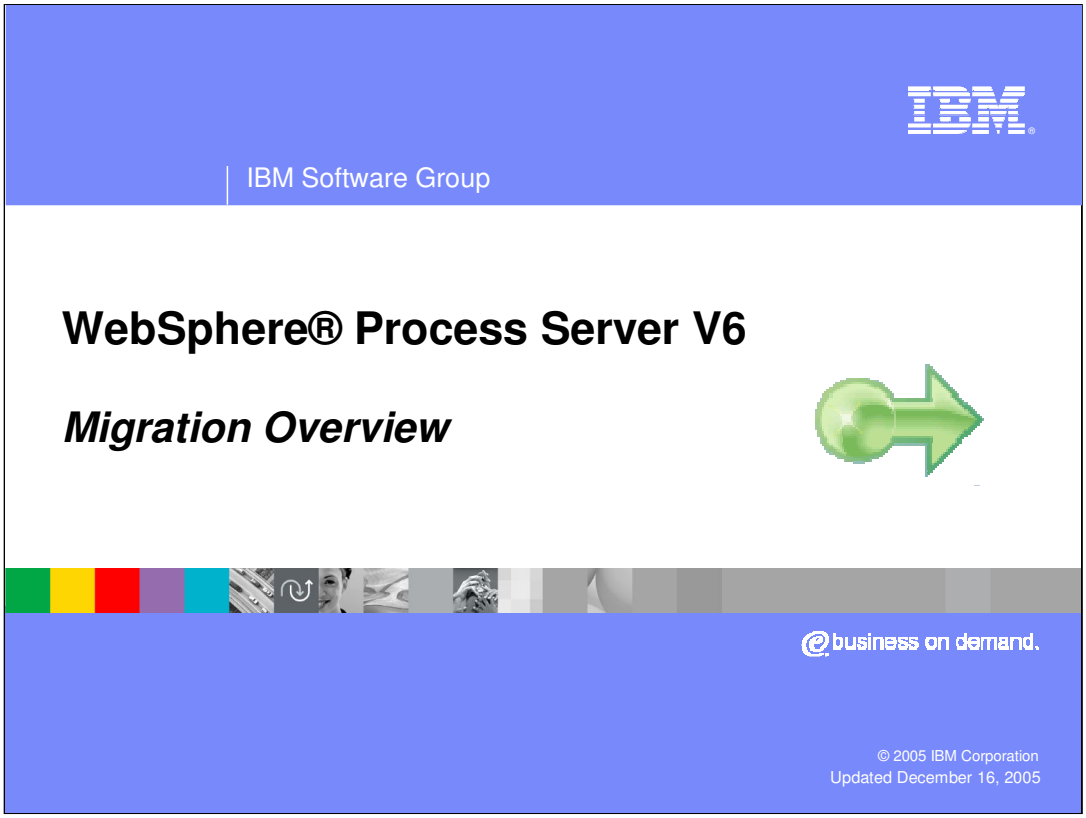
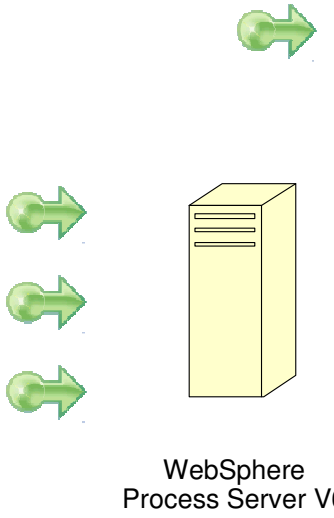



This presentation will cover migrating to WebSphere Process Server V6 from previous releases of WebSphere products.

# Goals

- High level overview of the migration utilities for
  - 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





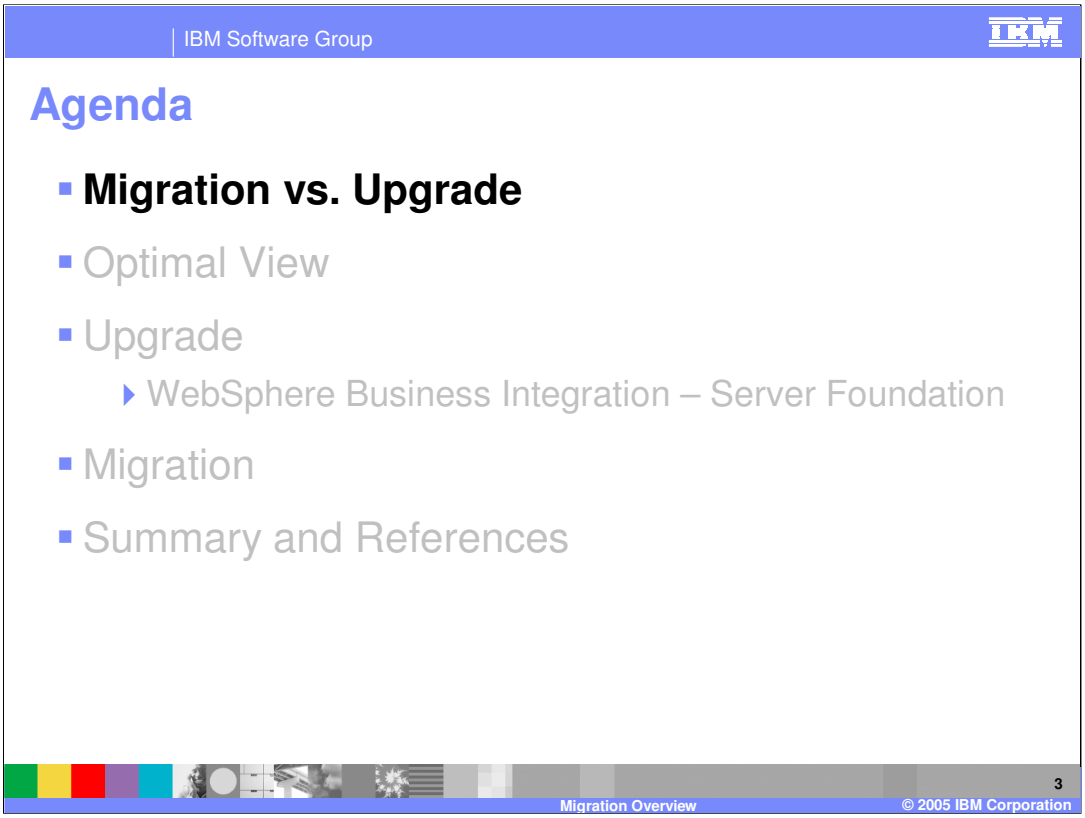


WebSphere Process Server is the merger of 3 existing product lines.

