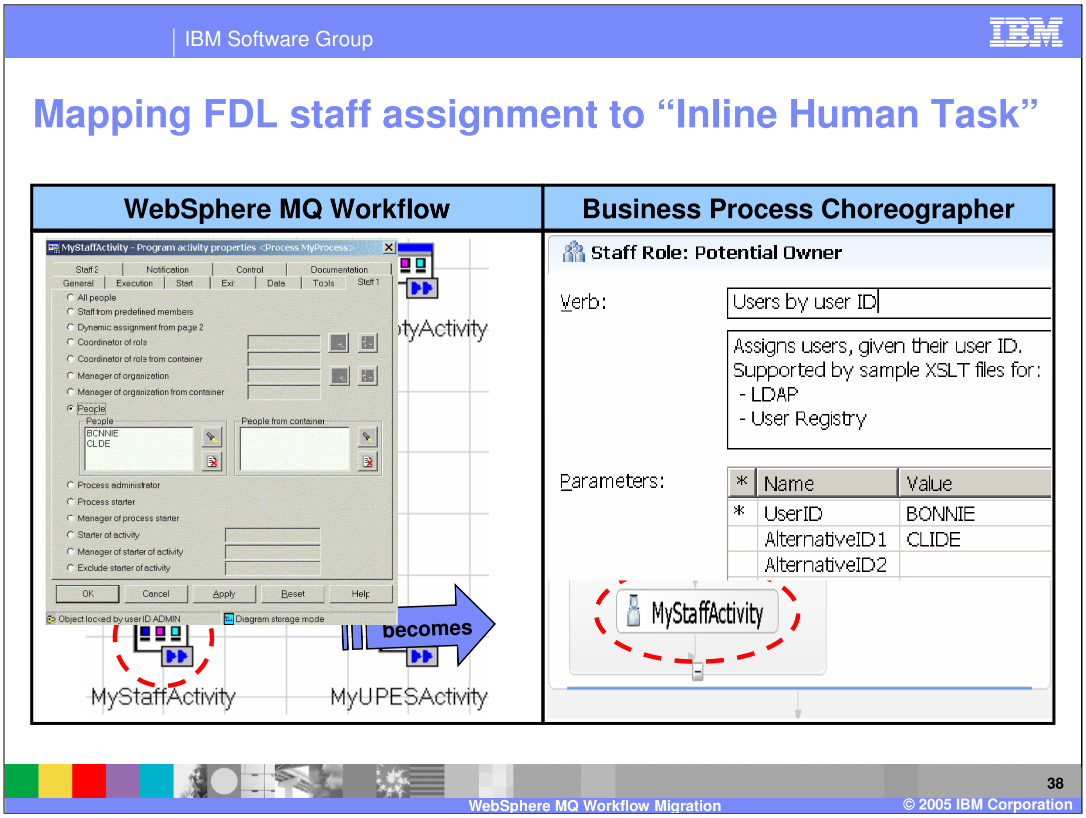


BPEL Variable with input business data

UPES input with:

- **Business data**
- **Context info**

BPEL Assignment activities will be inserted to copy the data to/from the BPEL variables and the UPES Business data and context.



IBM Software Group

Mapping FDL staff assignment to “Inline Human Task”

WebSphere MQ Workflow	Business Process Choreographer

MyStaffActivity - Program activity properties <Process MyProcess>

○ All people
○ Staff from predefined members
○ Dynamic assignment from page 2
○ Coordinator of role
○ Coordinator of role from container
○ Manager of organization
○ Manager of organization from container
● People

People: BCNNIE, CLDE

People from container:

○ Process administrator
○ Process starter
○ Manager of process starter
○ Starter of activity
○ Manager of starter of activity
○ Exclude starter of activity

Object locked by user ID ADMIN | Diagram storage mode

Staff Role: Potential Owner

Verb: Users by user ID

Assigns users, given their user ID.
Supported by sample XSLT files for:
- LDAP
- User Registry

Parameters:

*	Name	Value
*	UserID	BONNIE
	AlternativeID1	CLIDE
	AlternativeID2	

38

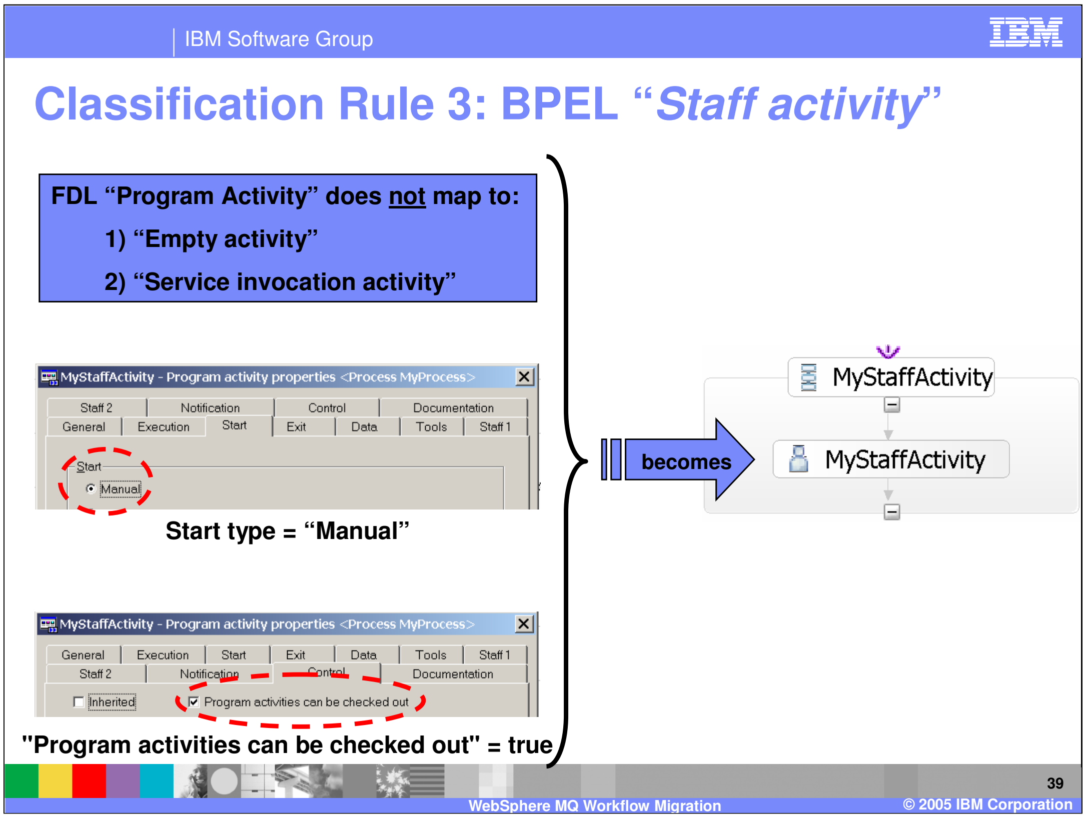
WebSphere MQ Workflow Migration

The WPS BPEL extension for the Human Task is a direct analog for the WMQWF Staff activity.

The inline Human Task activity provides strong integration capabilities between the business process and the human task but is not BPEL 2.0 compliant.

In the graphic shown:

1) The red circles highlight the areas of interest.
2) The WMQWF Staff becomes an inline Human Task.
3) The details of the two activities are displayed for comparison.
4) The Human Task is further expanded to show the association of the groups and people permitted to interact with the task.



