



WebSphere MQ Workflow to WebSphere Process Server Transition, Example of a custom process instance migration

Version 3, June 2009

© Copyright International Business Machines Corporation 2009.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to: *IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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 and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the

names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

- Redbooks
- WebSphere

The following terms are trademarks of other companies:

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

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Abbreviations

API	Application programming interface
EAR	Enterprise archive
EJB	Enterprise JavaBeans
FDL	Flow definition language, the file exchange format of WebSphere MQ Workflow definitions
JAR	Java Archive

WMQWF WebSphere MQ Workflow
WPS WebSphere Process Server
XML Extensible Markup Language
XSL Extensible Stylesheet Language
XSLT XSL Transformation

Contents

Notices	iii
Trademarks	iv
Abbreviations	iv
Chapter 1. Introduction	1
Version history	1
Version 2	1
Version 3	1
Package content	1
Prerequisites	2
Directory structure of the package	2
Chapter 2. Process Instance Migration Approach	5
Check feasibility and complexity of instance migration	6
Decide how to test process instance migration	8
Migrate the process model to WebSphere Process Server.	9
Prepare WebSphere MQ Workflow processes for migration	10
Suspend WebSphere MQ Workflow processes for migration	11
Extracting the data of running WebSphere MQ Workflow process instances	14
Extract the WebSphere MQ Workflow process instance data	14
Instantiate WebSphere Process Server process instances.	15
Chapter 3. Migrating process instances from other systems than WebSphere MQ Workflow to WebSphere Process Server	17
Chapter 4. Running the example.	19
Steps performed on the WebSphere MQ Workflow side.	22
Steps performed on the WebSphere Process Server side.	23
Bibliography	27

Chapter 1. Introduction

Migrating process instances is not a standard feature of either, WebSphere MQ Workflow or WebSphere Process Server. Instead, the APIs of the products provide capabilities that can be used for process instance migration. As a result, customers who migrated process instances from WebSphere MQ Workflow to WebSphere Process Server wrote specific tools for this task.

For this example, another approach was chosen: Based on the available APIs of both products, general tools for instance migration were developed. To show the functions of the tools, a sample process is provided.

This example is by no means a complete solution to migrate process instances of all possible process models from WebSphere MQ Workflow to WebSphere Process Server. Instead, an instance migration is possible with more or less effort for some process models. For other process models, an instance migration might not even be feasible with reasonable effort.

If you want to migrate process instances from an other system than WebSphere MQ Workflow to WebSphere Process Server, the approach shown here might also help you to create process instances on WebSphere Process Server

Version history

Both, the tool to extract the process instance data from WebSphere MQ Workflow and the process instantiate tool print out their version when started.

Version 2

Initial version.

Version 3

The following changes were made for this version:

1. The process instance extract tool for WebSphere MQ Workflow has the following new options:
 - instancesPerFile*
 - instancesPerQuery*
 - predefined*
2. Prior to version 3, the process instance extract tool for WebSphere MQ Workflow has problems extracting data structures containing names with blanks. This had been fixed.

Package content

This document describes a package containing the following parts:

1. Tools for process instance migration. The source code of these tools is included.
2. For a sample WebSphere MQ Workflow process model, it is shown how process instances can be converted to WebSphere Process Server.

3. A sample process instance data file which you can use if your are interested only in the creation of WebSphere Process Server instances.

At the Internet address: <http://www.ibm.com/support/docview.wss?rs=2307&uid=swg27015700> you can find instructions how to obtain this package.

Prerequisites

1. Minimum product level of WebSphere MQ Workflow is version 3.6.
2. Minimum product level of WebSphere Process Server is version 6.2.0.1. The reason for this requirement is that two WebSphere Process Server APIs used by the instantiate tool of this example are only available from this version on.
3. You should be familiar with process model transition from WebSphere MQ Workflow to WebSphere Process Server. For more information see “WebSphere MQ Workflow Transition to WebSphere Process Server” on page 27
4. If you plan to adapt the source code of the sample tools, you should be familiar with Java™, enterprise Java beans (EJBs), and using the WebSphere MQ Workflow Java API and WebSphere Process Server EJB API.
5. XML knowledge, especially knowledge on how to transform XML data might be helpful in case of migration tasks that involve name changes of artifacts.

Directory structure of the package

This is the version 3 of the package.

After extracting the files from the package into an empty directory, the following directory structure is created:

binlinux

Contains the command files to perform the commands on Linux®.

binwin

Contains the command files to perform the commands on Windows®.

lib

Contains the migration tools and the sample process models of both, WebSphere MQ Workflow and WebSphere Process Server ready to be deployed and used.

InstanceMigrationDocumentation.pdf

This document.

src\procInst001.xml

Sample output of the WebSphere MQ Workflow process instance data extract tool as reference.

src\SeqInst.zip

Process interchange file of the WebSphere Process Server sample process. Import this file into WebSphere Integration Developer to view the process model.

src\wmqwf

Contains the source code of the migration tools used on the WebSphere MQ Workflow side.

src\wps

Contains the source code of the migration tools used on the WebSphere Process Server side.

Chapter 2. Process Instance Migration Approach

Specially if a process instance is running for a longer time, there is sometimes the need to migrate not only the process model but also running instances. This means that relevant process data and process state must be extracted from WebSphere MQ Workflow and transferred to WebSphere Process Server. The following figure gives an overview of the various steps involved.