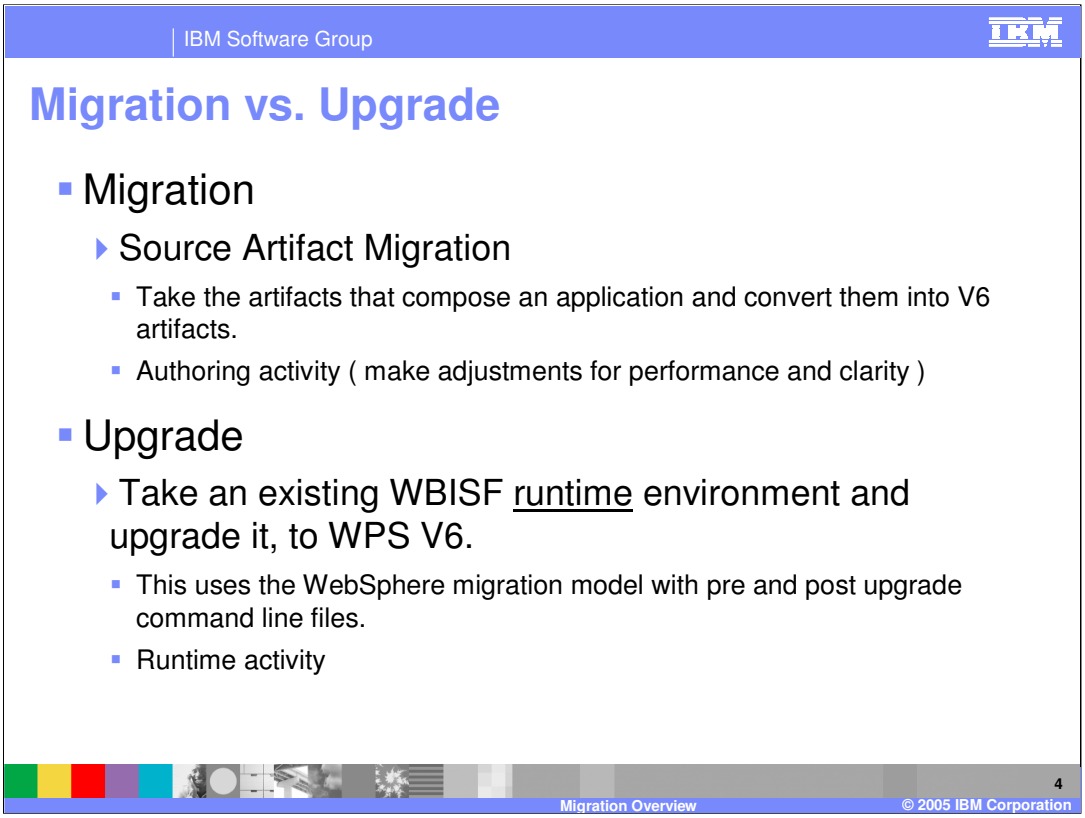
•WebSphere InterChange Server

•WebSphere MQ Workflow

•WebSphere Business Integration Server Foundation




The agenda for this presentation is to focus on the difference between migration and product upgrade.




Terminology is important and terms are often interchanged, leading to confusion. The terms Migration and Upgrade are defined here for clarity.

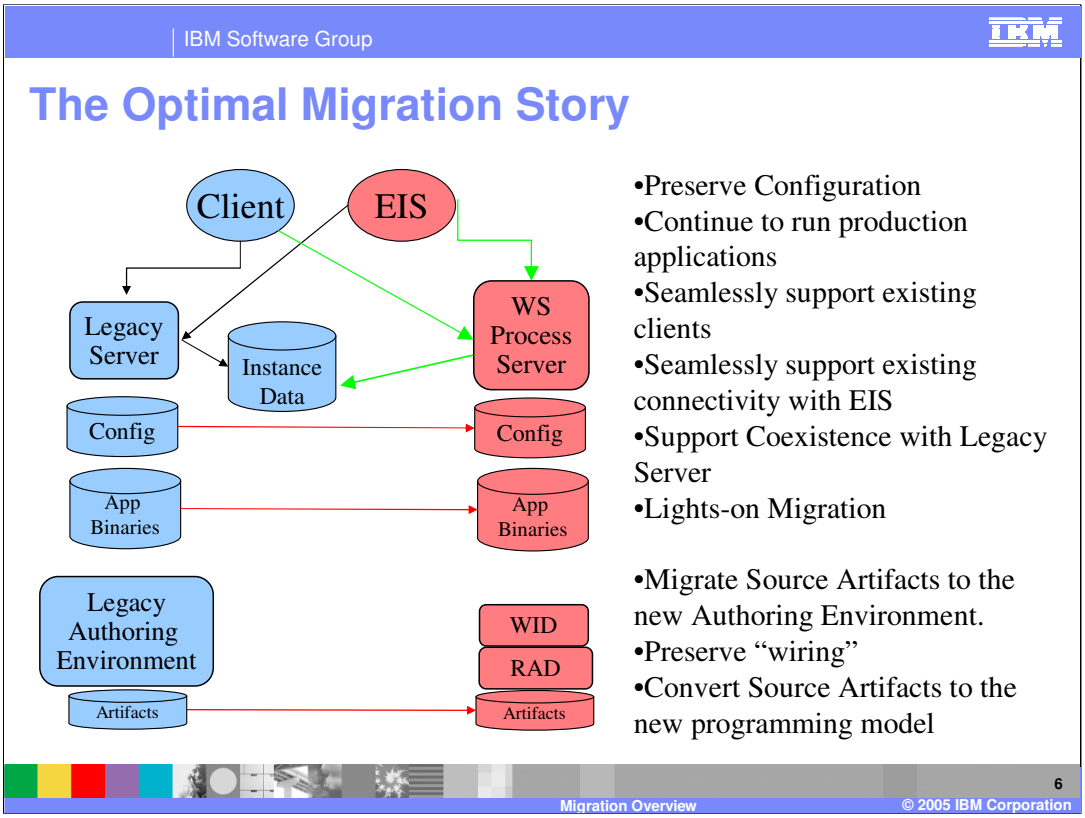
Note the emphasis on authoring as opposed to runtime environments. Source artifact migration is an activity that occurs during development using the WebSphere Integration Developer (WID) V6, while Upgrade is an activity specific to existing WBI SF runtime installations.

# Agenda

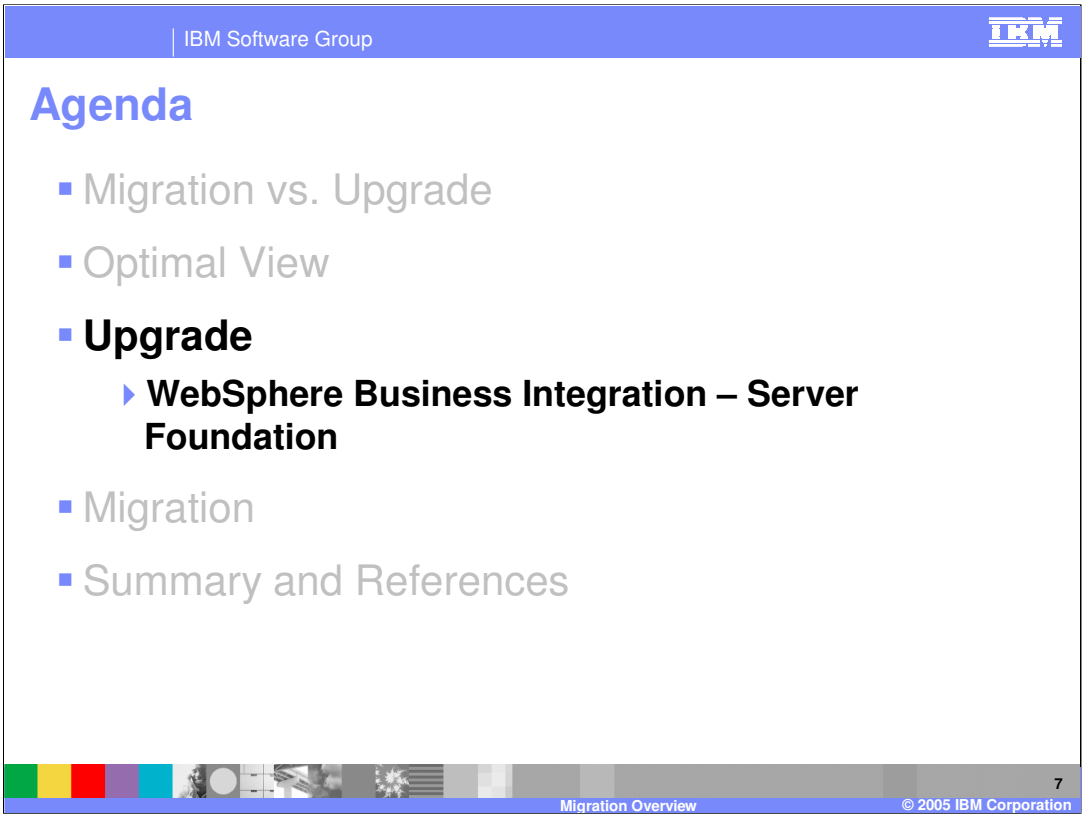
- Migration vs. Upgrade
- **Optimal View**
- Upgrade
  - WebSphere Business Integration – Server Foundation
- Migration
- Summary and References



