



IBM Software Group

## WebSphere® Process Server V6



### *WebSphere InterChange Server Migration*



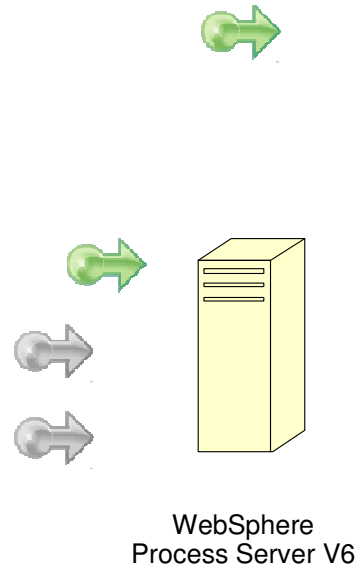
@business on demand.

© 2005 IBM Corporation  
Updated December 14, 2005

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

## Goals

- Detailed discussion of the migration utilities for WebSphere InterChange Server (WICS)
  - ▶ 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

This presentation will discuss the details of the WebSphere Interchange Server migration.

## Agenda

- **Introduction**
- **Overview**
- **Exporting the application from WICS**
- **Importing the application to WID**
- **Recommendations**
- **Troubleshooting**
- **References**



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

## Introduction - Before You Begin

- There are limitations and workarounds 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 InfoCenter
- Review the “*Best Practices for WebSphere InterChange Server migration process*” in the WID 6.0.1 InfoCenter.
- Upgrade to WID/WPS 6.0.1
- Become familiar with the new WebSphere Process Server SCA programming model by doing tutorials and exercises.



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 WID 6.0.1 InfoCenter

Upgrade to WID/WPS 6.0.1

Become familiar with the SCA programming model

## Introduction – Before You Begin

- This discussion will address *source artifact migration*.
- There are two approaches to choose from.
  - ▶ WebSphere Integration Developer V6 ( authoring tool )
  - ▶ WebSphere Process Server V6 ( runtime )
- Recommended: Begin with the WebSphere Integration Developer V6 (WID) Migration Wizard
  - ▶ Provides an environment for learning and understanding the process with the ability to make changes before deploying the migrated application.



Migration utilities are provided by both the WID authoring tool and the WPS runtime.

The recommended approach is to do the initial migrations using WID, and as the migration process is understood and refined, automate it using the runtime command line tools, reposMigrate, ANT and WSADMIN.

IBM Software Group IBM

## The InfoCenter

- InfoCenter

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

Contents

- ibm.com: About IBM - Privacy - Contact
- WebSphere Process Server for Multiplatforms
- Rational Software Development Platform
- WebSphere Integration Developer**
  - Introduction to WebSphere Integration Developer
  - Installing and migrating to WebSphere Integration Developer
  - Working with modules and libraries
  - Creating and mapping business objects

