

WebSphere® Studio Business Integrator Extensions for  
Windows NT®



# Developer's Guide

*Version 2.1*



WebSphere® Studio Business Integrator Extensions for  
Windows NT®



# Developer's Guide

*Version 2.1*

**Note**

Before using this information and the products it supports, read the information in "Notices" on page 93.

**First Edition (June 2001)**

This edition applies to Version 2.1 of the IBM® WebSphere® Business Integrator (program number 5724-A78) and to all subsequent release and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2001. All rights reserved.

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

---

# Contents

<b>Figures</b> . . . . .	<b>vii</b>
<b>Tables</b> . . . . .	<b>ix</b>
<b>Preface</b> . . . . .	<b>xi</b>
Who should read this book . . . . .	xi
Before you implement WebSphere Business Integrator . . . . .	xi
How to send your comments . . . . .	xii

---

## Part 1. Introduction and solution design . . . . . 1

<b>Chapter 1. Overview</b> . . . . .	<b>3</b>
Process integration flow . . . . .	3
The e-marketplace . . . . .	4
Solutions . . . . .	4
Solution design . . . . .	5
Solution development . . . . .	5
Solution deployment . . . . .	5
Role players . . . . .	6
Solution structure . . . . .	7
<b>Chapter 2. Solution Studio</b> . . . . .	<b>9</b>
Solution Studio overview . . . . .	9
Solution Studio and WebSphere Studio . . . . .	9
Project view . . . . .	10
Publishing view. . . . .	10
ClearCase LT repository . . . . .	11
Solution Studio wizards . . . . .	11
Solution Studio directory structure and usage . . . . .	12
Substitution strings . . . . .	13
Solution Studio templates . . . . .	15
Using Solution Studio. . . . .	16
Starting Solution Studio . . . . .	16
Working with solution artifacts. . . . .	17
Creating a solution artifact . . . . .	17
Modifying a solution artifact . . . . .	17
Exiting from Solution Studio . . . . .	18
<b>Chapter 3. Solution design</b> . . . . .	<b>19</b>
Solution design overview . . . . .	19
Business transformation . . . . .	19

Role players in solution design. . . . .	20
Solution Studio design tools. . . . .	20
Modeling . . . . .	21
Business-process modeling . . . . .	21
Task modeling . . . . .	22
Information modeling. . . . .	22
Organization modeling . . . . .	23
Client modeling. . . . .	23
Completing solution design. . . . .	24

<b>Chapter 4. Designs of the user-registration and purchase order (PO)-management sample templates.</b> . . . . .	<b>25</b>
Overview . . . . .	25
User-registration solution . . . . .	25
Business-process model . . . . .	25
Task model . . . . .	26
Approve registration . . . . .	28
Information model. . . . .	29
Organization model . . . . .	29
Client model. . . . .	30
PO-management solution . . . . .	31
Business-process model . . . . .	31
PO-management overview . . . . .	31
Create new PO . . . . .	32
Modify PO . . . . .	32
Cancel PO . . . . .	33
Approve PO . . . . .	33
PO-approval task model . . . . .	34
Information modeling. . . . .	35
Organization modeling . . . . .	35
Client modeling. . . . .	35

---

## Part 2. Creating artifacts for solutions . . . . . 37

<b>Chapter 5. Creating solution artifacts.</b> . . . . .	<b>39</b>
Overview . . . . .	39
Artifacts for the solution templates . . . . .	39
Creating artifacts . . . . .	40
Registering tools for use . . . . .	41
<b>Chapter 6. Applications</b> . . . . .	<b>43</b>
Overview . . . . .	43

Artifact types . . . . .	44
Supported tool . . . . .	45
Business Integrator considerations . . . . .	45
Creating the artifact . . . . .	46
Artifact example . . . . .	46
<b>Chapter 7. Business Process . . . . .</b>	<b>47</b>
Overview . . . . .	47
Artifact types . . . . .	47
Supported tools . . . . .	47
Business Integrator considerations . . . . .	48
Creating the artifacts . . . . .	48
Creating work flows . . . . .	48
Artifact examples . . . . .	49
<b>Chapter 8. Clients . . . . .</b>	<b>51</b>
Overview . . . . .	51
Artifact types . . . . .	51
Supported tools . . . . .	51
Creating the artifacts . . . . .	51
Artifact example . . . . .	53
Browser user registration JSP file . . . . .	53
<b>Chapter 9. Data . . . . .</b>	<b>55</b>
Overview . . . . .	55
Artifact types . . . . .	55
Supported tool . . . . .	55
Creating the artifact . . . . .	55
Artifact examples . . . . .	56
Registration Services . . . . .	56
Workflow Services . . . . .	56
Registration BDO . . . . .	56
<b>Chapter 10. Messages . . . . .</b>	<b>57</b>
Overview . . . . .	57
Artifact types . . . . .	57
Supported tools . . . . .	58
Creating the artifacts . . . . .	58
MQSeries Integrator . . . . .	58
DataInterchange. . . . .	59
Partner Agreement Manager (PAM) . . . . .	59
Overview of creating and editing public processes . . . . .	60
MQSeries Queue Definition . . . . .	61
Java Message Service . . . . .	63
<b>Chapter 11. Roles. . . . .</b>	<b>69</b>
Overview . . . . .	69
Artifact types . . . . .	69
Supported tool . . . . .	69