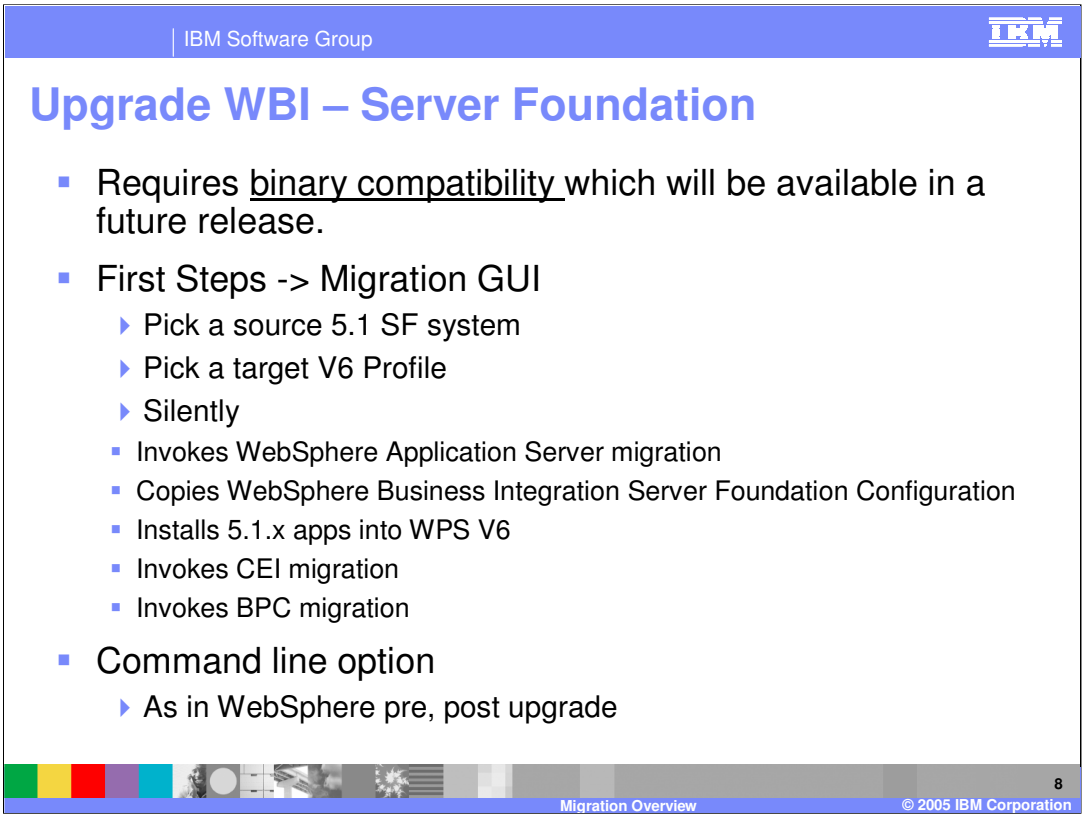

This section will provide an example of the optimal migration scenario.




Shown here is the ideal migration scenario. The existing systems are depicted on the left in blue. Application binaries and authoring artifacts are all moved seamlessly to the new environment, everything runs as is and the existing client applications continue to function with the existing applications as well as the new applications and the new client applications are able to work with the previously existing applications. The existing application binaries are moved to WPS V6, along with the appropriate configuration information, which is converted if necessary.




WebSphere Server Foundation V5.1.1 is the only legacy system for which the upgrade path makes sense. The other two, WebSphere InterChange Server and WebSphere MQ Workflow, come from a different program base and thus the upgrade option is not applicable to those cases.




Upgrade takes an existing WBI SF 5.1.1 system and installs the applications into a WPS V6 runtime, making any necessary configuration changes in the process. In order for this to work, the WBISF v5.1.1 applications must be able to interoperate with the WPS v6 applications. This is called binary compatibility, which is currently not available.

When the upgrade utility becomes available there will be options to invoke it either as part of the “First Steps”, which is an application that runs when WPS V6 is first installed, or from the command line.

When using the command line option, there are two steps; pre upgrade and post upgrade, as with the WebSphere upgrade/migration utility. It will first scan the existing system and allow you to create a backup of the current configuration. Upgrade is performed next, after which there is a post upgrade step that will modify the configuration.

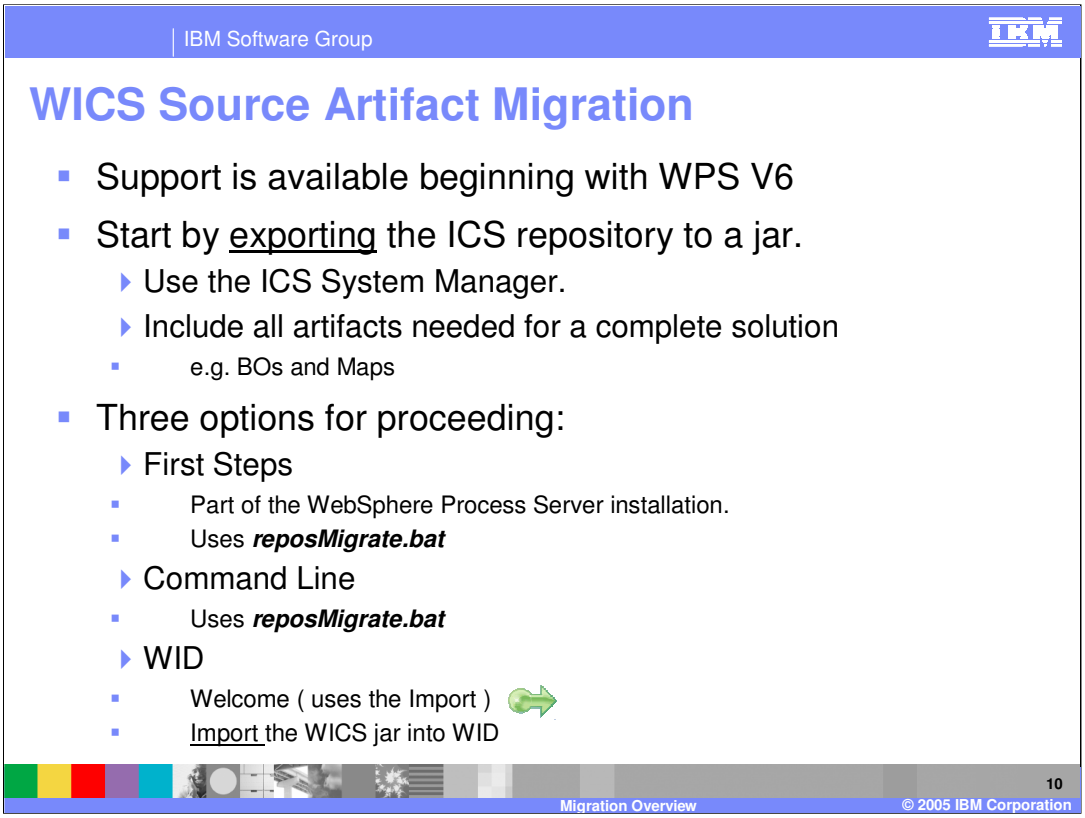
The ability to upgrade in this fashion is not available yet, but is planned for a future release.

# Agenda

- Migration vs. Upgrade
- Optimal View
- Upgrade
  - WebSphere Business Integration – Server Foundation
- **Migration**
  - **WebSphere InterChange Server**
  - WMQWF
  - WBISF
- Summary and References




This section will cover the migration of WebSphere InterChange Server.




Before beginning, you should review the SCA programming model.