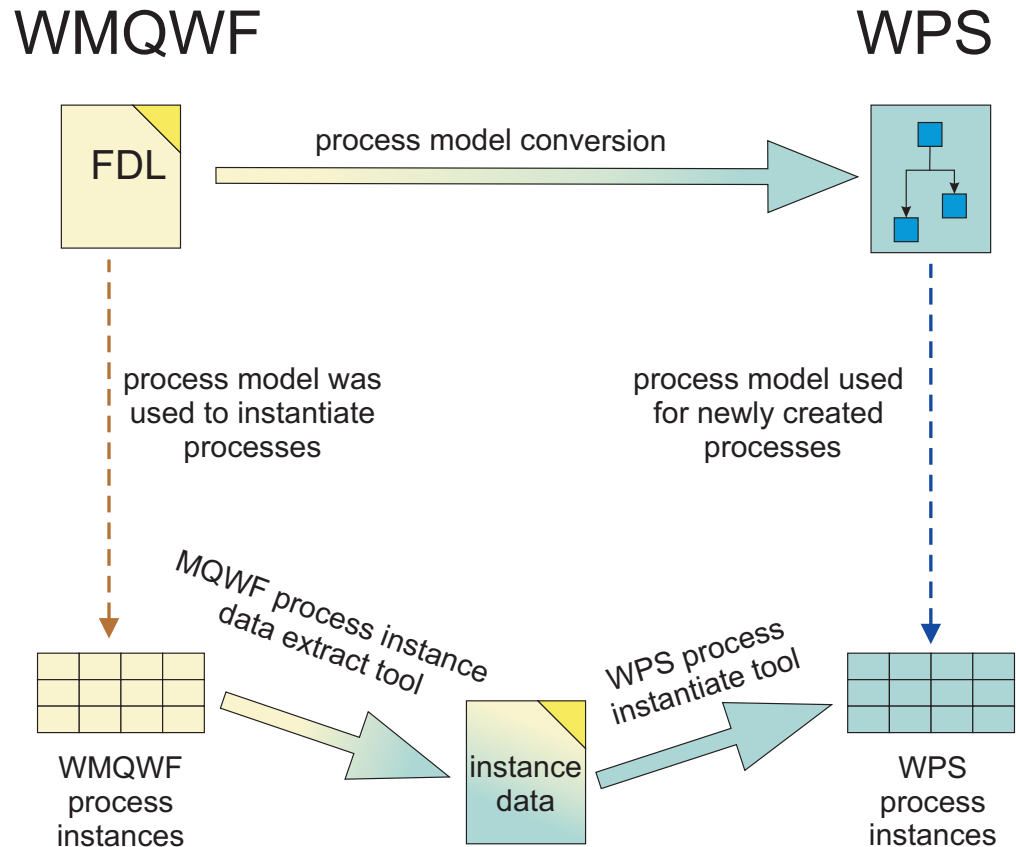


Figure 1. Migration overview

The following steps must be performed for a process instance migration:

- 1. Check feasibility and complexity of instance migration**

Depending on your process model and the features used, an instance migration might be more or less complex. In some cases, an instance migration with reasonable effort might even not be feasible. During this step, areas for potential problems are checked so that you can get a first impression of the effort necessary for a successful instance migration.

- 2. Decide how to test process instance migration**

Before migrating process instances in production, test your migration approach on a test system.

- 3. Migrate the process model to WebSphere Process Server**

At first, a suitable WebSphere Process Server process model must be created.

- 4. Prepare the WebSphere MQ Workflow processes for migration**

Of the process instances to be migrated, you should assure that no activities are in work.

5. **Extract the process instance data**

This step reads process instances of a given process template name and writes the information into files in XML format.

6. **Instantiate process instances in WebSphere Process Server**

The XML files are read, and for each process instance found in the file, a new process instance is created in WebSphere Process Server. As closely as possible, the state of the newly created process reflects the state of the original WebSphere MQ Workflow process instance.

In the next chapters, each of these steps is explained in detail.

Check feasibility and complexity of instance migration

It is crucial for a correct migration that all relevant data and state information is set correctly in WebSphere Process Server. To ensure this, you should check the following:

Decide which WebSphere MQ Workflow data containers need to be migrated to WebSphere Process Server

The process instantiate tool migrates only the process input container, the input container of the active activity and optionally its output container, and the global container to WebSphere Process Server. You should check the process model if there are other containers relevant for process migration. In this case, adapt the process instantiate tool to set this data.

If the WebSphere MQ Workflow process contains a global container, the global container content is available in the XML file. When initiating a WebSphere Process Server process, it is assumed that a variable with name `GLOBAL_CONTAINER` is set. If the global container fields should be queryable, you should use query properties for it.

Decide if WebSphere MQ Workflow predefined data members need to be migrated to WebSphere Process Server

When importing a WebSphere MQ Workflow process definition to WebSphere Process Server, you can either create predefined data members or not. Usually you create predefined data members in WebSphere Process Server if in the WebSphere MQ Workflow process model, predefined data members are used. Depending on this, the variables have a different format.

So, regarding instance migration, if you need the predefined data members, then you have to use the option **-predefined** of the WebSphere MQ Workflow data extraction tool. This also extracts the predefined data member values and stores them in a suitable format.

Check if you need to migrate process instances with parallel activities

Both WebSphere MQ Workflow and WebSphere Process Server have the capability to run activities of one process instance concurrently. Areas of a process model where multiple activities can run concurrently are called *parallel sections*. During transition, such parallel sections need special attention because of the following reason:

After creating a WebSphere Process Server process instance, the instantiate tool uses the WebSphere Process Server `jump()` API to navigate to the activity which will be the next to be worked on. As the `jump()` API can

have only one activity as target, this approach is only feasible if the current execution state is not inside a parallel section.

To avoid this problem, the sample process model does not contain parallel sections.

If you want to migrate process instances with parallel sections, a solution of this problem would be to jump to the activity before the parallel section, then set the output container of this activity and skip the activity. If needed, other activities must be skipped as well in order to navigate to the appropriate activities. You can do this by adapting the code of the migration tools to your needs.

Check if you need to migrate the WebSphere MQ Workflow process context

The WebSphere MQ Workflow process context information can be mapped to query properties in WebSphere Process Server. The process context information is available in the XML file. Therefore, if you want to migrate this information, you only need to adapt the instantiate tool.

Check if you need to migrate activity starter information

The instantiate tool starts the processes under the user ID performing the instance migration. If you want to keep the process starter information, the security context must be set to the correct user ID before starting the process in WebSphere Process Server.

Check if you need to migrate to process model with different valid-from dates

In both, WebSphere MQ Workflow and WebSphere Process Server, you can have multiple versions of a process model with different valid-from dates. This implies, that at migration time, process instances of different template versions with the same process template name can exist.

When extracting the process instance data, the valid-from dates are also extracted. However, the process instantiate tool on the WebSphere Process Server side does not use this information. Instead, it is assumed that all process instances are created using the currently valid process template. This code can be adapted in a way that different process templates (for example templates which were valid at an earlier time) are used to instantiate the processes depending the WebSphere MQ Workflow valid-from date or the process starting time.