Classification Rule 3: BPEL “*Staff activity*”

FDL “Program Activity” does not map to:

1) “Empty activity”

2) “Service invocation activity”

Start

● Manual

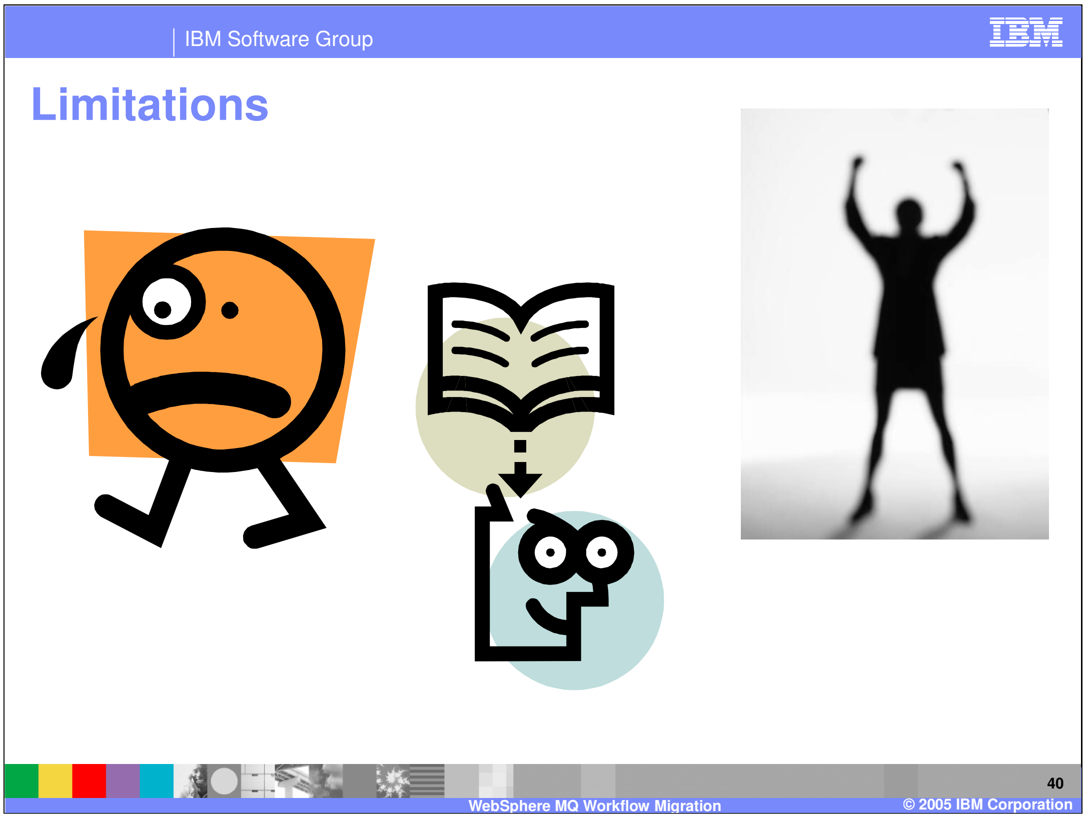
Start type = “Manual”

□ Inherited

☑ Program activities can be checked out

"Program activities can be checked out" = true

The third classification rule checks the properties of the WMQWF “Program Activity” to determine if it has been configured to be checked out and requires manual intervention. If this is the case, then the appropriate corresponding BPEL activity will be an inline Human Task activity.



IBM Software Group

Limitations

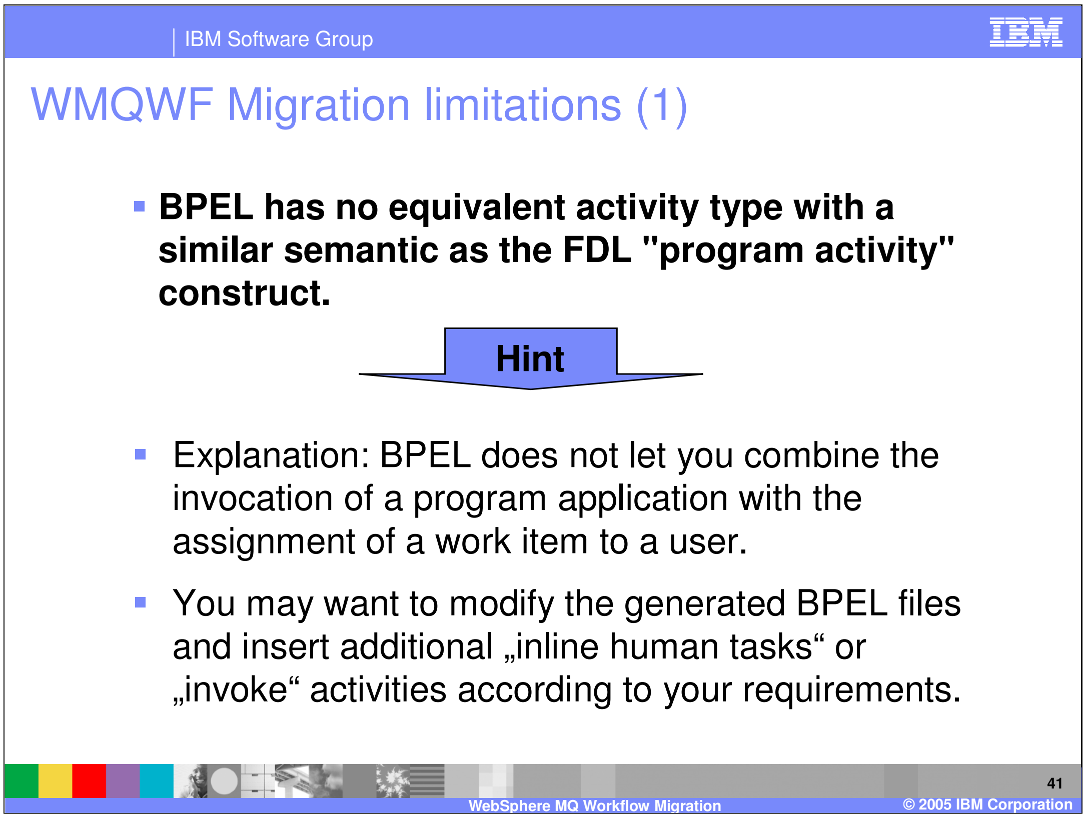
40

WebSphere MQ Workflow Migration

© 2005 IBM Corporation

When mapping from one programming model to another, different programming model there will always be some features that are new, different or missing and therefore require manual intervention.

The mismatch between the WMQWF model and the WPS V6 SCA programming model is captured here in the form of limitations. Knowing the limitations and discontinuities ahead of time is extremely valuable in determining the amount of effort required to complete the migration.



IBM Software Group

WMQWF Migration limitations (1)

- **BPEL has no equivalent activity type with a similar semantic as the FDL "program activity" construct.**

Hint

- Explanation: BPEL does not let you combine the invocation of a program application with the assignment of a work item to a user.
- You may want to modify the generated BPEL files and insert additional „inline human tasks“ or „invoke“ activities according to your requirements.