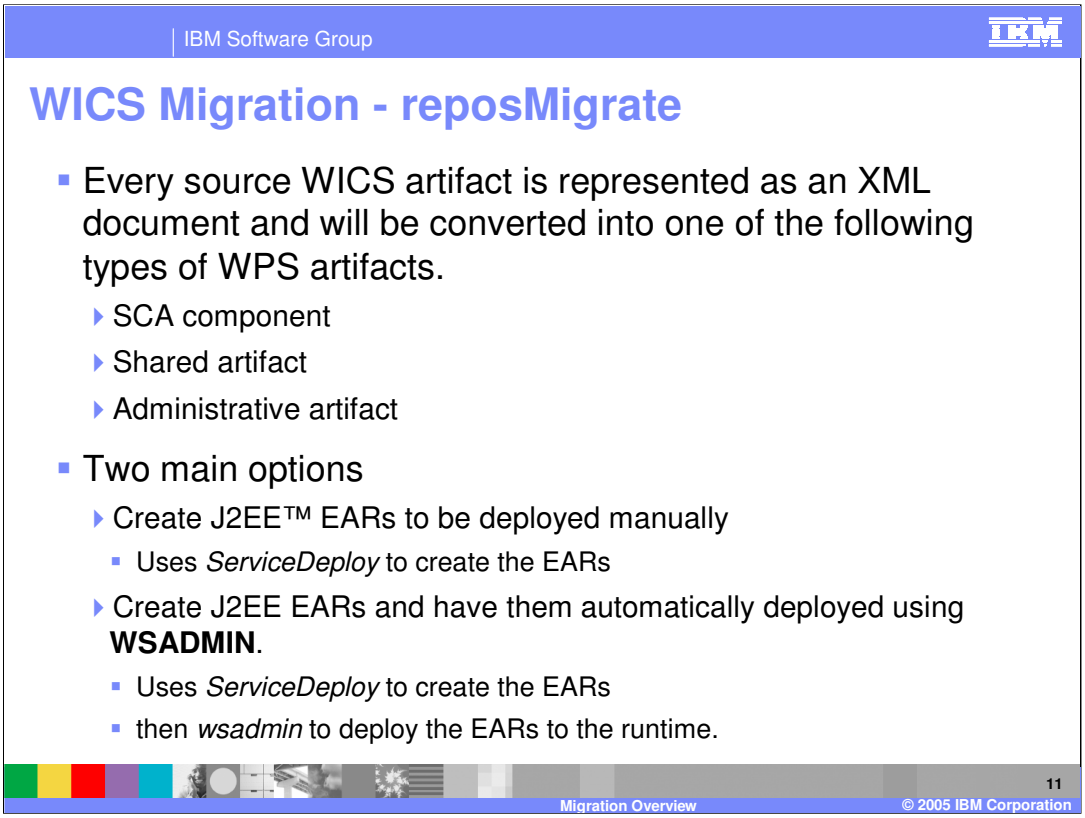
The process of source artifact migration for WebSphere InterChange Server (WICS) begins by exporting the artifacts from the WICS system to a jar file. Using the ICS System Manager, export all the artifacts that comprise a complete solution. That is to say, get everything that is referenced so that there will be no unresolved references when importing to WPS V6.

The next step is to import the jar into WebSphere Integration Developer (WID) V6. There is a special import type that will recognize the artifacts and make the necessary conversions, creating new SCA artifacts.

It is recommended that WID be used initially. This will provide the opportunity to understand how the conversions are made and to become familiar with the SCA components that are generated. Once a thorough understanding is achieved, the command line approach can be used to automate the process.

The migration can also be done from the “First Steps” application or from the command line using the reposMigrate.bat.

**reposMigrate.bat** is located in the bin directory of the WebSphere runtime installation and is described in the WebSphere Integration Developer ‘Help’ under the Migration topic



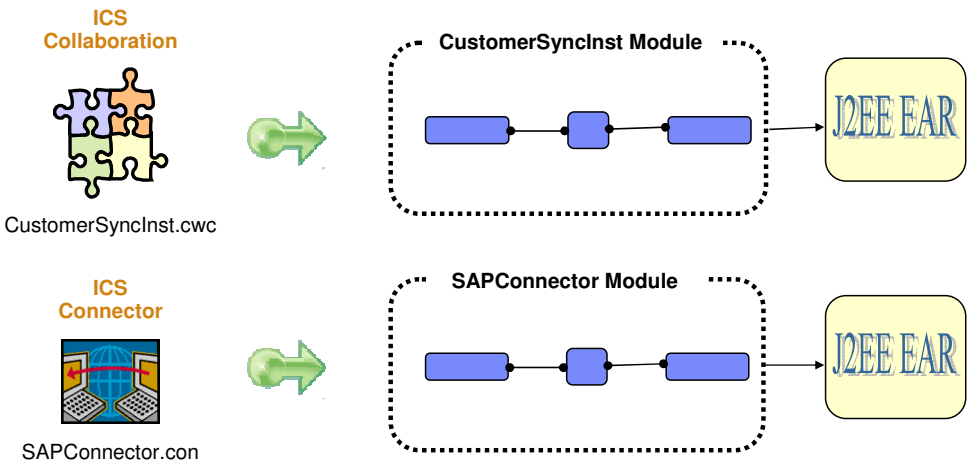

ServiceDeploy is the command utility for the WPS V6 used to compile and package the SCA and J2EE artifacts programmatically. The reposMigrate utility receives the WICS artifact jar and creates the SCA components and then invokes the ServiceDeploy utility to create the J2EE Ear. The J2EE EAR can then be deployed manually using the WPS Adminconsole, or programmatically using WSADMIN ( the WebSphere scripting language). Additionally the reposMigrate utility includes an option to invoke the WSADMIN utility after the Service Deploy has created the EAR.

All of these command line utilities can be further automated and managed using ANT in combination with a source code control system.

Before the EARs are deployed to the runtime, whether you do it automatically or manually, the WebSphere InterChange Server environment must be prepared by quiescing the system as described in the WID help.

## WICS Artifacts and reposMigrate

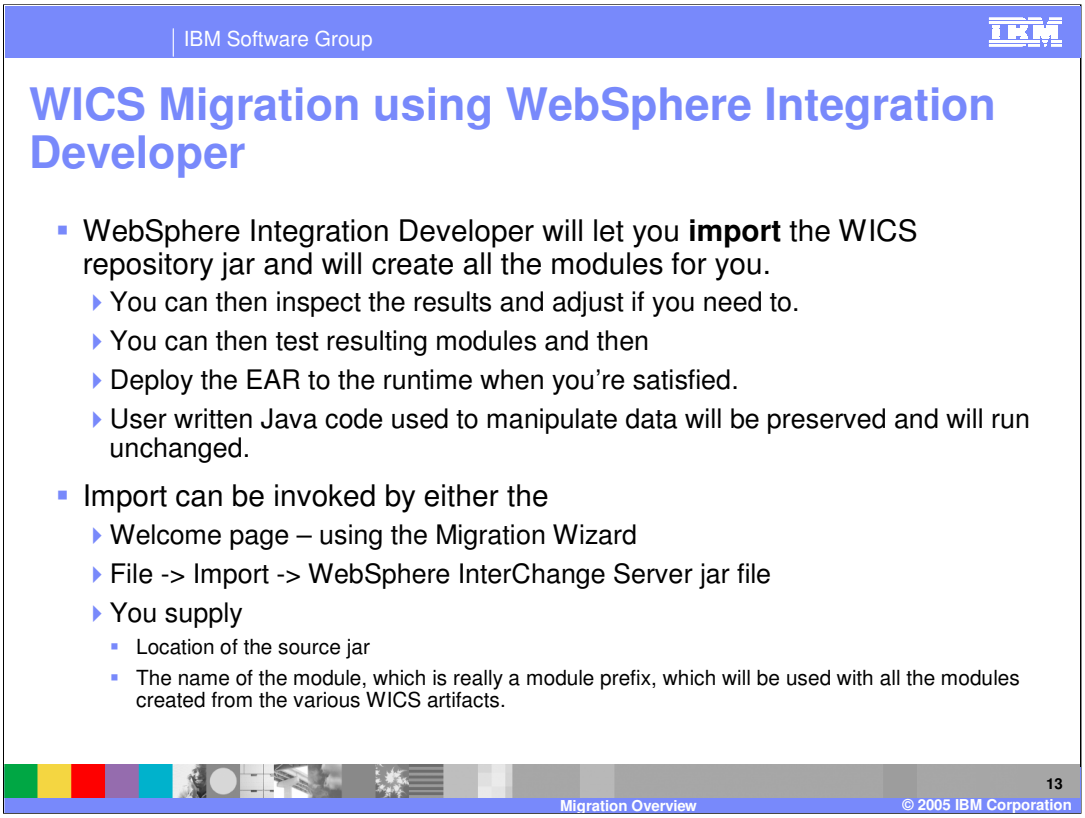
- Converts each artifact to its corresponding WebSphere Integration Developer artifact.