Notification and expiration data not migrated

With notification (or escalation in WebSphere Process Server) and expiration, deadlines can be set in a business process.

Any timers set in a WebSphere MQ Workflow process instance are not migrated to WebSphere Process Server. For example if an activity must be completed within five days and in WebSphere MQ Workflow, three days have already passed, then, after migrating the process instance to WebSphere Process Server, the escalation timer would again start from zero.

A solution to this problem can be to assign the timer values dynamically to the WebSphere Process Server process instance.

Only the status of completed activities is migrated

For all processes migrated from WebSphere MQ Workflow to WebSphere Process Server, activities are either performed in WebSphere MQ Workflow or WebSphere Process Server. So before migrating processes to WebSphere

Process Server, you should complete all outstanding work in running activities. If a process instance with a running program or subprocess activity is migrated, this activity is considered as not yet started.

Not initialized array members are not supported in all situations by WebSphere Process Server

In WebSphere Process Server, arrays defined in a variable must fulfill the following condition: If an array member is non-empty, then all previous array members must also be non-empty. In order to avoid errors later, the WebSphere MQ Workflow process data extract tool initializes array members if necessary. Suitable warning messages are emitted, as you will see when performing the example.

Decide how to test process instance migration

Before migrating process instances in production, you should test the migration on a separate WebSphere Process Server system.

You should have a test system of WebSphere Process Server, where you can test to migrate your process instances. Depending on migration approach (see “Prepare WebSphere MQ Workflow processes for migration” on page 10), on the WebSphere MQ Workflow side, a test system containing sample process instances can also be necessary:

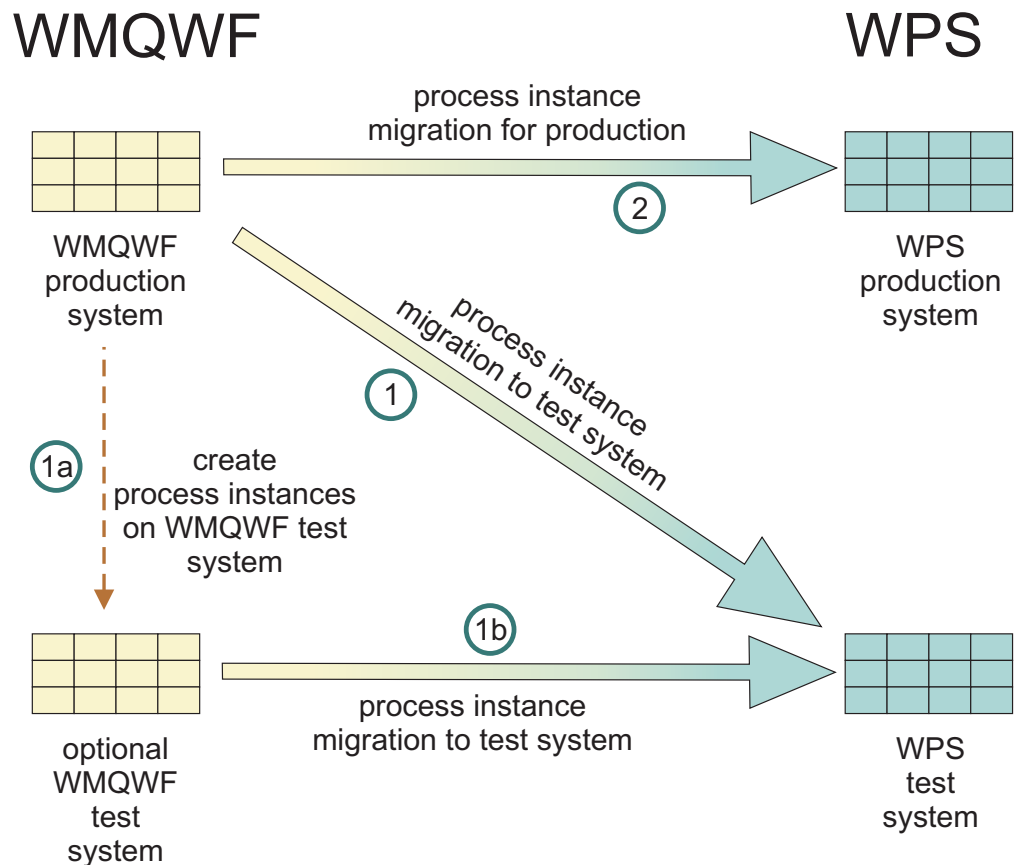


Figure 2. Test approach for process instance migration

- If you migrate running process instances, with read access to the WebSphere MQ Workflow production system you can extract the process instance data and instantiate the process instances on the WebSphere Process Server test system as described as step 1 in the figure above.
- If you want to migrate suspended process instances, you might not be allowed to suspend the process instances of the production system for testing purposes. Therefore, a WebSphere MQ Workflow test system containing sample process instances of your application might be necessary. This is described as step 1a in the figure above. Then, with step 1b the process instances can be migrated to your WebSphere Process Server test system.
- After successfully testing your migration, migrate your production system as described with step 2 in the figure above.

Migrate the process model to WebSphere Process Server

When creating the WebSphere Process Server process model for migration you should notice the following points:

Use Generalized Flows

After creating a process instance in WebSphere Process Server representing a WebSphere MQ Workflow process instance, navigation in this process instance should continue at the current activity of the WebSphere MQ Workflow process. This can be done in two ways:

1. Use the WebSphere Process Server `jump()` API to jump to this activity.
2. Model the process in a way that depending on a variable value, navigation continues at the appropriate activity.

Here, the first alternative is used.

The `jump()` API does not allow to jump into a nested activity. Therefore, in your process, all activities should be directly contained in the same generalized flow activity. Use a backward link if activities must be performed multiple times. This is the approach shown in the process models included in this example.

When modeling diverging gateways in the generalized flow you should use *split* gateways if possible. If you use a *fork* or *inclusive OR* diverging gateway instead, then this gateway is the starting point of a parallel section. As described in “Check if you need to migrate process instances with parallel activities” on page 6, all following activities cannot be used as jump targets.

Alternatively, if you only have a sequential execution of activities, use a sequential structured activity.