Select the book for WID

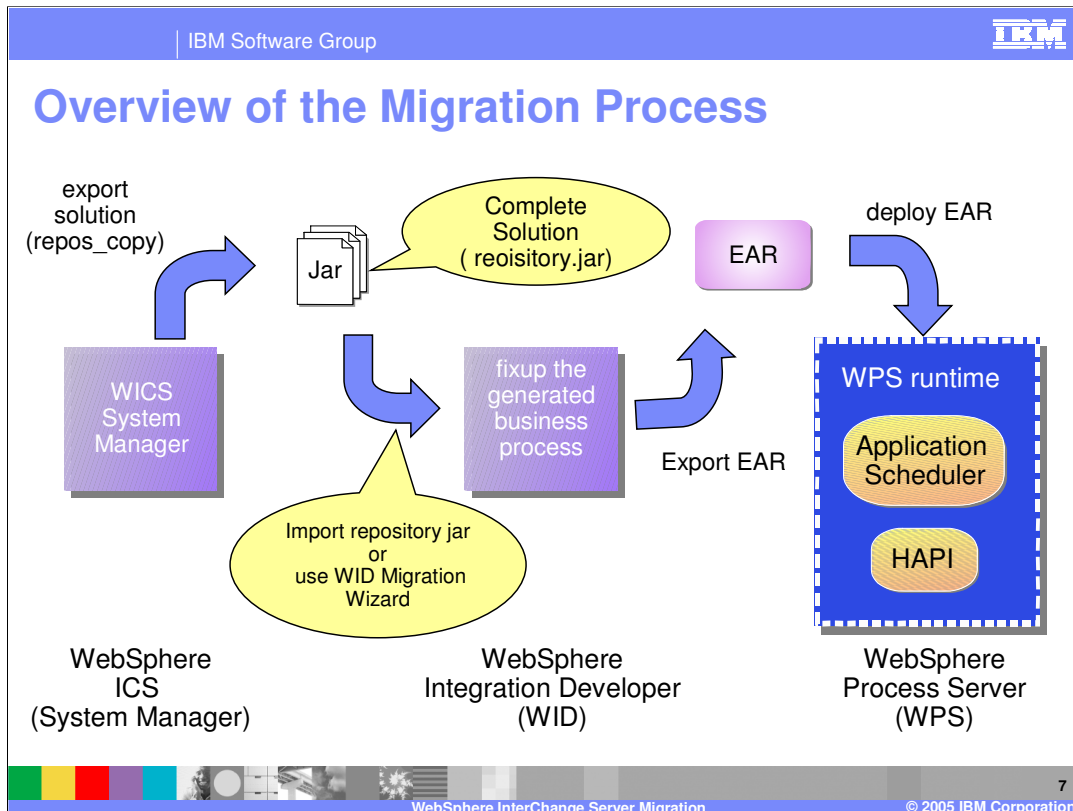
Then on the right, select the topic migrating applications

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

WebSphere InterChange Server Migration 6  
 © 2005 IBM Corporation

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

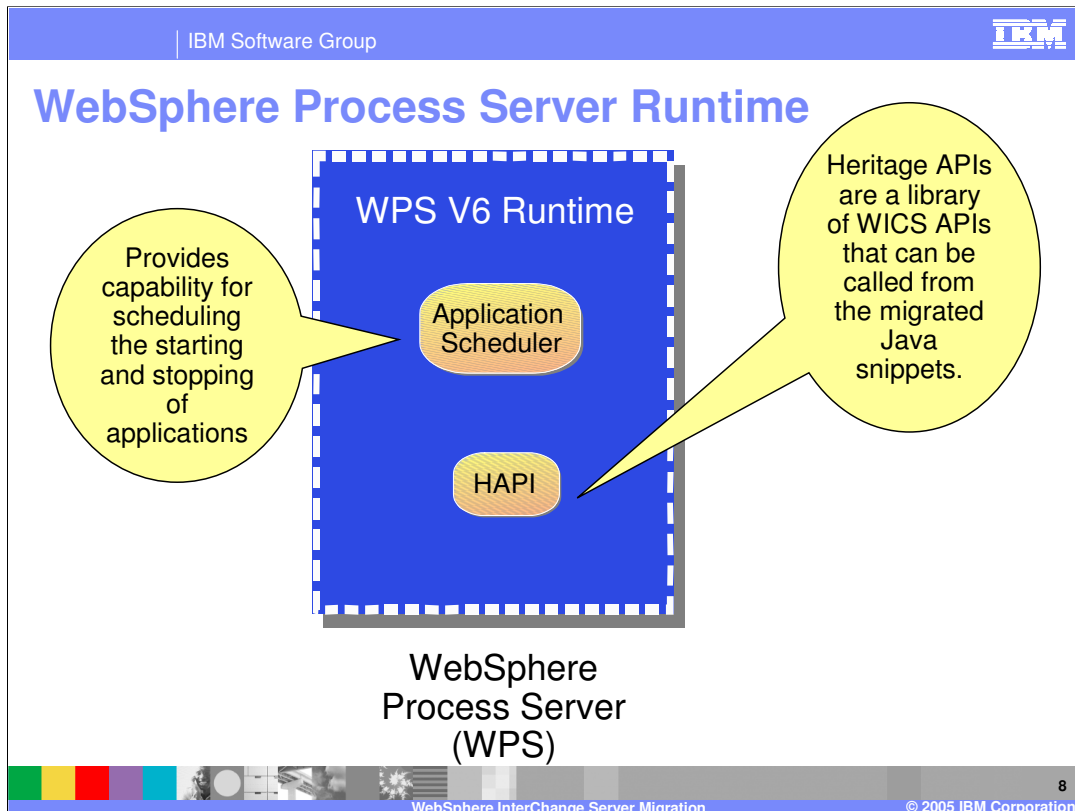
A PDF version of the migration guide is also available from the Welcome Page of WebSphere Integration Developer V6.