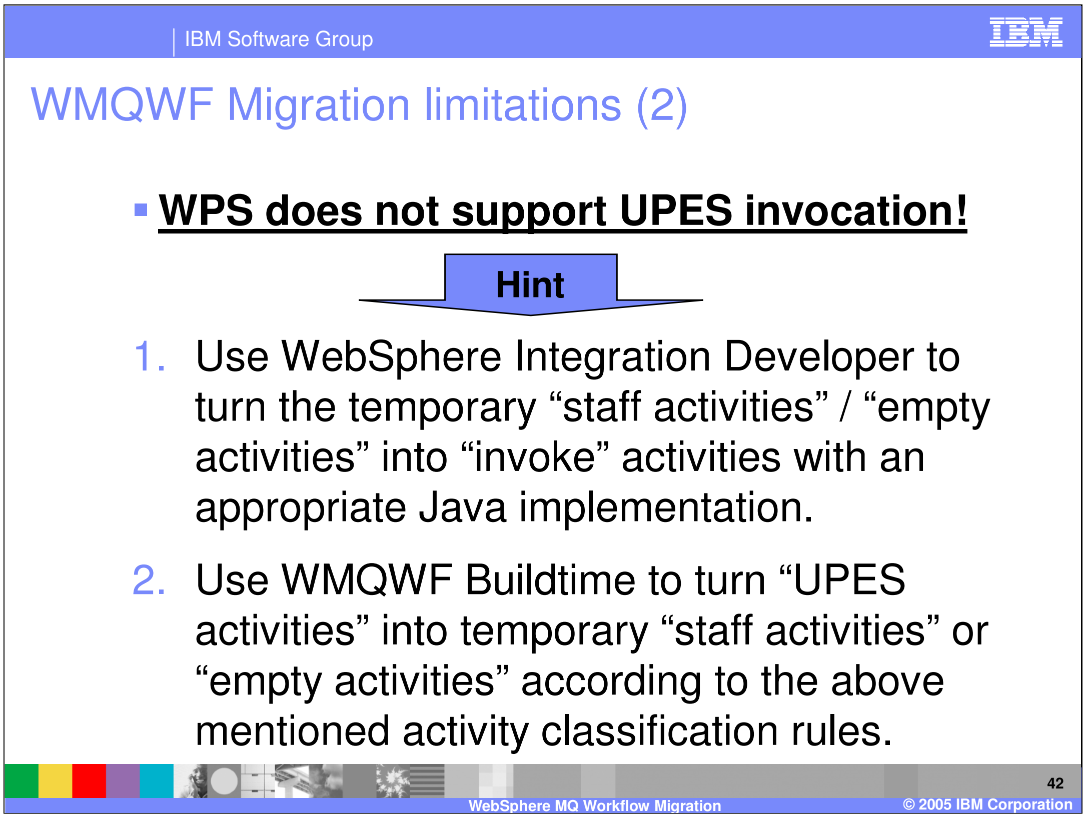
41

WebSphere MQ Workflow Migration

© 2005 IBM Corporation

•The FDL concept of a „program activity“ means that the „program execution agent“ (PEA) executes the application on the workstation of the user who claimed the respective workitem. With BPEL you cannot specify the **location** of the program execution as the location of the user.

•You could insert additional **synchronization** points with user interaction. For example you could insert an „inline human task“ that precedes an „invoke“ activity so that the user can decide when the respective service is allowed to execute.



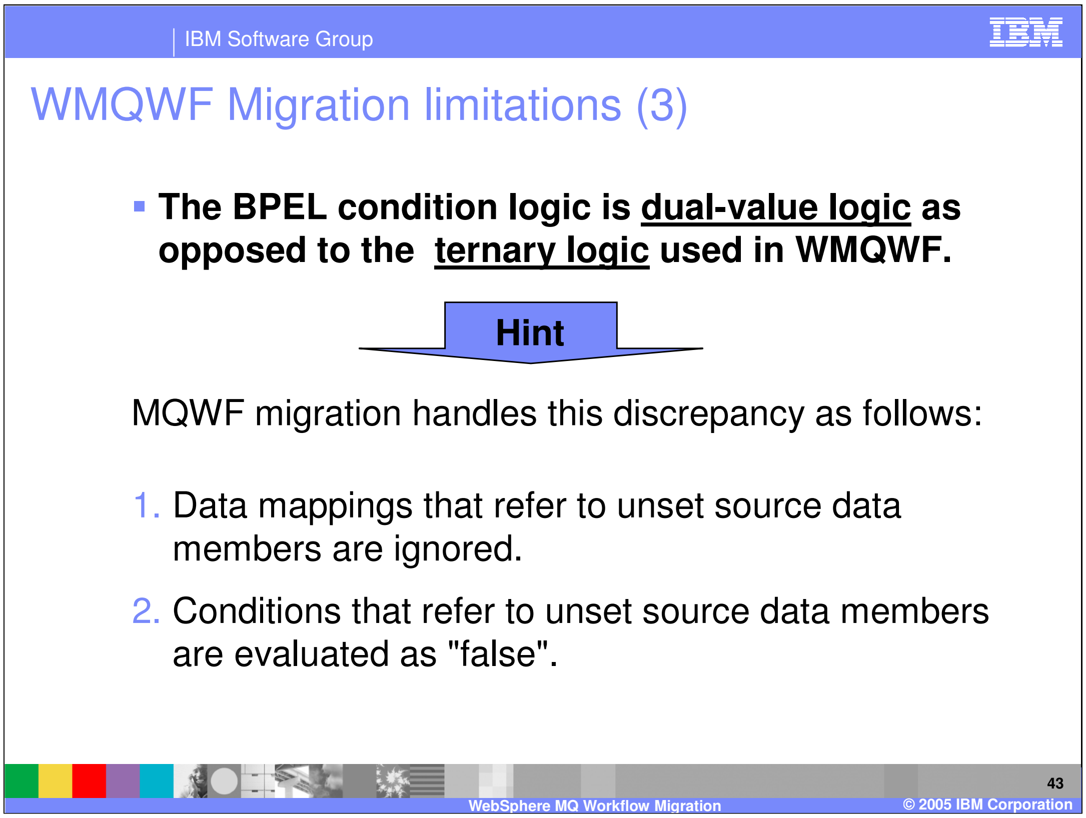
WMQWF Migration limitations (2)

- **WPS does not support UPES invocation!**

Hint

1. Use WebSphere Integration Developer to turn the temporary “staff activities” / “empty activities” into “invoke” activities with an appropriate Java implementation.
2. Use WMQWF Buildtime to turn “UPES activities” into temporary “staff activities” or “empty activities” according to the above mentioned activity classification rules.

WPS does not yet support UPES invocation.