If possible, keep names unchanged when migrating a process model from WebSphere MQ Workflow to WebSphere Process Server.

If, for example, an activity name in a WebSphere Process Server process is different to its corresponding WebSphere MQ Workflow activity name, a suitable mapping must be done during instance migration. This can be done either by adapting the Java code or with an XSLT stylesheet which converts the process instance data files. In the sample process here, the names have been set so that they can be used in both WebSphere MQ Workflow and WebSphere Process Server without change.

Keep the variable structures compatible to the WebSphere MQ Workflow container structures

During migration, the process input container and the input container of the current activity are read and are used when creating the process instance in WebSphere Process Server. For the process instance migration tool, it doesn't matter which variables are used as input variable of the receive activity and input variable of the current activity. You only must ensure that the WebSphere MQ Workflow container content can be written into these variables.

So, when migrating the process model to WebSphere Process Server, you can always extend a data structure. However, removing data elements from a data structure leads to an error if a process instance is migrated where this element is set.

Depending on your decision, to migrate predefined data members in addition or not, the data is extracted from WebSphere MQ Workflow in a format compatible to the FDL import wizard of WebSphere Integration Developer

Taking this into account, the question arises how to create the WebSphere Process Server process model. The following approach is based on our experience.

Create the process graph manually in WebSphere Integration Developer

There are several reasons for this:

1. The WebSphere Integration Developer FDL import wizard generates the activity graph in a parallel activity. Converting this activity to a Generalized Flow is not an easy task.
2. The number of activities and variables generated by the WebSphere Integration Developer FDL import wizard is usually larger compared to a manually created WebSphere Process Server process model. In larger models it becomes difficult to manually optimize the process due to the increasing complexity of the process model.

Use the WebSphere Integration Developer FDL import wizard in special situations

For example your process model contains large numbers of data structures or there are many usages of UPES (user defined program execution server). With the FDL import wizard you can create the necessary definitions and then copy them into your manually created process model.

Prepare WebSphere MQ Workflow processes for migration

For this step, two approaches are possible:

Migrate running process instances

You can extract the process instance data from running activities. Ensure that after extracting the process instance data, no work can be performed anymore with the WebSphere MQ Workflow process instances. This approach is described in the chapter "Extracting the data of running WebSphere MQ Workflow process instances" on page 14.

Migrate suspended process instances

You can suspend the process instances before extracting the data from WebSphere MQ Workflow. In this case, consider the following

1. Process instances in state SUSPENDED cannot be changed anymore. Therefore, the instances are in a defined state and you only must assure that the instances are not resumed.

2. As this approach changes the process states in your WebSphere MQ Workflow system, testing this approach might not be feasible on your production system. Instead, you might setup a WebSphere MQ Workflow test system with sample process instances.
3. Specially if you want to migrate a larger amount of process instances in a limited time, suspending the process instances before extracting the data might take too much time.

This approach is described first.

Suspend WebSphere MQ Workflow processes for migration

The example contains a process suspend utility suitable for this task. A process is in state `SUSPENDED` only if activities which are currently in work are either completed or canceled.

At first, a list of the possible activity states is given together with any actions necessary before the process instance data can be extracted.

A process remains in state `SUSPENDING` as long as activities in the following two states exist:

CHECKED_OUT

To complete ongoing work: check-in, cancel-checkout, force-finish or force-restart the activity.

RUNNING

To complete ongoing work: execute, force-finish or force-restart the activity

In addition, there are several options how to handle the following activity states:

EXECUTED

This state is characteristic for a manual exit in WebSphere MQ Workflow. In WebSphere Process Server, this can be modeled by an additional staff activity after the activity in question. The migration of a manual exit is not supported by the FDL import wizard of WebSphere Integration Developer. You have the following two options:

1. Complete the activity on the WebSphere MQ Workflow side, that means you must finish, force-finish, restart or force-restart the activity. This is the approach chosen by the sample code provided.
2. As an alternative, an activity in this state can be mapped to a WebSphere Process Server process instance in such a way that the WebSphere Process Server process instance continues its work at the human task representing the manual exit.

IN_ERROR, TERMINATING, TERMINATED

In WebSphere MQ Workflow, these states are not final states, navigation has stopped. The equivalent activity state in WebSphere Process Server is `STOPPED`. You have the following two options:

1. Complete the activity on the WebSphere MQ Workflow side. This means you must force-finish or force-restart the activity. This is the approach chosen by the sample code provided.

2. As an alternative, the WebSphere Process Server process can be modeled in a way that after creating the WebSphere Process Server process, the activity can be brought into the state STOPPED. This can be done as follows:
 - a. In WebSphere Process Server, the process attribute *Continue on error* must be set to false. Verify that unwanted side effects do not occur.
 - b. An additional boolean variable is defined, for example *setToStopped*.
 - c. Before this activity, a Java snippet activity is defined which sets the variable *setToStopped* to *false*.
 - d. If the activity should go into the STOPPED state after resuming the process, the variable *setToStopped* is set to *true*.
 - e. Then, for this activity, create an exit condition which must be evaluated on entry. As expression language select *Java*.
 - f. The Java exit condition contains the following code:


```
// @bpe.readWriteVariables names="setToStopped"
if(setToStopped) {
    setToStopped=false;
    raiseFault()
} else {
    return true
}
```
 - g. If during instance migration, a WebSphere MQ Workflow activity is found in one of the states: IN_ERROR, TERMINATING, TERMINATED, then in the WebSphere Process Server process the variable *setToStopped* is set to true and the activity is started. Because of the exit condition, it immediately goes into the state STOPPED.
 - h. If this activity should be processed as usual, then either the variable *setToStopped* is set to *false* before jumping to this activity or jump to the Java snippet activity which sets the variable correctly.
 - i. Then, either by means of the BPC Explorer or the BPC API, you can handle the activity, for example restart it or complete it.