From a high level perspective, the migration process consists of the following steps: export from the source system, import to the target system, edit, resolve errors and tune the migrated artifacts for BPEL / SCA and deploy the application to the WPS V6 runtime.

Beginning with the Export:

- Export the complete WICS solution from the WICS environment using the WICS System Manager or the repos\_copy command.
  - This requires a complete solution, meaning all the dependent artifacts are in the repository jar.
- Import the repository jar into WebSphere Integration Developer V6 (WID)
  - To import the repository jar into WID, use either the Migration Wizard or import the jar directly using the File → Import menu.
  - The import utility will convert the WICS artifacts to WPS V6 artifacts and create the appropriate modules as needed. The conversion will be described in subsequent slides. (**Note:** the Wizard will call the import utility transparently.)
- Once the WPS V6 source artifacts have been created in WID, it will be necessary to review the artifacts and based on the errors identified and the known limitations, fix up the Business Processes to eliminate any errors.
- Once the fix up is complete, export the EAR and deploy it to the WebSphere Process Server V6 runtime (WPS)



There are 2 new components in the WPS V6 runtime introduced specifically to support WICS migration.

The **Application Scheduler** provides the capability for scheduled starting and stopping of applications. Refer to the WebSphere Process Server for Multi-platforms Information Center and Search on Scheduler for more information about the Application Scheduler.

A System Administrator can interact with the Application Scheduler using the WPS V6 administrative console. Additionally, you can generate scheduler entries during the migration of a WebSphere InterChange Server repository that includes WebSphere InterChange Server scheduler entries. Use the Application Scheduler panel in the administrative console to administer these migrated scheduler entries as well.

In an Network Deployment environment, the Application Scheduler is automatically installed for every managed server and cluster member created - no additional action is needed.

In a stand-alone server environment, the Application Scheduler is optional. While creating the stand-alone server profile, you select a check box to configure and install Application Scheduler on that server.

The **Heritage APIs** are provided to facilitate the migration process.

These APIs are provided only to support migrated WebSphere InterChange Server applications until they can be modified to use the new Process Server APIs. The WebSphere InterChange Server APIs are all deprecated and will be removed in a future release.

Refer to the WebSphere Integration Developer V6.0.1 Information Center For a detailed list of the Heritage APIs.



## WICS Source Artifact Migration: Export

- Make sure that WICS is at 4.2.2 or greater
- Start by exporting a part of the WICS repository that represents a complete solution.
  - ▶ Keep the solution as small as possible while maintaining a complete and self contained solution.
  - ▶ Include all of the artifacts required for the complete solution.
  - ▶ Use the WICS System Manager
    - The `repos_copy` command.



Before beginning a migration, make sure the version of WICS is supported by the migration utilities.