IBM Software Group

WMQWF Migration limitations (3)

- **The BPEL condition logic is <u>dual-value logic</u> as opposed to the <u>ternary logic</u> used in WMQWF.**

Hint

MQWF migration handles this discrepancy as follows:

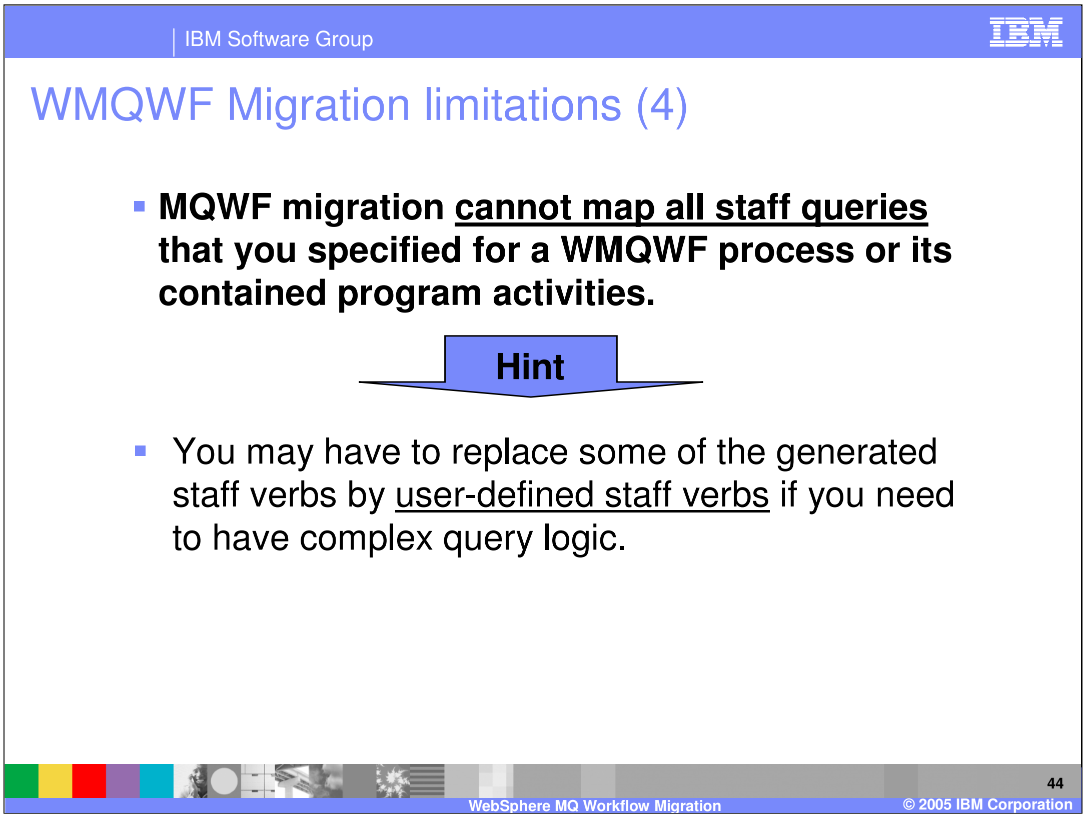
1. Data mappings that refer to unset source data members are ignored.
2. Conditions that refer to unset source data members are evaluated as "false".

43

WebSphere MQ Workflow Migration

© 2005 IBM Corporation

The BPEL condition logic is dual-value logic as opposed to the ternary logic used in WMQWF.



IBM Software Group

WMQWF Migration limitations (4)

- **MQWF migration cannot map all staff queries that you specified for a WMQWF process or its contained program activities.**

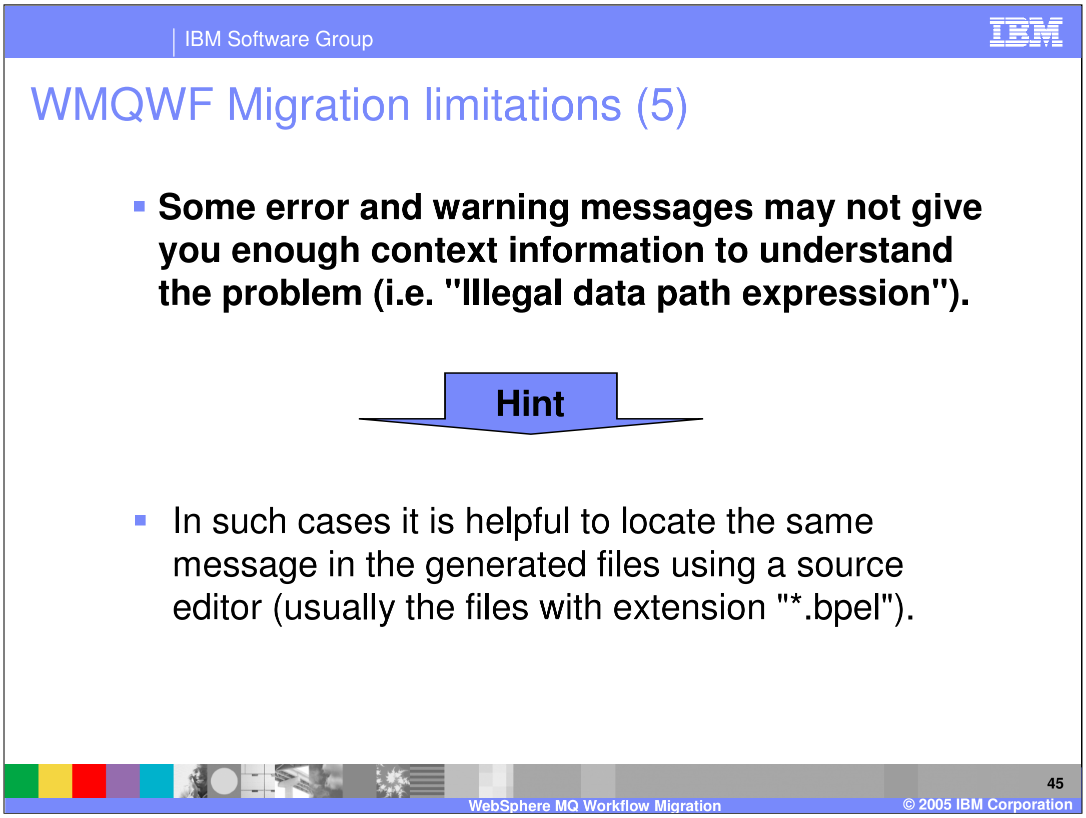
Hint

- You may have to replace some of the generated staff verbs by user-defined staff verbs if you need to have complex query logic.

44

WebSphere MQ Workflow Migration

MQWF migration cannot map all staff queries that you specified for a WMQWF process or its contained program activities.



IBM Software Group

WMQWF Migration limitations (5)

- **Some error and warning messages may not give you enough context information to understand the problem (i.e. "Illegal data path expression").**