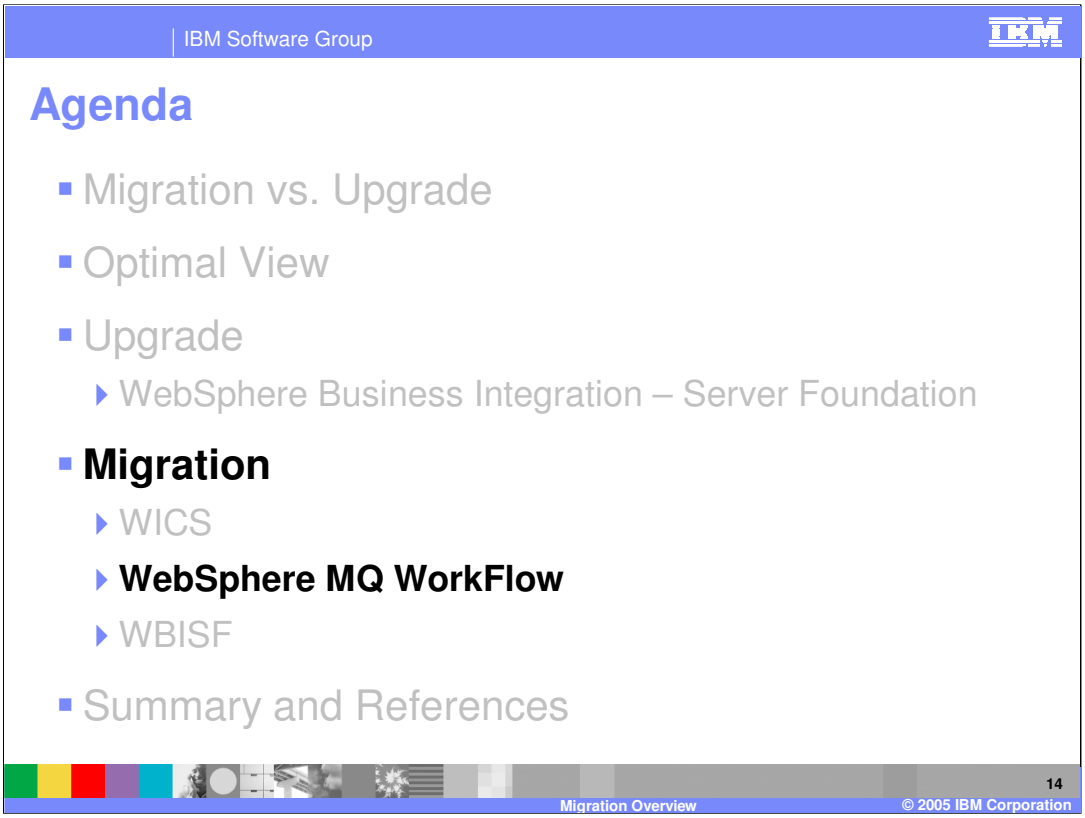
It is important to note that the WICS artifacts get mapped to SCA Modules and that SCA Modules get deployed as J2EE EARs. That is to say, that there will be an EAR for each of the WICS artifacts.




Importing the WICS repository jar into WID is the recommended approach in the beginning. This will provide the opportunity to easily inspect the results and make any necessary changes.

The WICS APIs have been deprecated, but to facilitate the migration an interface library is provided which will map the WICS API calls to WPS/SCA API calls. This will provide the user written Java code to run unchanged.

For a given WICS repository jar there will be many SCA modules generated. The module name supplied to the wizard will be pre-pended to the name of the artifact. For this reason it is recommended that the module name be short and distinctive.




The next item on the agenda is WebSphere MQ Workflow.

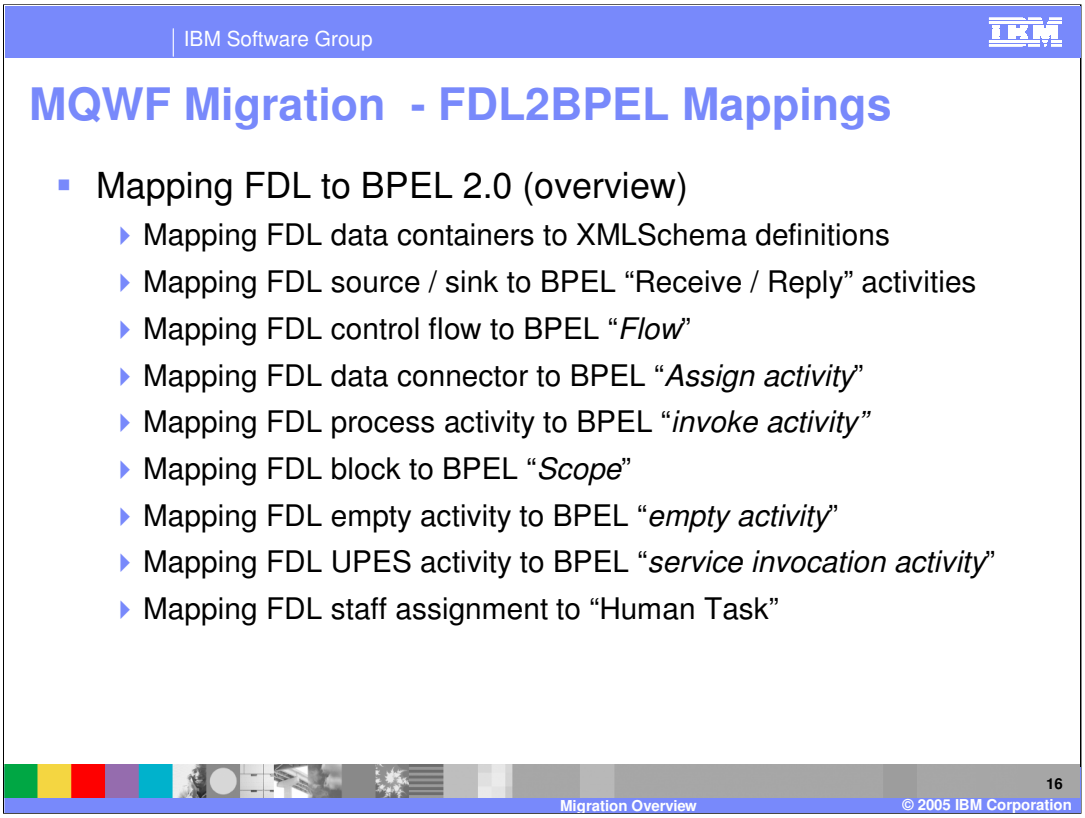
## WMQWF Migration

- Support is available beginning with WPS V6
- Start by exporting the FDL from WMQWF buildtime
  - Requires semantically complete FDL
    - e.g. export DEEP from WMQWF buildtime
- Use WebSphere Integration Developer (WID) to invoke the FDL2BPEL translation utility
  - Welcome ( uses the Import ) 
  - Import
- Optimize the generated output based on knowledge of WPS and BPEL.