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

## WICS Source Artifact Migration: Import

- There are several ways to create the WPS artifacts from the WICS repository jar.
  - ▶ WID V6 authoring tools
    - Migration Wizard from the Welcome Screen 
      - Uses the special “WebSphere InterChange Server” import type
    - Import
      - File → Import menu
      - Uses the special “WebSphere InterChange Server” import type
  - ▶ WPS V6 runtime tools
    - Command Line
      - Uses **reposMigrate.bat**
    - First Steps
      - Migration GUI
      - Uses **reposMigrate.bat**



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


The **Migration Wizard** provides a quick and easy way to launch the import utility and setup the new project at the same time. It can be invoked from the WebSphere Integration Developer Welcome Screen.

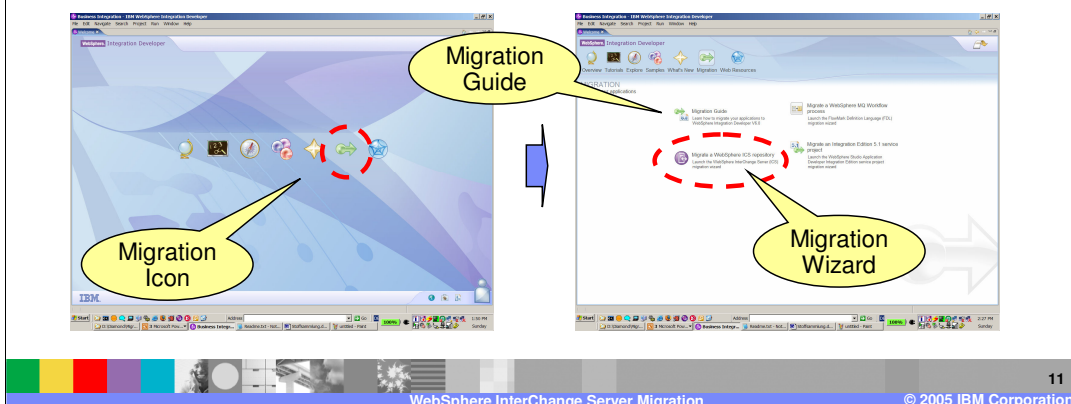
Again, its recommended that WID be used initially. This will allow you the opportunity to understand how the conversions are made and to become familiar with the SCA components that are generated. Once a thorough understanding is achieved, the command line approach can be used to automate the process.

The “**First Steps**” and **reposMigrate.bat** are both tools provided by the WebSphere Process Server V6 runtime. The “First Steps” application is presented at the completion of the WPS V6 runtime installation.

**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. It will do the source artifact conversions and can also be used with the runtime ServiceDeploy utility to create and package the EAR, bypassing WID altogether.

## 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 ICS Repository*” option.



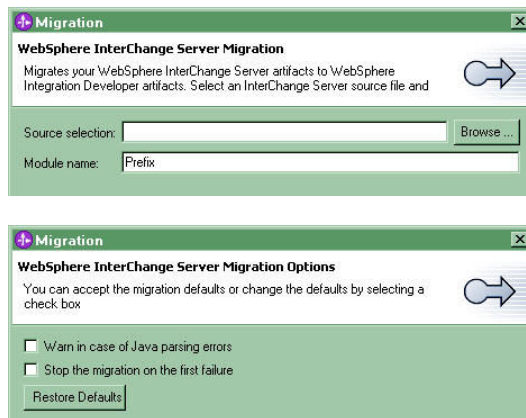
The Migration Wizard is available from the Welcome page of the WebSphere Integration Developer V6.

To get to the Welcome page in WebSphere Integration Developer, select **Help** → **Welcome** menu.

While on the Welcome page, notice the link to the migration guide, which is available in PDF format for easy printing.

## Using the Migration Wizard

- WebSphere InterChange Server repository jar is used as the input source.



- Source Selection
  - Same as `-i` param in `reposMigrate`
- Module Name
  - Prefix given to all generated modules i.e. (Prefix\_<ModuleName>)
- Warn in case of Java parsing errors
  - Same as `-wi` param in `reposMigrate`
- Stop the migration on the first failure
  - Same as `-fh` param in `reposMigrate`

The first panel presented by the Migration Wizard is the prompt for the module name. The module name prefix will be prepended to the name of the source artifact being migrated.