Hint

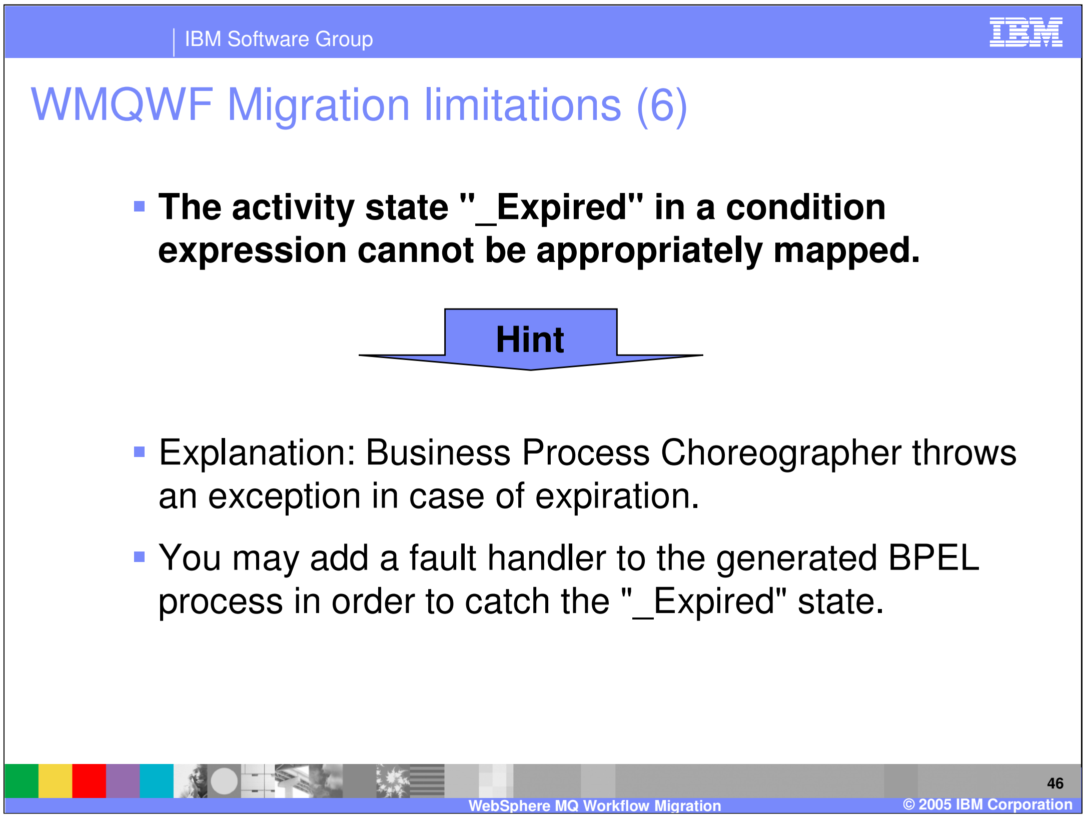
- In such cases it is helpful to locate the same message in the generated files using a source editor (usually the files with extension "*.bpel").

45

WebSphere MQ Workflow Migration

© 2005 IBM Corporation

While FDL2BPEL generates the target constructs, the context of a problem situation is sometimes unknown, and the appropriate information is not available to give a clear explanation. But the error message will always be inserted as a comment annotation at the respective location in the generated BPEL or WSDL file where the problem occurred. So, usually finding the same message in the file will help to understand the problem. For instance, the above message "Illegal data path expression" might be found in a condition expression that refers to an undefined data member in the input variable of an activity.



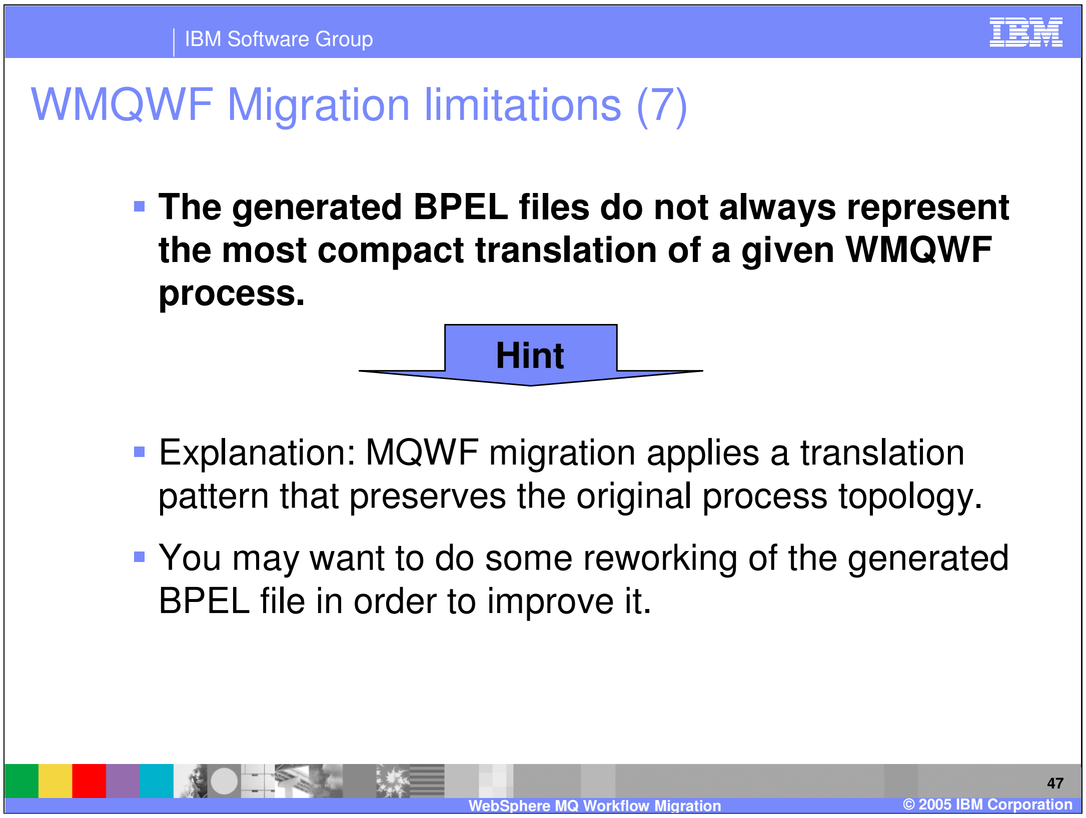
WMQWF Migration limitations (6)

- **The activity state "_Expired" in a condition expression cannot be appropriately mapped.**

Hint

- Explanation: Business Process Choreographer throws an exception in case of expiration.
- You may add a fault handler to the generated BPEL process in order to catch the "_Expired" state.

The activity state "_Expired" in a condition expression cannot be appropriately mapped.



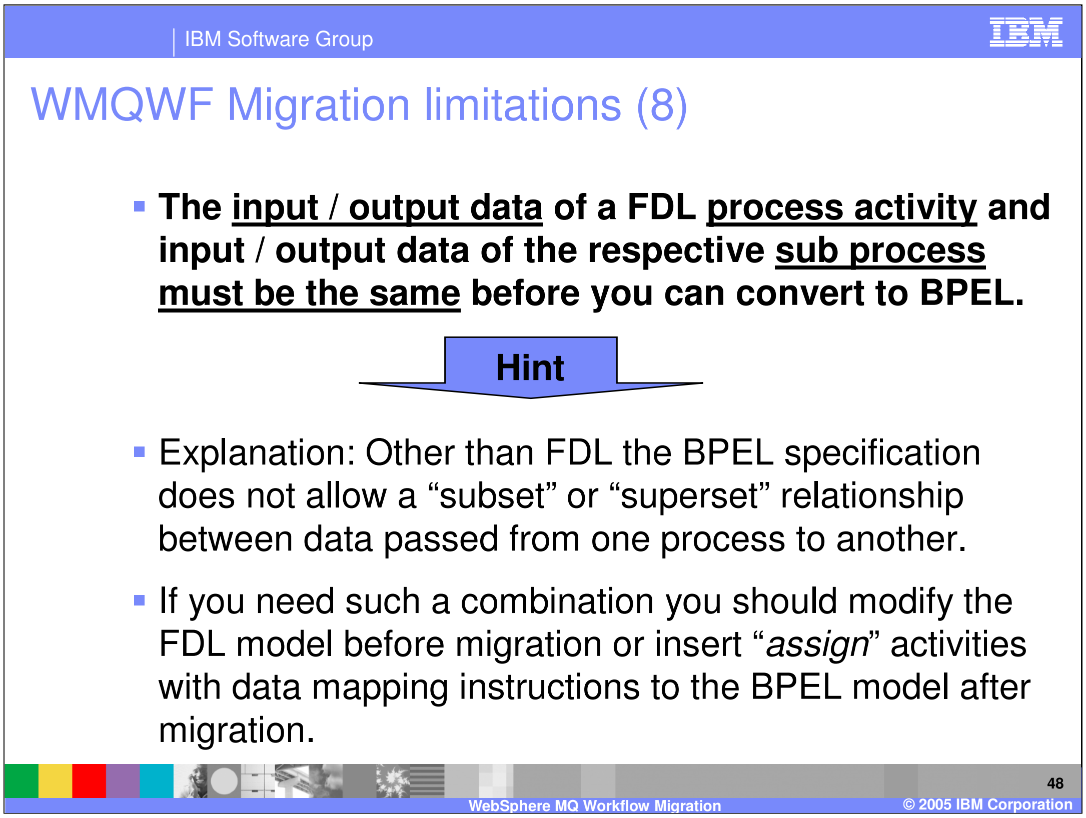
WMQWF Migration limitations (7)