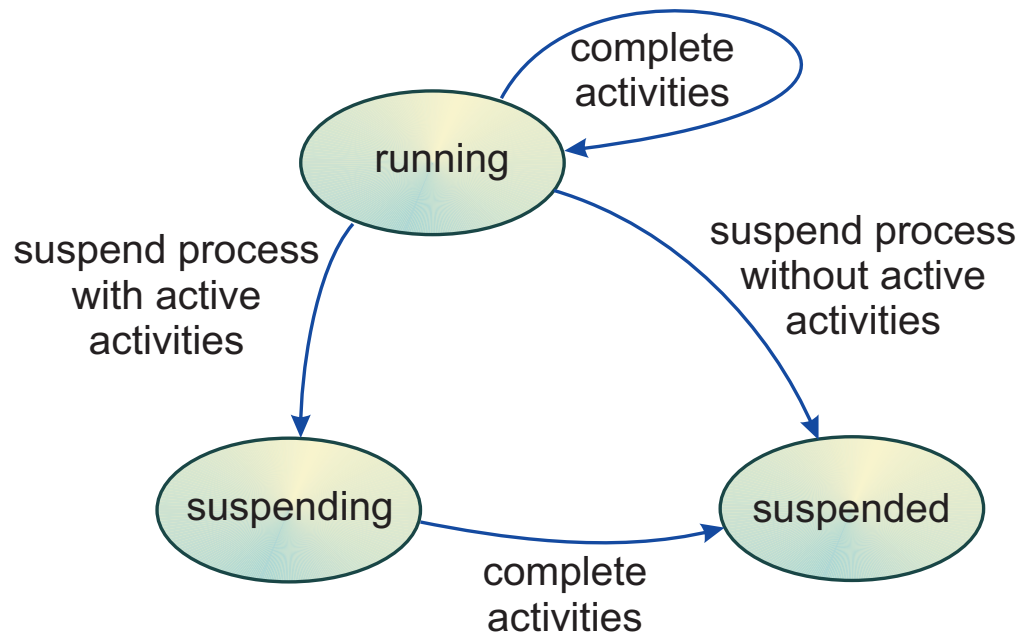


Figure 3. WebSphere MQ Workflow process states when suspending a process

If a process instance has activities in one of the states listed above, there are two approaches possible when suspending the process:

Complete the activities before suspension

Here, the suspend utility checks if the process instance contains activities in one of the states described above. If this is the case, the process instance is not suspended and a suitable message is emitted to standard output.

Complete the activities after suspension

If the suspend utility is called with option *-force*, then a process instances is suspended even if it contains an activity in one of the states described above. In this case, a message is emitted to standard output as reminder that work has to be completed.

You can use this option if you cannot complete activities before suspending the process. This is typically the case, if after the activity to be completed, a long-running automatic activity should not be started. If you use this option, observe the following points:

1. The last activity that was performed in the process instance is determined by comparing the *Last Updated* time stamp of the activities. If multiple WebSphere MQ Workflow systems run on different machines, check that the system clocks are synchronized.
2. It is assumed that the activity with the latest *Last Updated* time stamp was the only active activity. That means, the tool cannot not check that multiple activities are being worked on which is not supported during instance migration.

Extracting the data of running WebSphere MQ Workflow process instances

If you want to avoid the step of suspending the process instances, you can alternatively extract the instance data information from running process instances. In this case, assure that no further processing of the WebSphere MQ Workflow instances occurs because this can lead to inconsistencies.

When using this approach, all activities which are being processed are considered as not yet started. In addition, only the input container of the active activity is used for migration, not the activity output variable. This approach has the same effect as if you had restarted or force-restarted the active activities.

Extract the WebSphere MQ Workflow process instance data

The process instance data extract utility must be started with the following mandatory options:

WebSphere MQ Workflow configuration identifier

This identifies the WebSphere MQ Workflow configuration you are working with.

User ID

The WebSphere MQ Workflow user ID you want to use. This user ID is needed to log on to the WebSphere MQ Workflow API.

Password

The WebSphere MQ Workflow password you want to use

Process template name

Only the data of process instances with this template name is extracted.

Process state

Here, specify **-suspended** or **-running**, depending on your decision to extract the instance data from either running or suspended process instances.

XML base file name

All data retrieved from WebSphere MQ Workflow is written into XML files. In order to limit the file size, no more than 1000 process instances are written into one file. Then the next file is used. The file names are constructed as follows: The base file name is appended with a three-digit number followed by the prefix **.xml**

In addition, after the mandatory options, the following options can be added:

-instancesPerFile <number>

This number specifies how many process instances are written at most into one data file. As before writing the XML data to the file, the whole data is held in memory, use this parameter to avoid out-of-memory situations. The default value is 1000.

-instancesPerQuery <number>

This number specifies how many process instances are processed with one query. As the maximum message size exchanged between the WebSphere MQ Workflow runtime server and the WebSphere MQ Workflow client is limited, only a limited number of process instances can be returned with one query. If the number of process instances to extract is larger than this

number, then multiple queries are performed by the tool. Therefore, changing this value does not change the number of extracted process instances. The default value is 700.

-predefined

Using this option has two consequences: The predefined data member values are also written to the XML files. In addition, the data format is compatible to the data format used by the FDL import wizard of WebSphere Integration Developer when selecting the *Create predefined data members* option.

Instantiate WebSphere Process Server process instances

Creating the WebSphere Process Server process instance is done with the following steps:

1. Initiate and Suspend process instance

A new process instance is created from the process template name specified in the XML file. The process template currently valid is used to instantiate the new process.

2. Jump to the current activity

The activity where the wave front is located is used as jump target. In addition, the input variable of this activity is set using the data provided from WebSphere MQ Workflow.

3. Optionally skip the actual activity

In case this activity has already being processed on the WebSphere MQ Workflow side, the output variable of this activity is set using the data provided from WebSphere MQ Workflow. Then, this activity is skipped.

4. Resume the process

As last step, the newly created process instance is resumed. The following two cases can occur:

- a. If the actual activity was skipped, navigation continues after this activity. That means that link conditions of all outgoing links are evaluated and the link targets change their states accordingly.
- b. If the actual activity was not skipped, this activity is either in ready state (in case of a manual activity) or is started immediately. Note that a restarted automatically started activity in WebSphere MQ Workflow goes into the READY state. In such a case they must be manually started in WebSphere MQ Workflow. When migrating such a process to WebSphere Process Server, such activities are always automatically started.

The tool requires the following options:

-create This option selects the mode to create process instances in WebSphere Process Server.

file name

This can be either a specific file name or the file name prefix specified when extracting process instance data from WebSphere MQ Workflow. In both cases, potentially multiple files are read and processed.