For Example, a module name of **Simple** and a WICS connector called `ClarifyConnector`, will result in the `Simple_ClarifyConnector` WPS module.

The second panel will provide the opportunity to specify the error handling options.

The `reposMigrate` options used by the Migration Wizard are shown on the right of the slide.

## Conversions

- WICS BOs to WPS BOs and BGs
- WICS Maps to WPS Data Maps
- WICS Relationships to WPS Relationships
- WICS Collaboration Templates to WPS BPEL and WSDL files
- WICS Database Connections to WAS DataSources (reposMigrate only)
- WICS Schedule Entries to WPS Application Scheduler Entries (reposMigrate only)
- WICS Collaboration Objects and ICS Connector Definitions to WPS Modules containing SCA components, interface maps, and wiring
- One Project/EAR generated for each ICS Collaboration Object and Connector Definition

13

WebSphere InterChange Server Migration

© 2005 IBM Corporation

Shown here is a mapping from the WebSphere InterChange Server artifacts to their corresponding WPS V6 artifacts.

Many of the WICS artifacts, such as, BOs, Maps, and Relationships, map directly to SCA artifacts.

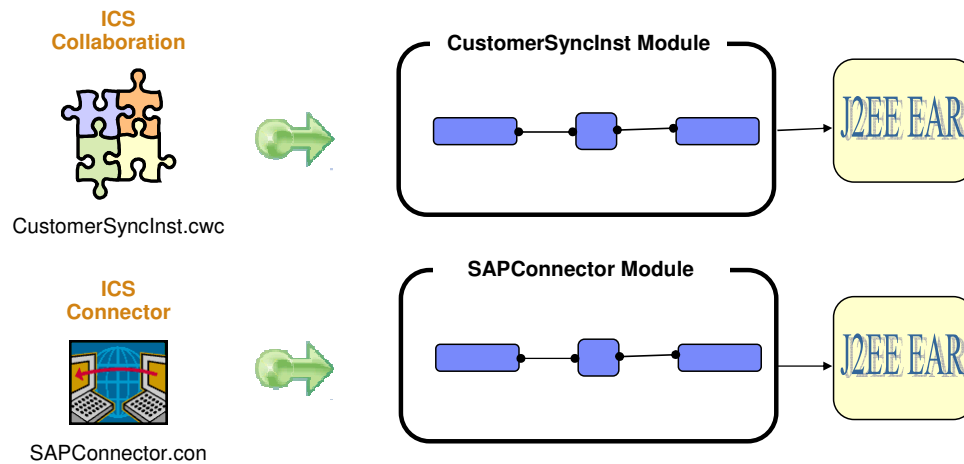
With WICS, business process flows are defined using Collaboration Templates and in WPS V6 flows are defined using BPEL.

Notice that WAS DataSources and the Scheduler entries are only available using the reposMigrate command line utility. These must be manually configured when testing in WID.

If there are Scheduler entries in the application being migrated then the WPS Administrative console can be used to set and configure the Application Scheduler from WID.

## WICS Artifacts

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



It is important to note that the WICS artifacts are mapped to SCA Modules, which are deployed as J2EE EARs, so there will be an EAR for each of the WICS artifacts. This has the potential for generating a lot of EAR files.

## WICS Migration - reposMigrate

- There are two approaches that can be used.
  - ▶ Create J2EE™ EARs to be deployed manually
    - Uses *ServiceDeploy* to create the EARs
  - ▶ Create J2EE EARs and have them automatically deployed using **WSADMIN**.
    - Uses *ServiceDeploy* to create the EARs
    - then *wsadmin* to deploy the EARs to the runtime.
- The apps can run in both the WICS and the WPS servers without interfering with each other, but the transition with respect to adapters, relationship database, EIS systems and other factors external to WPS and ICS need to be worked out.