- **The generated BPEL files do not always represent the most compact translation of a given WMQWF process.**

Hint

- Explanation: MQWF migration applies a translation pattern that preserves the original process topology.
- You may want to do some reworking of the generated BPEL file in order to improve it.

For instance, a data connector that passes data unchanged from one activity to another might sometimes be better mapped to a shared BPEL variable where the FDL2BPEL conversion tool generates two BPEL variables and an "assign" activity that does the data mapping.



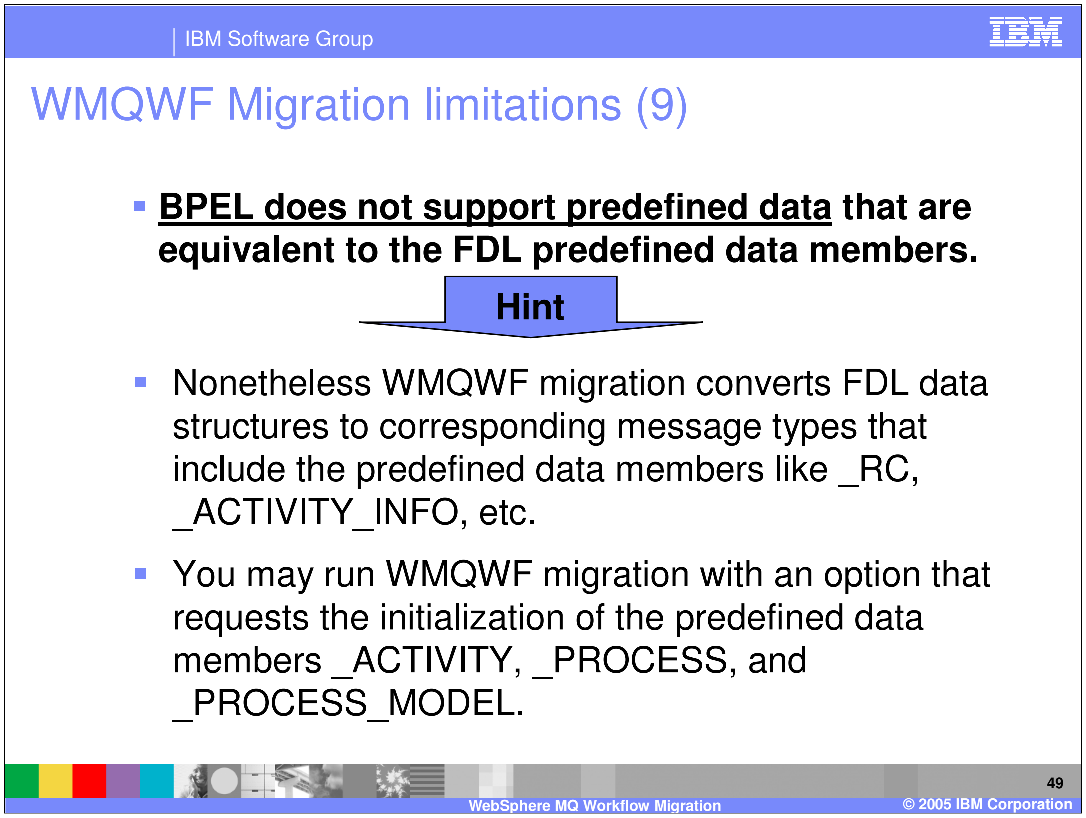
WMQWF Migration limitations (8)

- **The <u>input / output data</u> of a FDL <u>process activity</u> and input / output data of the respective <u>sub process</u> <u>must be the same</u> before you can convert to BPEL.**

Hint

- Explanation: Other than FDL the BPEL specification does not allow a “subset” or “superset” relationship between data passed from one process to another.
- If you need such a combination you should modify the FDL model before migration or insert “*assign*” activities with data mapping instructions to the BPEL model after migration.

The input / output data of a FDL process activity and input / output data of the respective sub process must be the same before you can convert to BPEL.



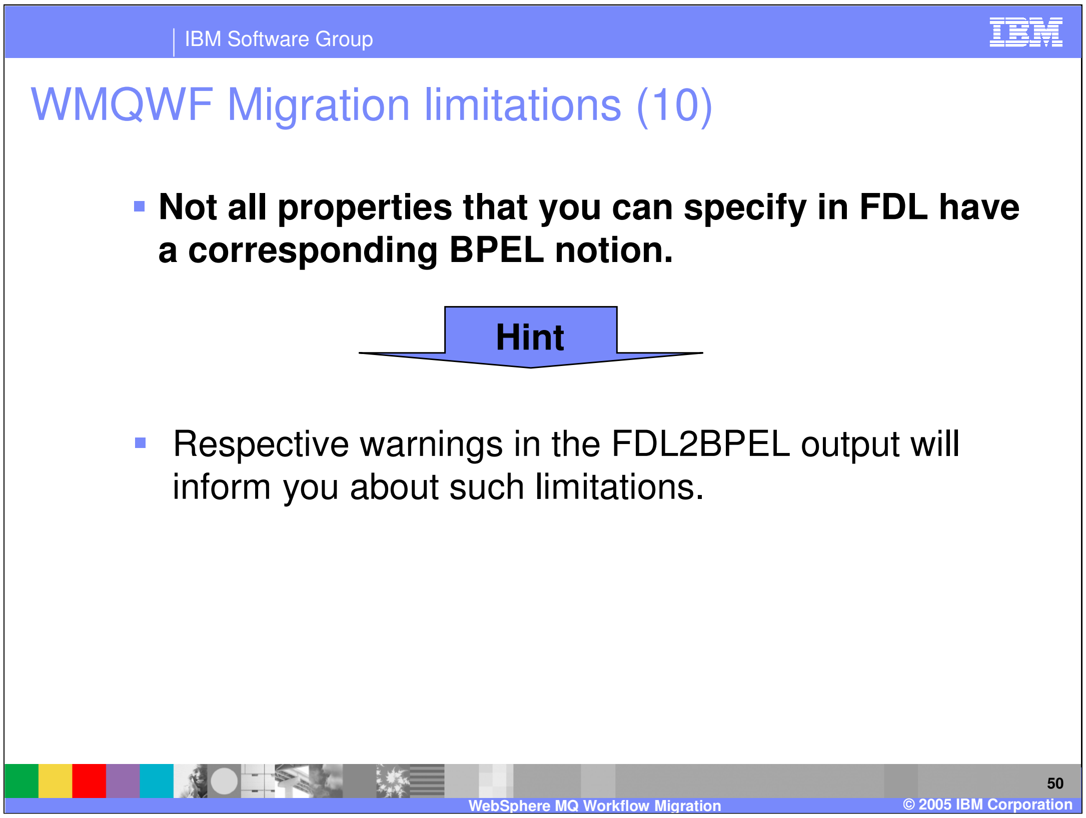
WMQWF Migration limitations (9)

- **BPEL does not support predefined data that are equivalent to the FDL predefined data members.**

Hint

- Nonetheless WMQWF migration converts FDL data structures to corresponding message types that include the predefined data members like _RC, _ACTIVITY_INFO, etc.
- You may run WMQWF migration with an option that requests the initialization of the predefined data members _ACTIVITY, _PROCESS, and _PROCESS_MODEL.