Chapter 3. Migrating process instances from other systems than WebSphere MQ Workflow to WebSphere Process Server

If you want to migrate process instances to WebSphere Process Server, this example can be a starting point for your approach. You can do the following:

1. Make yourself familiar with how the process instances are instantiated in WebSphere Process Server.
2. Create an instance data file in a format similar to the sample file in the `src` directory of this example.
3. Use and, if necessary, adapt the WebSphere Process Server instantiate tool of this example to your needs.

Chapter 4. Running the example

The capabilities of the process instance migration tools can be demonstrated with the included sample process. The process definition can be found at `lib\SeqInstance.fdl` in the package.

See Figure 4 for a graphical representation of the process model.

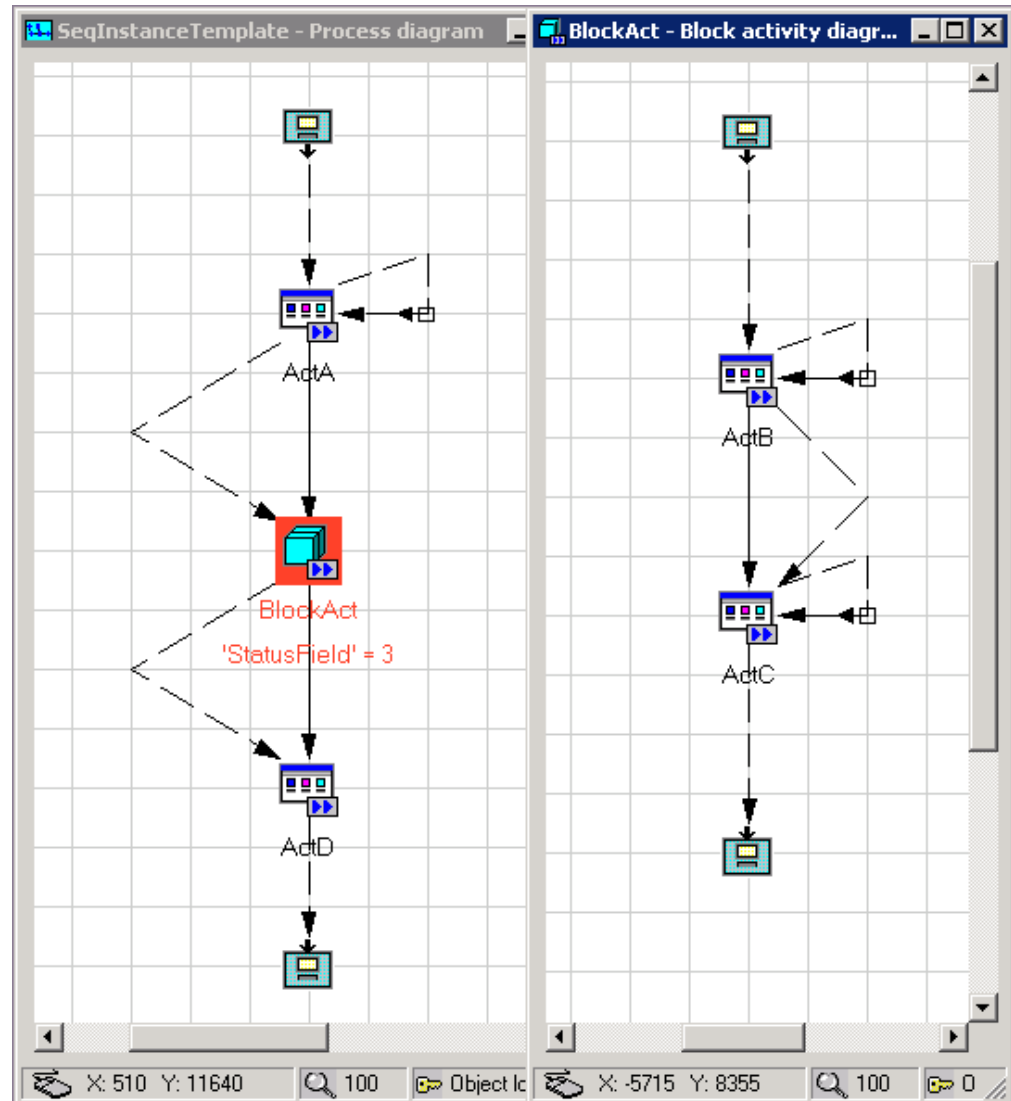


Figure 4. WebSphere MQ Workflow process model of this example

In WebSphere MQ Workflow, this process can be characterized as follows:

- On the topmost level, this process contains three sequentially performed activities: a manual activity, then a block, and then again a manual activity. After starting the block, the activities contained in the block are performed until the exit condition is true.
- Inside the block is an empty activity which simulates an automatic UPES activity followed by a manual activity.

- All activity input and output container use the same data type containing array elements and an other data type.
- A subset of the input container of the first activity is set from the process input container.
- Data flow is performed in parallel to the control flow by data default connectors and data connectors. Always, identity mapping is used.
- A subset of the output container from the last activity is mapped to the process output container.

The corresponding WebSphere Process Server process is available at **src\SeqInst.zip** as process interchange file. It is graphically represented in Figure 5 on page 21.