**reposMigrate** is the WPS command line utility for converting the WICS artifacts to WPS/SCA artifacts. **reposMigrate** can also invoke the *ServiceDeploy* utility to package the WPS/SCA artifacts into jars and EARs.

**ServiceDeploy** is the command utility used with WPS V6 to compile and package the SCA and J2EE artifacts programmatically. The **reposMigrate** utility receives the WICS artifact jar, 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 Administrative console, 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.

**reposMigrate** and *ServiceDeploy* can be used with ANT and a source code control system to automate the process even further.

Before the EARs are deployed to the runtime, whether you do it automatically or manually, the details for cutting over with respect to the adapters, relationship database and other external dependencies must be worked out.

## reposMigrate Command

Syntax of Repos Migrate command-line utility:

- Required params
  - ▶ -i<SourceArtifactJar> The input WICS jar file to migrate
  - ▶ -o<OutputDirectory> The output directory for the jars/ears created by migration
- Server params (EAR deploy and creation of DataSources/Schedule Entries)
  - ▶ -s<TargetName> Attempt to deploy created EAR files to target server/cluster
    - e.g. -scell=MyCell,node=MyNode,server=server1 or -scell=MyCell,cluster=MyCluster
  - ▶ -sh<hostname> Hostname for SOAP access to server
  - ▶ -sp<port number> Port for SOAP access to server
  - ▶ -sf<PropertiesFile> Properties file to use for client SOAP access (Needed for security information)
  - ▶ -r Prevent relationship coexistence (Does not point relationship at ICS datasource)
- Warning/Failure params
  - ▶ -fh Halt at first Failure
  - ▶ -ai Warn on duplicate datasources
  - ▶ -wi Ignore java conversion warnings (see best practices)
- Logging params
  - ▶ -lf<LogFileName> Logs migration messages to a file
  - ▶ -lv Log verbose (Default logging level is WARNING, verbose raises level to INFO)
- Misc params
  - ▶ -ao Create artifact jar files only, does not attempt to call serviceDeploy (cannot be used with -s)



Shown here are the many options that can be used with reposMigrate.

Take a moment to study them.

It is Recommended that you use the -ao flag when using reposMigrate to generate artifact jar files only.

This also eliminates the need for the -s, -sh, -sp, -sf, -r, and -ai parameters.

Once the artifacts have been edited, serviceDeploy can be called to create EAR files from the JARs.



## Recommendations for Migrating WICS Artifacts

- Turn off “Build Automatically” in WID before doing a migration. Not doing so will greatly slow the migration process.
- Use as small an input jar as possible that is still a self contained solution. Between the large number of artifacts created during migration and WID performance, the migration operation and manipulation of artifacts in WID can take an extensive amount of time.



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 WID will speed up the migration process.

Using the smallest possible input jar to contain a solution will also streamline the migration and reduce the time required to complete.

## Troubleshooting – Known limitations

- Relationship Coexistence does not work in this release.
- CWT Message Recipients are not migrated.
- CWT Iterator nodes are migrated to empty conditional While nodes, you must setup all variables and the condition for the while loop.
- CWT Break nodes are migrated as empty nodes that would not break a loop.
- All exceptions are currently handled with Catch Alls, meaning they are not distinguished individually from each other.
- Correlation Sets are not migrated.
- Benchmarks are not migrated.
- The "Pause" ability of Schedule Entries is not migrated.
- Access EJB is not part of this release.
- Security information is not migrated in this release.
- Failed Event and Email APIs are not supported this release.
- Sync Adapter Input and Async Adapter Response Input are not supported this release, LARD only supports Async Input and Async/Sync Output.
- This release, the XML snippet to Java snippet conversion uses a hard coded template, not allowing any user templates to be used.



Shown here are some known limitations with respect to migration.

Take a few minutes and review them.

CWT as used in this slide stands for CrossWorlds Template.