Business Process Choreographer does not offer support for setting predefined data members.



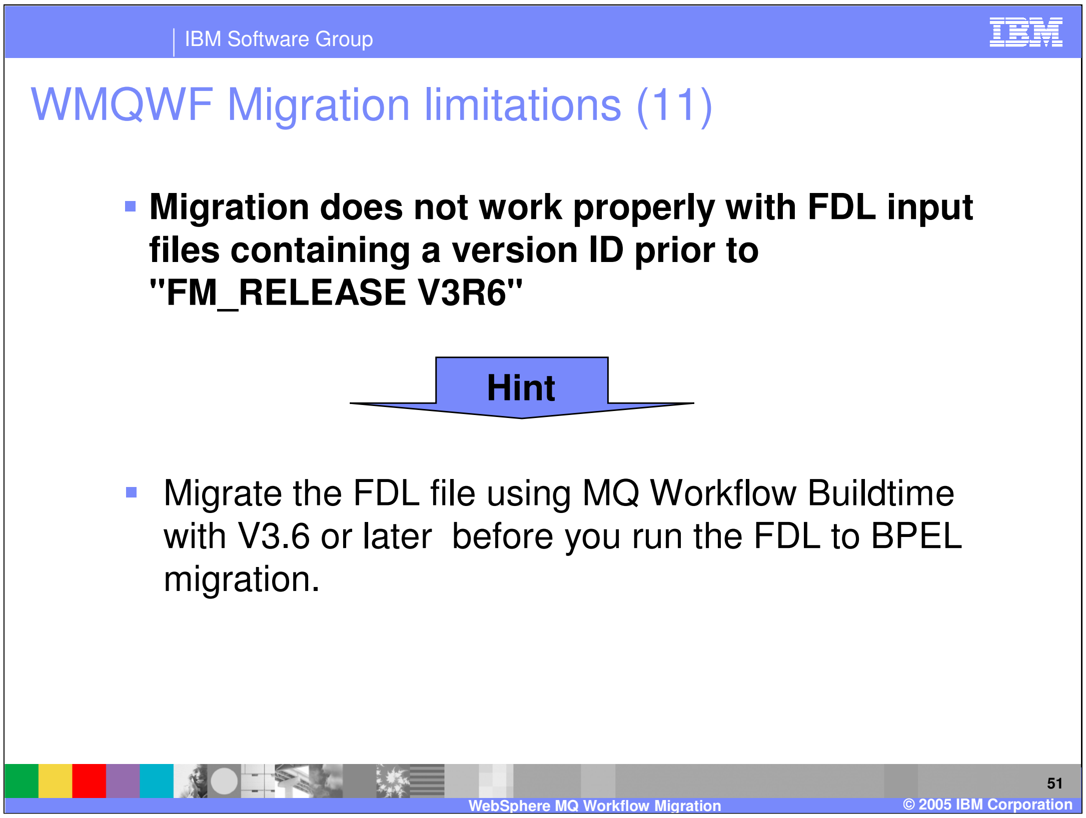
WMQWF Migration limitations (10)

- **Not all properties that you can specify in FDL have a corresponding BPEL notion.**

Hint

- Respective warnings in the FDL2BPEL output will inform you about such limitations.

The concept of a „global container“ is not yet supported in BPEL. Nevertheless, the WMQWF migration generates a BPEL variable for it.



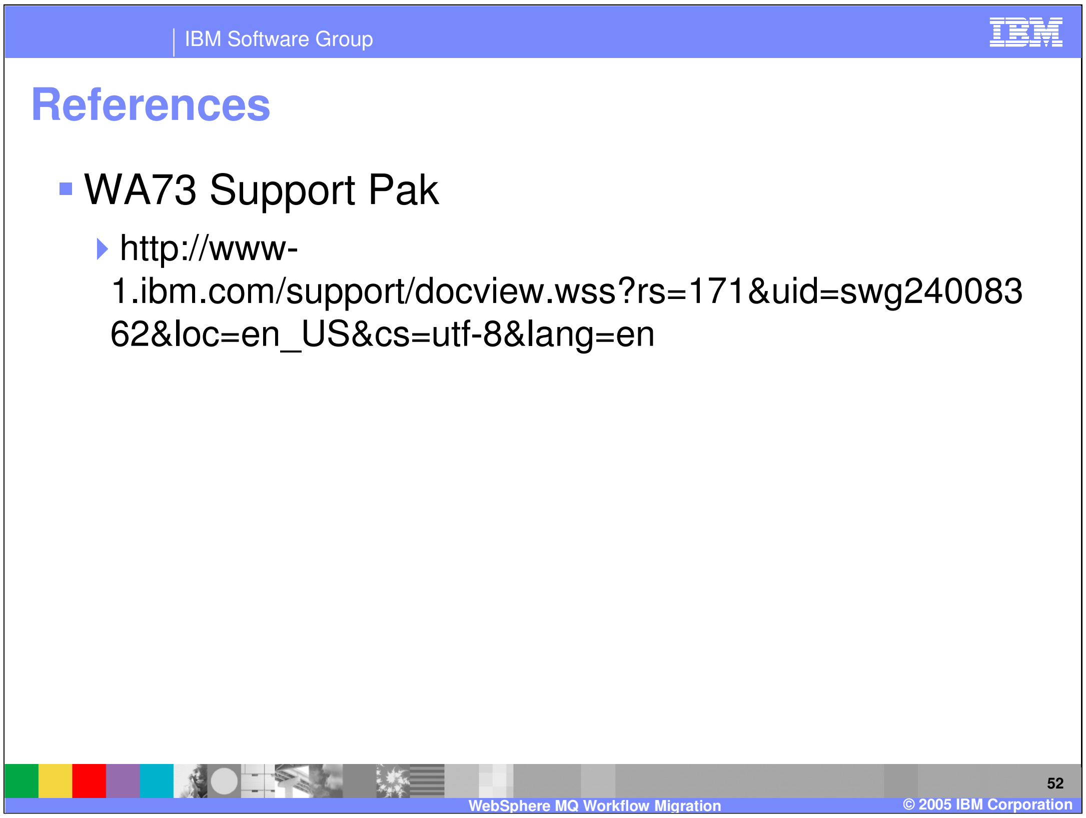
WMQWF Migration limitations (11)

- **Migration does not work properly with FDL input files containing a version ID prior to "FM_RELEASE V3R6"**

Hint

- Migrate the FDL file using MQ Workflow Buildtime with V3.6 or later before you run the FDL to BPEL migration.