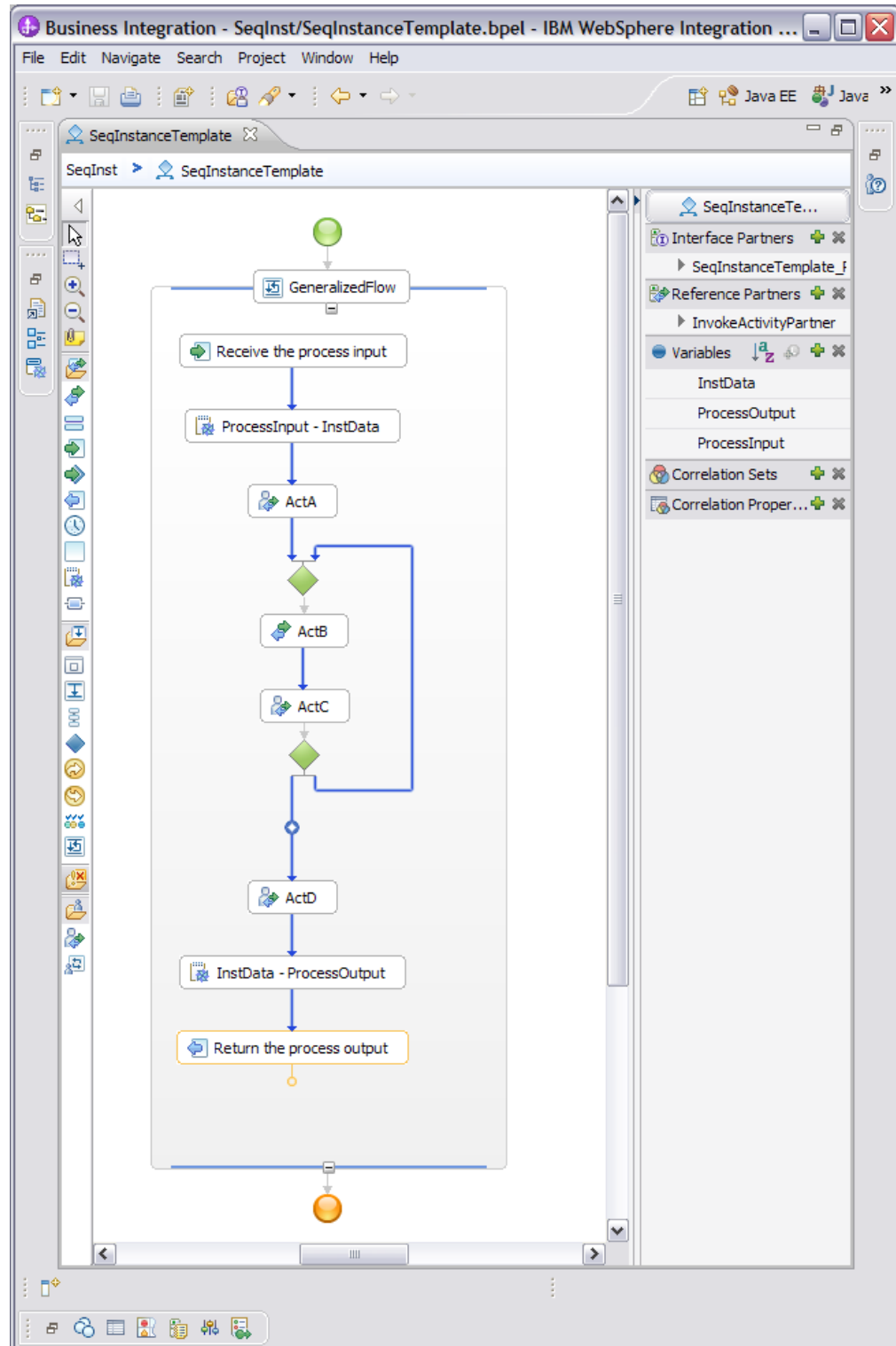


Figure 5. WebSphere Process Server process model of this example

The corresponding WebSphere Process Server process can be characterized as follows:

- All activities are included in a generalized flow structured activity.
- The loop is constructed by a backward link.

- The activity **ActB** is modeled as invoke activity. It invokes a simple Java component also contained in the EAR file.
- The loop exit condition is modeled as transition condition of the link to the last human task activity.
- In addition to the process input and output variable, only one variable **InstData** is used. This variable holds the process state information during the lifetime of the process instance.

Steps performed on the WebSphere MQ Workflow side

If you do not want to perform the following steps on a WebSphere MQ Workflow system, skip the steps and start on the WebSphere Process Server side. In this case, use the WebSphere MQ Workflow process instance data file available at the **src** directory of this example.

1. Import and translate process model

Import the file `SeqInstance.fdl` found in the `lib` directory of the package into the WebSphere MQ Workflow runtime database:

Run the command:

```
fmcibie -u<your WebSphere MQ Workflow user ID>
        -p<your WebSphere MQ Workflow password> -i SeqInstance.fdl -ot
        -y<your WebSphere MQ Workflow configuration identifier>
```

2. Adapt the commands to your environment

All the following command files can be found in the **binwin** subdirectory of the package. This document describes the steps performed on a Windows platform. If you are using Linux, you will discover the scripts in the **binlinux** directory of the package. The steps to perform are quite similar to the steps performed on Windows.

To adapt the commands to your environment edit the file `WMQWFSetup.bat`. Here, set the following values:

- a. The WebSphere® MQ installation directory.
- b. The WebSphere MQ Workflow installation directory.
- c. Your WebSphere MQ Workflow user ID and password. The user ID is necessary to access the WebSphere MQ Workflow API. It should have enough authority to read and suspend the process instances of interest.
- d. The WebSphere MQ Workflow configuration identifier of the WebSphere MQ Workflow configuration you are using.

If a class-not-found exception occurs, at first, check that with the settings you made, the referenced JAR files can be found.

All these command files call the JAR file **migrateMQWF.jar** which is available in the **lib** directory of this example. For reference and necessary adaptations, the source code is available in the `src/wmqwf` directory of this example.

3. Create sample instances

Run the command file provided with the example, `WMQWFCreate.bat`. This command creates six process instances in various states.

4. Optionally suspend the process instances

If you want to suspend the process instances before migration, perform the following steps:

- a. **Suspend some process instances**

Run the command file provided with the example, **WMQWFSuspend.bat**. This command suspends all process instances that have no activities that need further processing. In addition, it lists the process instances which need further treatment.

b. **Treatment of active activities**

Of the three checked-out activities, the activity ActD is force-restarted, ActC is force-finished. The checked-out activity ActA needs special treatment, as we assume the following: The successor of ActA is an empty activity which for simplicity reasons represents an activity started automatically. We assume that this activity should not be performed on the WebSphere MQ Workflow side anymore, therefore, we avoid force-finishing the activity ActA while its process is in state RUNNING.

c. **Suspend remaining process instances**

In order to suspend all three running processes, we run **WMQWFSuspendForce.bat**. In contrast to the suspend program mentioned above, even process instances with active activities are suspended. Now, except for one process instance, all process instances are in state SUSPENDED.

d. **Force-finish the last checked-out activity**

Now, ActA can safely be force-finished or checked-in without the risk that the automatic activity ActB is started. When checking this process instance, you will see that although ActA has completed, navigation to the next activity has not been performed.