## Troubleshooting – Best Practices

- Make Windows TEMP dir path short
- Always assign an initial value to all declared variables in a map/cwt
- Don't use spaces or other non-NCName characters in BO attribute names, suggestion: use "\_" instead
- Don't depend on preserved order of entries in a BusObjArray
- Don't index a BusObj in a BusObjArray (above, if you can't depend on order, you wouldn't know which you were getting)
- Only use the WICS tooling to manipulate your ICS artifacts
- Use only the documented ICS APIs
- For all BO attributes, specify as exact a type and length as possible. i.e. (Do not use Strings to represent integers, do not use a 255 length attribute to represent a zip code)
- Avoid using static/final/transient/native in CWT java snippets
- Use Activity Editor whenever possible to write java snippets
- Use implicit DB transaction bracketing
- Use only values in map set operations, not Java code
- Create Maps in pairs so that if there is a map for ASBO->GBO, there will be a corresponding GBO->ASBO map
- In java code, don't spawn threads, use java.io.\* (store data in db instead), and in general write the code following the limitations of the EJB 2.1 spec
- Never alter the relationship database information manually or by using SQL, always allow on relationship designer to make these changes
  
- Possibly just this release:
  - Have all collaboration objects and templates use only one triggering port
  - Explicitly specify all maps in connector files i.e (Do not use implied connector maps)

Shown here are some troubleshooting best practices.

Take a few minutes to study them.

On the Windows platform, the length of the path/filename is limited to 256 characters.

When the Workspace is nested in a directory structure and the object have long names and namespaces, its very easy to exceed the Windows 256 character limit.

## Troubleshooting – Debugging Migration

- First level of debugging – the log file
  - ▶ The log file is enabled by the `-lf` param, and can contain a higher level of tracing if the `-lv` param is used as well
  - ▶ Any error messages match up to either a current know issue or best practice that was not followed
  - ▶ If the failure does not produce an error message, or produces the “Unknown Exception” error message, further debugging is needed
- Second level of debugging – reproduction
  - ▶ With the input jar file, the migration can easily be recreated
  - ▶ The options used during migration are needed to accurately recreate the scenario
    - WID - What wizard options were selected
    - reposMigrate – What command line options were used
      - If server parameters were used, then we will need to know about the user’s environment. (i.e. If they chose to deploy to server1, does server1 exist, what is its hostname, what is its soap port number, does it have security enabled, if so does the soap.client.properties file match its security settings)
  - ▶ If the error is not during migration, but in the validation of the artifacts created from a migration, then
    - The error may correspond to a known issue or best practice that was not followed. If so, the user should follow basic WPS procedure for solving these issues as if the artifacts were created brand new by themselves, as opposed to created by migration
    - If the error does not have a known cause, then it must be determined why that artifact is invalid and what should be done to make that artifact pass validation. Then the migration code can be debugged to discover why it did not generate the artifact in that way

Shown here are some tips for troubleshooting problems encountered during migration, including the location of log files and the type of information contained in them.

There are also some tips for advanced troubleshooting if the error is not logged or is generic in nature.



## Summary

- Migrating from WICS to WPS is not a perfect fit.
  - ▶ There will be post migration work involved.
- The overall process is one of
  - ▶ Export from WICS
    - repos\_copy
  - ▶ Import to WebSphere Integration Developer
    - Migration Wizard
    - Import
    - reposeMigrate + import
  - ▶ Tune the new model for BPEL / SCA
  - ▶ Deploy to the WPS runtime
- The process can be automated using ANT, reposMigrate, ServiceDeploy and WSADMIN
- What can be mapped is well defined.
- There are some WICS features that are not migrated to BPEL / SCA and this is captured in the limitations section.



To summarize.

Depending on the complexity of the WICS 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 WICS repository jar into WID, there are several options available, the Migration Wizard, Import **or** reposMigrate + Import.

Using reposeMigrate + 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	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. 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.