Note that a command line version of the FDL2BPEL translation utility is available as part of support pak WA73

The recommended migration strategy is to use the migration tool to generate the first cut of the BPEL implementation, which will capture the flow, and then use this as the starting point for refinement and optimization of the BPEL implementation.




**FDL Classification Rules:**

In FDL, the invocation types are distinguished using the properties associated with the activities. In BPEL there are explicit kind-of activities such as *'empty', 'Human Task', and 'Service'.*

Many but not all of the MQ Workflow constructs have BPEL analogs. The most notable difference is in the area of the invocation types, as noted above.

# MQWF Migration - Control Flow

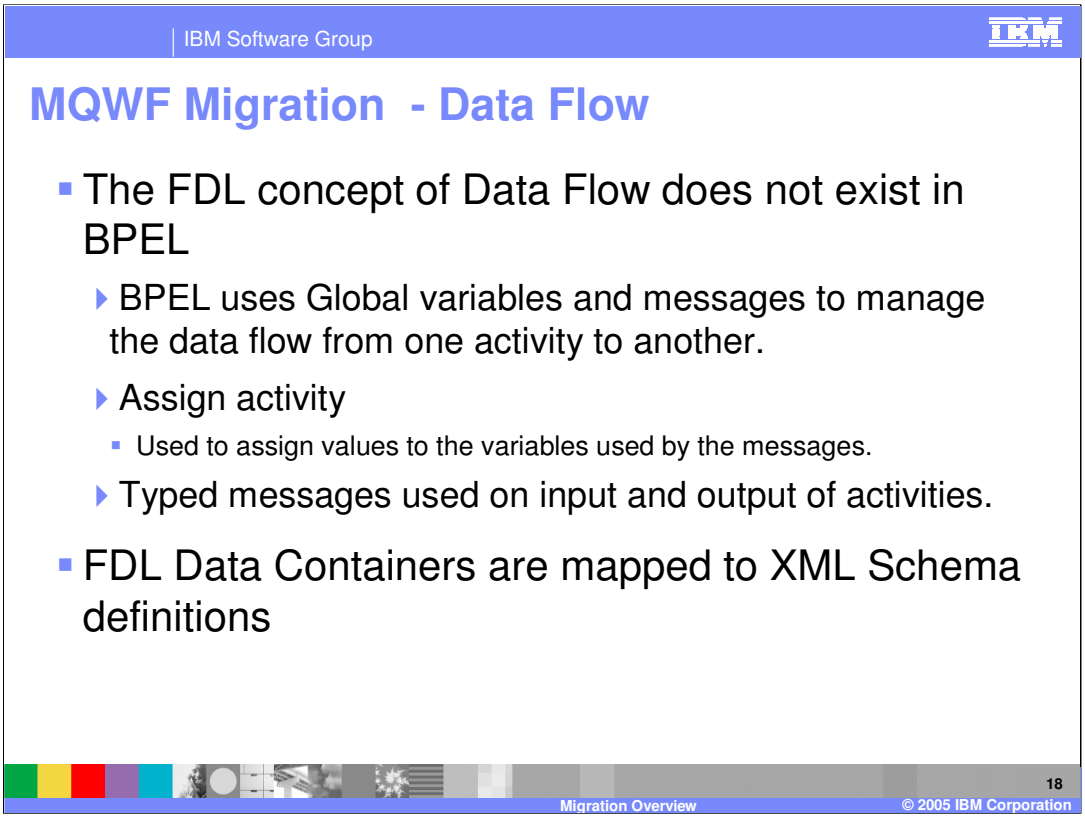
- BPEL link element connects activities

| 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) |



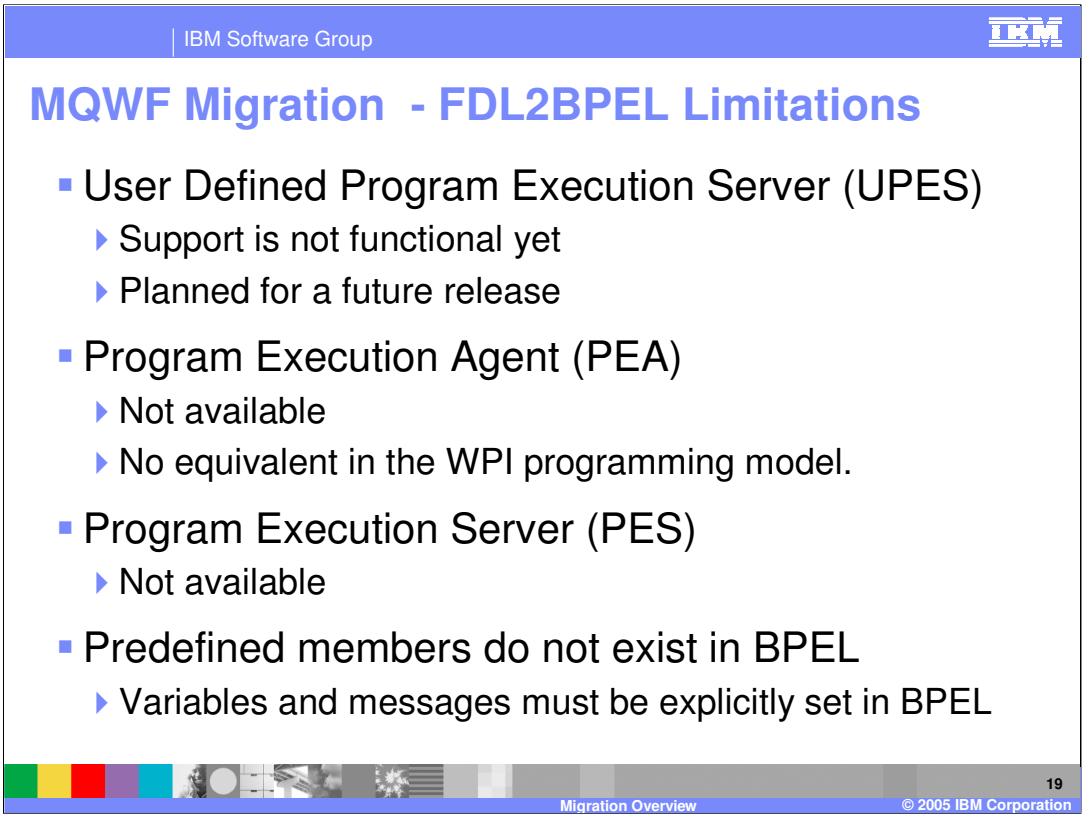

In BPEL, the Link element is represented graphically as the line joining 2 activities. The BPEL link has the '*transition condition*' and the '*join condition*' modifiers that control the flow to provide the same functionality as the '*transition condition*' and the '*start condition*' in MQWF.

MQWF uses an implied loop to keep the activity executing until the exit condition is met. BPEL uses a while construct that must be explicitly specified.




The area of Data Flow is where MQWF and BPEL differ greatly. With BPEL there is no explicit data flow and data are stored in global and local variables and can be passed from one activity to another as a parameter when invoking an activity. The data are implemented as SCA Business Objects (BOs) in the WPS V6 BPEL business processes.

With FDL the data are defined with the data containers and in SCA/BPEL they are defined using XML schema definitions.




***Best Practices*** to use when building WMQWF-based workflow applications are listed below.:

•Runtime clients: Use the browser based Web client of WMQWF. Do not use the standard, ActiveX- based Windows runtime client and do not 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 using the PES invocation mechanism and PES data mapping. Do not use the PES container API for such activities.