5. **Extract the process instance data**

Depending on your choice to extract the data of either suspended or running instances, run either **WMQWFReadSuspended.bat** or **WMQWFReadRunning.bat**. This command extracts the process instance data to files which serve as input to instantiating the process instances in WebSphere Process Server.

6. **Remove the process instances**

If you want to clean up the system, use the command file **WMQWFRemove.bat** provided with the example, to remove the process instances. When migrating real-life processes, be sure that the migration completed successfully before removing the WebSphere MQ Workflow process instances from the system.

Steps performed on the WebSphere Process Server side

1. **Deploy the business process**

The file *SeqInstApp.ear* must be deployed and started. This file contains the process definition. It is available in the lib directory of the package. This can be done for example with the WebSphere Process Server Admin Console with the following steps:

- a. On the left side, expand the section *Applications* and select *Enterprise Applications*.
- b. In this panel, click *Install*.
- c. Then, as *Full path*, specify the directory and file name of the file *SeqInstApp.ear* and click *Next*.
- d. On the next panel named *Step 1*, accept the proposed values and click *Next*.
- e. On the panel named *Step 2* select the modules and map them to servers and click *Next*.

- f. After reviewing the setting in the summary page, click *Finish* to start the deployment process.
- g. After the application was installed successfully, click *Save* to save the application to the master configuration.
- h. Then, in the panel *Enterprise Applications* containing the list of installed applications, select the application *SeqInstApp* and press *Start*.

2. Deploy the process instantiate tool

The file *MigrateWPS.ear* contains two components:

- a. A launchClient front end so that this tool can be scripted.
- b. An EJB part which enables the tool to access schema information of the process model used.

Therefore, deploy the EAR file on the WebSphere Process Server server. This file is available in the lib directory of the package.

- a. On the left side, expand the section *Applications* and select *Enterprise Applications*.
- b. In this panel, click *Install*.
- c. Then, as *Full path*, specify the directory and file name of the file *MigrateWPS.ear* and click *Next*.
- d. On the next panel named *Step 1*, accept the proposed values and click *Next*.
- e. On the panel named *Step 2* select the module and map it to a server and click *Next*.
- f. After reviewing the setting in the summary page, click *Finish* to start the deployment process.
- g. After the application was installed successfully, click *Save* to save the application to the master configuration.
- h. Then, in the panel *Enterprise Applications* containing the list of installed applications, select the application *MigrateWPSEAR* and press *Start*.

3. Adapt the commands to your environment

All the following command files can be found in the **binwin** subdirectory of the package. This document describes the steps performed on a Windows platform. If you are using Linux, you will discover the scripts in the **binlinux** directory of the package. The steps to perform are quite similar to the steps performed on Windows.

To adapt the commands to your environment, edit the file **WPSSetup.bat**. Here, set the following values:

- a. The WebSphere Process Server installation directory.
- b. Your WebSphere Process Server user ID and password. This user ID is necessary to access the WebSphere Process Server EJB API. It is also used to create the process instances.
- c. If you want to use an other WebSphere profile instead of the default profile, you must set the option *-profileName=<your profile name>* of the launchClient tool in **WPSInstantiate.bat** and **WPSRemove.bat**.
- d. If you connect to the WebSphere profile with an other port than the standard port 2809, set the option *-CCBootstrapPort=<your port number>* of the launchClient tool in **WPSInstantiate.bat** and **WPSRemove.bat**.

All these command files call the EAR file **migrateWPS.ear** which is available in the **lib** directory of this example. For reference and necessary adaptations, the source code is available in the directory **src/wps** of this example.

4. Run the process instantiate tool

This tool is started with the command **WPSInstantiate.bat**. The instance data extracted from WebSphere MQ Workflow is now read. If you did not create this file with a WebSphere MQ Workflow system, you can alternatively use the WebSphere MQ Workflow process instance data file **procInst001.xml** available in the directory **src** of the example.

5. **Check correctness of process instance migration**

Using the BPC Explorer on WebSphere Process Server and the WebClient on WebSphere MQ Workflow you can compare the state and data of the newly created processes with the original WebSphere MQ Workflow process instances.

6. **Remove the process instances**

If you want to clean up the system, With the command file **WPSRemove.bat** provided with the example, remove the process instances from the system.

7. **Remove the applications**

You can remove the two installed applications. The application **SeqInstApp** can only be removed if you have removed its process instances in the previous step.

Bibliography

WebSphere MQ Workflow product documentation

This documentation is shipped together with the install images of WebSphere MQ Workflow. It can also be found at <http://www-01.ibm.com/software/integration/wmqwf/support/doc.html>. Especially *Getting Started with Buildtime* and the *Programming Guide* might help you understand process model definitions and the usage of the WebSphere MQ Workflow API to extract the process instance data.

WebSphere Process Server information center and WebSphere Integration Developer information center

The product documentation can be found at <http://www-01.ibm.com/software/integration/wps/library/>. In particular, the following information might be helpful:

- An introduction into the capabilities of generalized flows can be found in the WebSphere Integration Developer book, chapter *Developing business processes*.
- The JavaDoc information of the WebSphere Process Server APIs can be found in the WebSphere Process Server book, chapter *Reference*.
- A description of the possible dynamic modifications of process instances at run time can be found in the WebSphere Process Server book, chapter *Product overview*.

FDL migration wizard documentation

The FDL migration wizard of WebSphere Integration Developer is documented in the TechDoc: <http://www.ibm.com/support/docview.wss?rs=2307&uid=swg27013872>. It contains detailed information about features and limitations of the wizard and explains how to migrate process models.

WebSphere MQ Workflow Transition to WebSphere Process Server

This Redbook is available at: <http://www.redbooks.ibm.com/abstracts/sg247282.html>. In addition to the conversion of processes the Redbook covers also the many other aspects of a transition from WebSphere MQ Workflow to WebSphere Process Server.

Business Process Management Samples & Tutorials

The page <http://publib.boulder.ibm.com/bpcsamp/index.html> contains an introduction into the usage of WebSphere Process Server. This can help you especially when developing your first WebSphere Process Server process model or your first program using a WebSphere Process Server API.