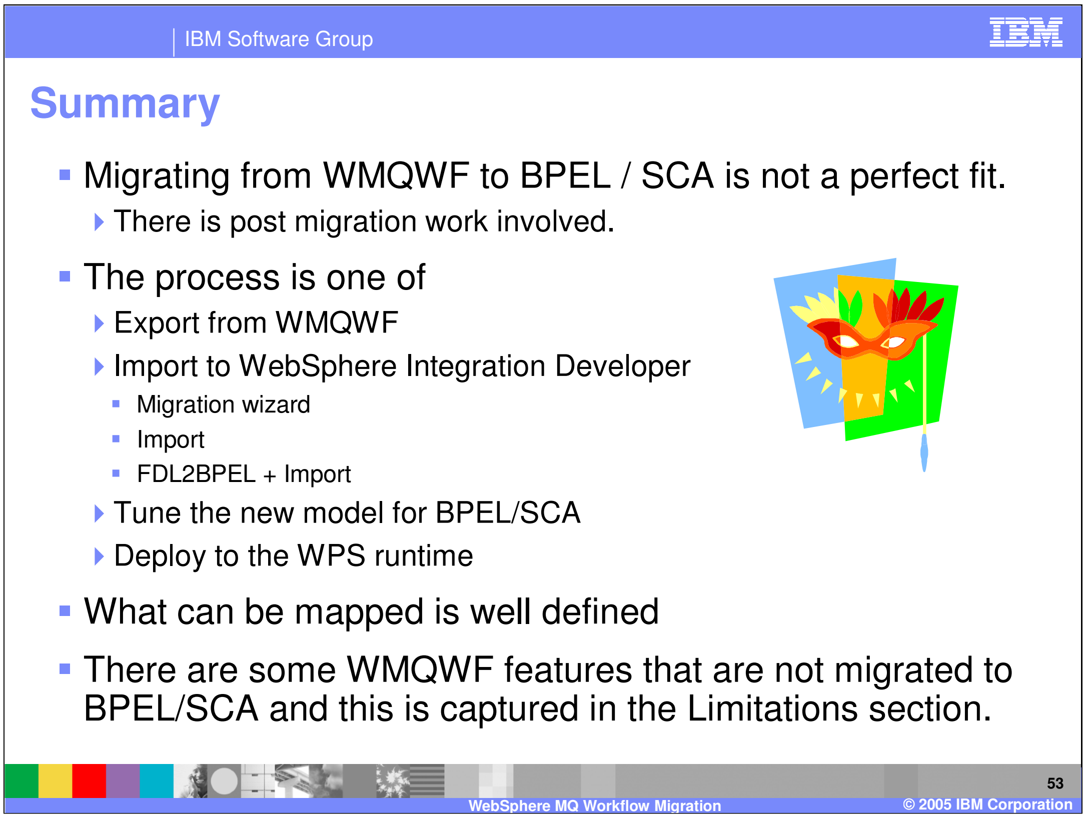
The concept of a „global container“ is not yet supported in BPEL. Nevertheless, the WMQWF migration generates a BPEL variable for it.



References

- WA73 Support Pak
  - http://www-1.ibm.com/support/docview.wss?rs=171&uid=swg24008362&loc=en_US&cs=utf-8&lang=en

The WA73 support pak contains the FDL2BPEL utility and documentation.



Summary

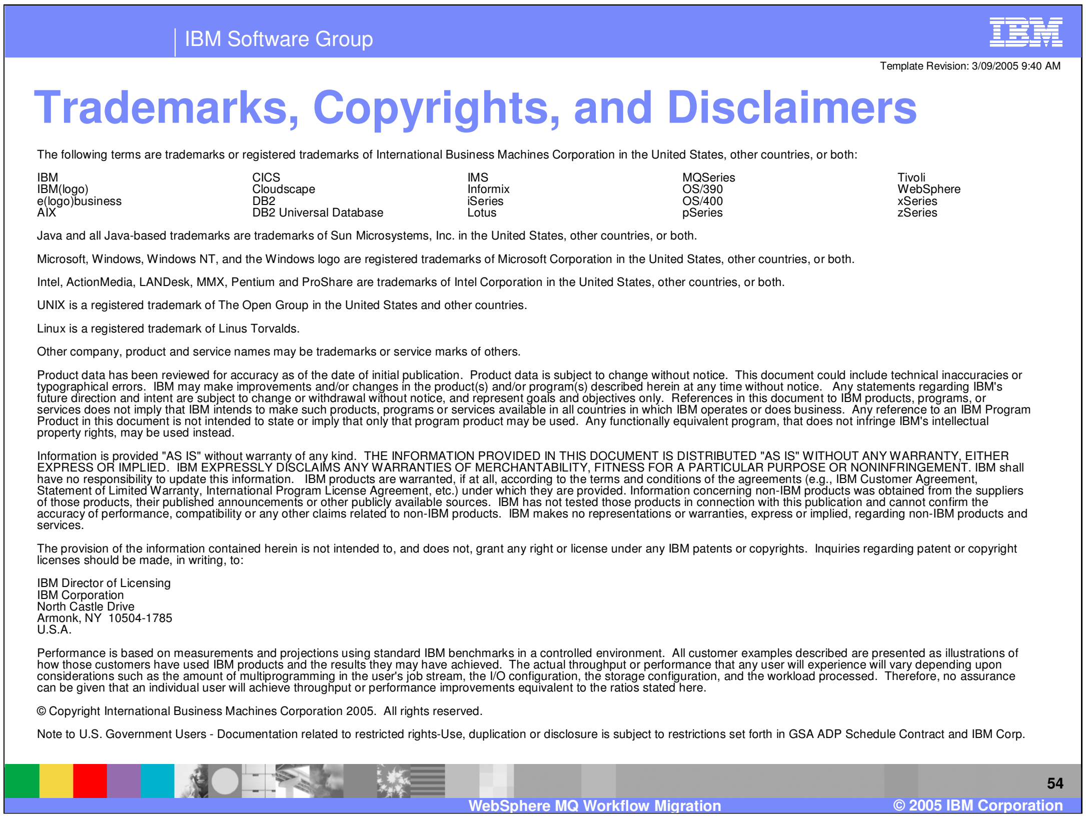
- Migrating from WMQWF to BPEL / SCA is not a perfect fit.
  - There is post migration work involved.
- The process is one of
  - Export from WMQWF
  - Import to WebSphere Integration Developer
    - Migration wizard
    - Import
    - FDL2BPEL + Import
  - Tune the new model for BPEL/SCA
  - Deploy to the WPS runtime
- What can be mapped is well defined
- There are some WMQWF features that are not migrated to BPEL/SCA and this is captured in the Limitations section.

To summarize, depending on the complexity of the WMQWF application there will be some post migration work involved.

The basic process is to export the application from the source system, import to the target system, edit, resolve errors and tune the application for BPEL / SCA and deploy the application to the WPS V6 runtime.

When importing the WMQWF FDL into WID, there are several options available, the Migration Wizard, Import **or** FDL2BPEL + Import.

Using FDL2BPEL + Import provides more control and flexibility over the process and is useful for debugging problematic migrations.



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
IBM(logo)
e(logo)business
AIX

CICS
Cloudscape
DB2
DB2 Universal Database

IMS
Informix
iSeries
Lotus

MQSeries
OS/390
OS/400
pSeries

Tivoli
WebSphere
xSeries
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.

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.