•API: Only use the Java API

# Agenda

- Migration vs. Upgrade
- Optimal View
- Upgrade
  - WebSphere Business Integration – Server Foundation
- **Migration**
  - WICS
  - WMQWF
  - **WebSphere Business Integration Server Foundation**
- Summary and References

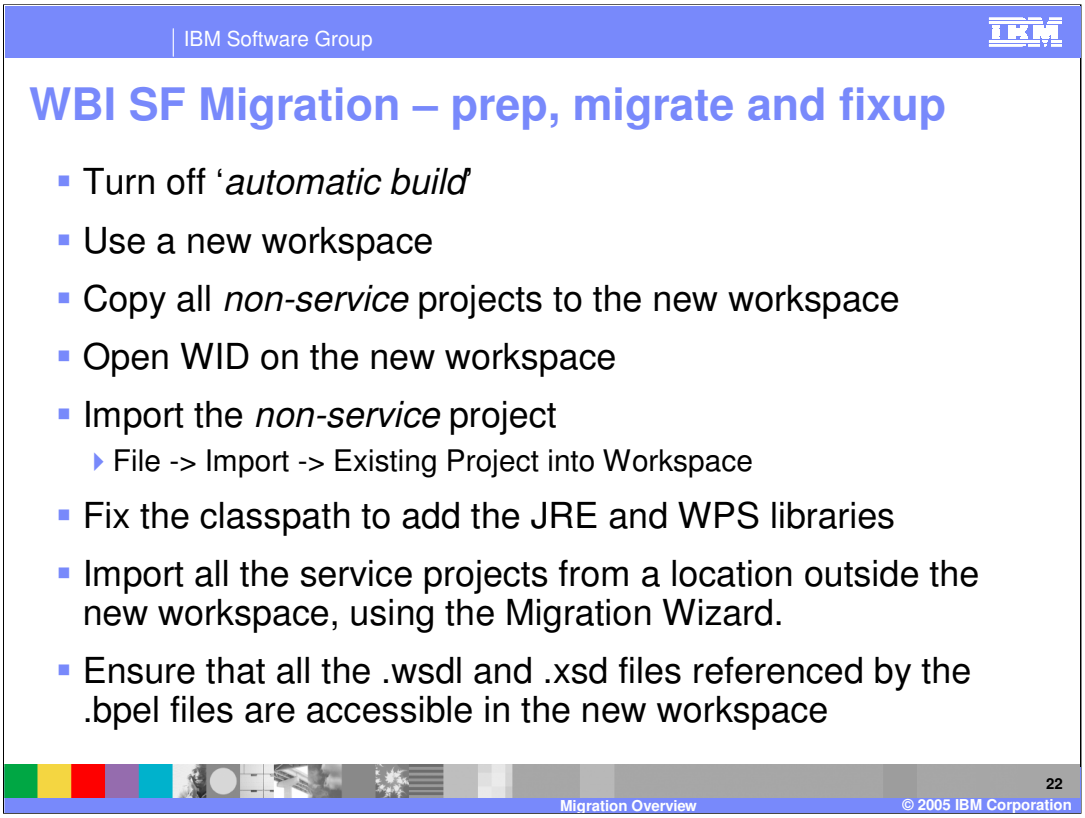



This section will cover WBI SF v5.1.1


Migrating from WBI SF v5.1.x requires a preparation step, which involves manually moving any dependent utility applications or jars, meaning the non-service projects. This is described in more detail on the next slide. The migration wizard primarily migrates business process artifacts, but it may be run for any v5.1 service project. It migrates the business process artifacts (including the BPEL Java™ snippets where possible) from the WBISF V5.1 to the WPS V6 supported BPEL specification and creates an SCA component for each migrated business process. Since it is only operating against the service projects, the non-service projects must be imported into the workspace manually before the wizard is run. In other words, all of the dependent utility applications or jars must be in place before running the wizard.

**Hint:** Keeping the SCA module the same name as the 5.1 service project will reduce the amount of post migration fixup due to the classpath and project dependencies.

The Migration Wizard only handles source artifacts and not application binaries. See the description of the limitations in the WID on-line help.




This is a high level outline of what must be done to migrate existing WBI SF V5.1.1 artifacts to WPS V6.

Details for completing these steps can be found in the WebSphere Integration Developer help, in the preparation step for the WBISF migration.

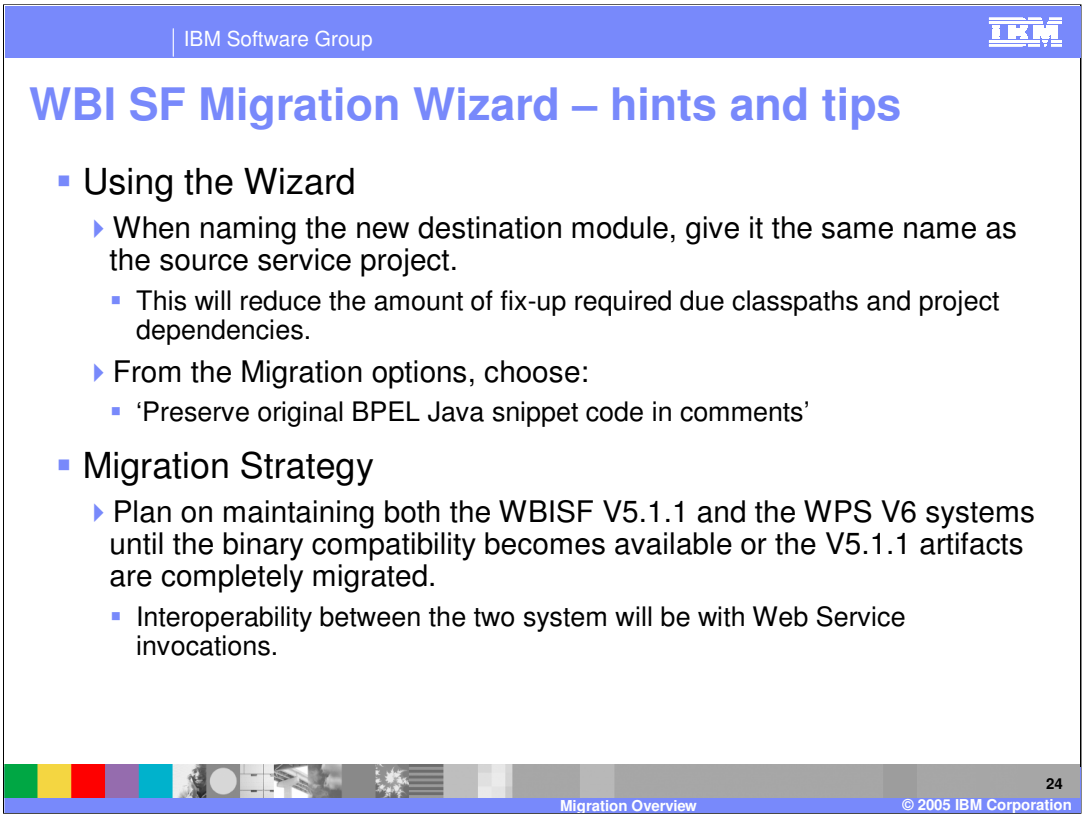
## WBI SF Migration Wizard

- 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 WBISF 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 WBISF ( if necessary)




This slide outlines the steps performed by the WBI SF migration wizard. The link below provides detailed information related to migrating from WBI SF V5.1.x to WPS V6.

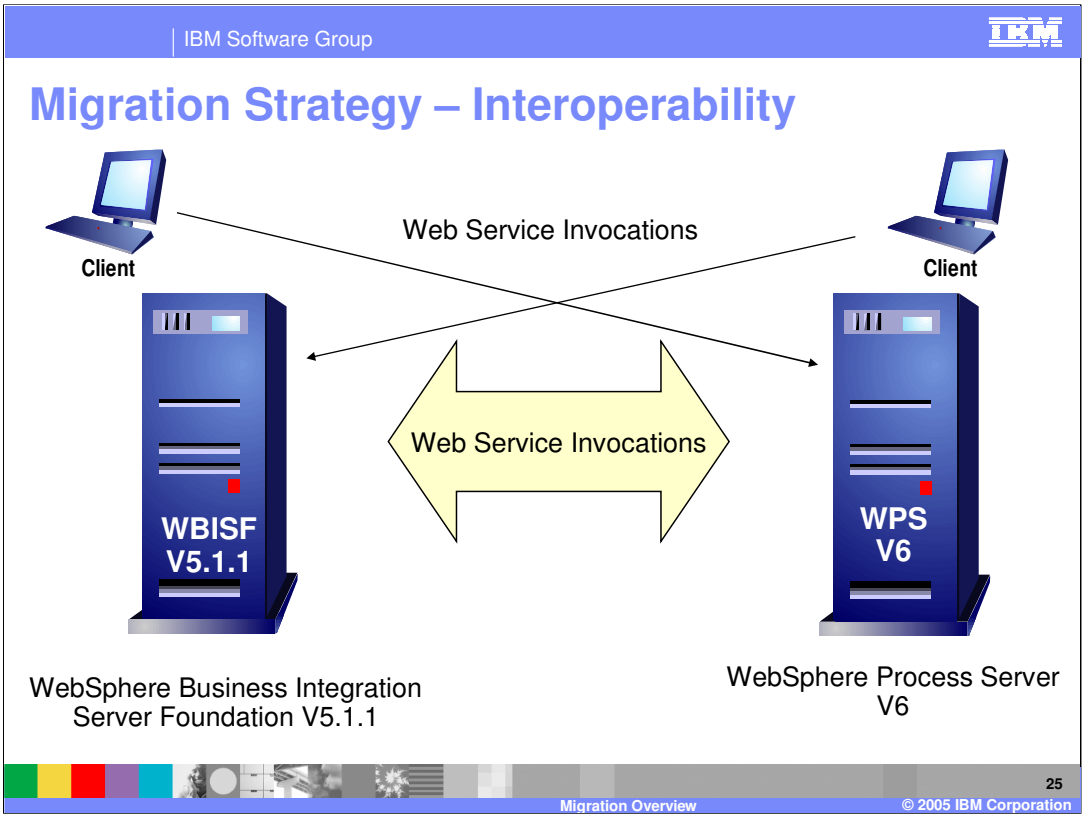
http://www.ibm.com/developerworks/websphere/library/techarticles/0509_iyengar/0509_iyengar.html




When using the migration wizard, it is necessary to preserve the BPEL Java snippets because WBI SF v5.1 uses the Web Services Invocation Framework (WSIF) to perform service invocations, where data sent are stored in a WSIFMessage, whereas the WebSphere Process Server programming model uses SCA to perform service invocations, where data being sent are stored in a Business Object (BO). Each model has a different set of APIs for invoking a service and accessing and manipulating the data.

The BPEL Java snippets are specific to the programming model and data type used by the underlying platform. Although the migration wizard migrates the v5.1 BPEL Java snippets as much as possible, manual rework could be required for Java code that manipulates complex data types. This is why it is a good idea to preserve the existing Java snippets as comments in the migrated Java snippet so you can see exactly what the v5.1 snippet did, enabling you to manually fix-up the complex snippets after migration.

Migrating existing applications can be very complex and requires detailed plans for cutting over from one system to the next with considerations for dual maintenance and development until binary compatibility becomes available.




Interoperability between the two different systems is achieved using Web Services.

It is possible that some WBISF v5.1.1 applications are never migrated but still require interoperability with WPS V6.

If all WBISF 5.1.1 applications are migrated to WPS V6, then the WBISF 5.1.1 system can be retired.

## WBI SF Migration – Inbound/Outbound Services

- After the Migration Wizard is finished, there are still several manual steps required.
- Wire up the new components using the Assembly Editor.
  - 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 WBISF business process.
    - EJB™ (always selected)
    - IBM® Web Service (SOAP/JMS)
    - IBM Web Service (SOAP/HTTP)
    - Apache Web Service (SOAP/HTTP)
    - JMS




With WBI SF 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 many possible combinations, the Wizard does not attempt to guess what the user wants to do and leaves this up to the user to complete. Therefore, its imperative to have a solid understanding of the WPS/SCA programming model before attempting the source artifact migration.

# WBI SF Migration - Limitations

- Must Read!
- Based on Service Projects and **not workspaces**
  - One Service Project at a time
- Does not migrate application binaries
  - Only source artifacts found in the WBISF Service Project
- Multiple replies for the same operation, is not supported
- 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
  - Read-only BPEL variables
  - Many-valued primitive properties in complex types
  - Instantiation of generated classes representing complex types

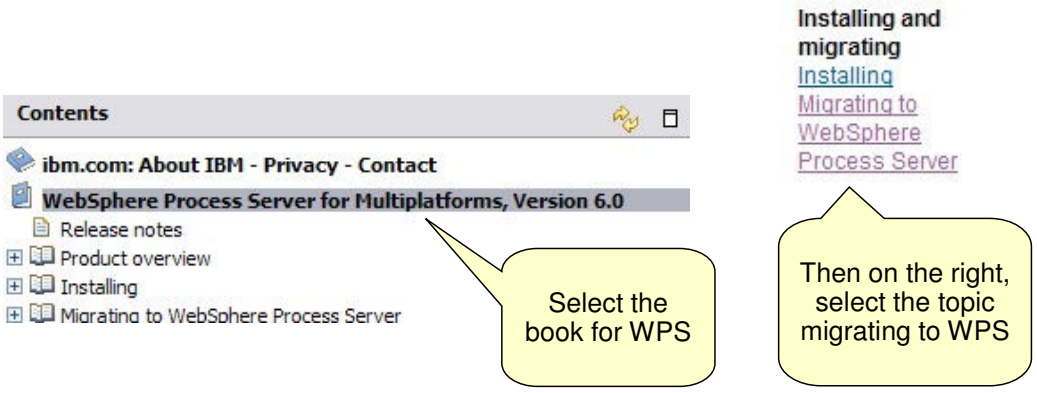



Details can be found in the WebSphere Integration Developer help, in the Limitations step for the WBISF migration.

# Deprecated Features

- Infocenter
  - There are many deprecated features to be aware of which are documented in the InfoCenter.

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







Deprecated features can be found in the Information Center.

# Agenda

- Migration vs. Upgrade
- Optimal View
- Upgrade
  - WebSphere Business Integration – Server Foundation
- Migration
- **Summary and References**




This section will provide a summary and references.

## Summary

- Source artifact migration is available today with WebSphere Process Server V6.0 and beyond for:
  - WebSphere InterChange Server (WICS)
  - WMQWF
  - WBISF
- Post migration ‘fixup’ will be required.

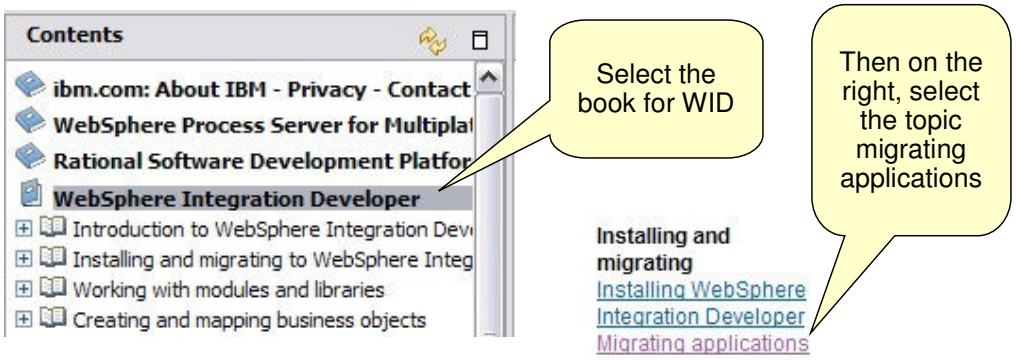



With WPS V6 and beyond, there is the capability for source artifact migration from all three converging products, WICS, WMQWF and WBISF. The source artifacts are converted to the WPS / SCA programming model as they are imported into the WID authoring tool. As with most migration efforts, when converting from an existing model to a new model, 100% conversion is not possible. The areas that cannot be automatically converted must be manually converted by the Integration Developer as a post migration task.

# References

- Infocenter

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







You can find details concerning all three migration paths in the Information Center.

# 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.



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.