Business Integrator considerations . . . . .	69
Creating the artifacts . . . . .	69
Artifact example . . . . .	71
LDAP configuration file . . . . .	71
Policy Director commands file . . . . .	72
Policy Director map file . . . . .	72
Policy Director configuration file . . . . .	72

<b>Chapter 12. Servlet . . . . .</b>	<b>73</b>
Overview . . . . .	73
Artifact types . . . . .	73
Supported tool . . . . .	73
Business Integrator considerations . . . . .	74
Creating the artifact . . . . .	74
Creating the servlet artifact file. . . . .	74
Artifact example . . . . .	75

<b>Chapter 13. Resources and rules (Personalization) . . . . .</b>	<b>77</b>
Overview . . . . .	77
Artifact types . . . . .	78
Supported tool . . . . .	78
Business Integrator considerations . . . . .	78
Creating the artifacts . . . . .	78

---

## Part 3. Solution deployment . . . . . 79

<b>Chapter 14. Solution publishing. . . . .</b>	<b>81</b>
Overview . . . . .	81
Before creating the deployable solution package . . . . .	82
Deployable solution package . . . . .	83
Solution deployment model. . . . .	83
Creating the solution metadata file . . . . .	84
Creating a deployable solution package . . . . .	84

<b>Chapter 15. Solution deployment . . . . .</b>	<b>89</b>
Deploying the solution . . . . .	89
Deploying the solution package . . . . .	89

---

## Part 4. Appendixes . . . . . 91

<b>Notices . . . . .</b>	<b>93</b>
Trademarks . . . . .	95

<b>Bibliography . . . . .</b>	<b>97</b>
IBM WebSphere Business Integrator library . . . . .	97
Related documentation . . . . .	98

WebSphere Partner Agreement Manager library . . . . .	99	Other libraries . . . . .	99
DataInterchange library . . . . .	99	<b>Index . . . . .</b>	<b>101</b>





---

## Figures

1.	Project and publishing views of a Solution Studio user-registration solution . . . . .	10
2.	The structure of the WebSphere Studio templates directory . . . . .	16
3.	Overview of business-processing modeling . . . . .	21
4.	Process flow of the user-registration solution . . . . .	25
5.	The approval process as modeled in MQSeries Workflow Buildtime . . . . .	27
6.	A subprocess as modeled in MQSeries Workflow Buildtime . . . . .	28
7.	Process for managing purchase orders	31
8.	The Wizard Registration wizard . . . . .	41
9.	Workflow wizard . . . . .	48
10.	Buildtime GUI for creating FDL files	49
11.	The WebSphere Artifact Configuration wizard . . . . .	52
12.	Sample JSP file as shown in WebSphere Studio Page Designer . . . . .	54
13.	Sample JSP file as resolved on a browser	54
14.	MQSeries Queue Definition wizard	62
15.	The Application Configuration wizard	64
16.	The Adapter Routing wizard . . . . .	65
17.	The <b>General</b> page of the Adapter Routing wizard . . . . .	66
18.	The <b>Command</b> page of the Adapter Routing wizard . . . . .	67
19.	The <b>Worker</b> page of the Adapter Routing wizard . . . . .	68
20.	Solution Deployment wizard . . . . .	70
21.	Roles folder containing wizard-built and stored configuration files . . . . .	71
22.	Publishing the Business Integrator solution by using WebSphere Studio . . . . .	82
23.	Publishing the solution in WebSphere Studio . . . . .	85
24.	Compressed solution elements for the deployable solution package. . . . .	86
25.	Deployable solution package file	87
26.	Storing the deployable solution package for movement to the run-time test or production environment . . . . .	88



---

## Tables

1. Solution Studio wizards . . . . .	11	9. Information model . . . . .	35
2. Solution Studio directory structure and usage . . . . .	12	10. Client model for PO-management solution approve-PO process . . . . .	35
3. Substitution specifications . . . . .	13	11. Types of artifacts for solutions based on the folder of the Solution Studio template . . . . .	40
4. Components, subcomponents, and artifact types . . . . .	14	12. Relationship between MQSeries Adapter Builder and the MQSeries Adapter Builder wizard . . . . .	44
5. Supported types and corresponding products . . . . .	14	13. Artifact type and tool for message element. . . . .	57
6. Data required by the tasks in an RFQ process model . . . . .	22	14. Types of solution artifacts deployed to Business Integrator managers . . . . .	81
7. Information model . . . . .	29		
8. Client model for user registration solution . . . . .	30		



---

## Preface

This book provides information for creating an IBM® WebSphere® Business Integrator solution by using IBM WebSphere Studio Business Integrator Extensions (hereafter referred to as Solution Studio). This book covers the design, development, and deployment phases needed to create Business Integrator solutions to business problems.

---

### Who should read this book

This book is intended for the following users:

- Business transformation and Business Integrator consultants responsible for defining and refining business processes
- Solution developers responsible for composing and implementing solutions
- Administrators responsible for deploying solutions to the run-time environment

It assumes an understanding of business-to-business processes.

### Before you implement WebSphere Business Integrator

WebSphere Business Integrator uses multiple underlying products and technologies to support the solutions that you create and run. In general, before you implement Business Integrator, you need to understand the underlying products and technologies that support your solution.

Before you implement Business Integrator, you or other members of your organization need to be generally skilled in the activities listed below for similar solutions, products, and underlying products and technologies. If you and other members of your organization do not possess these skills, you need to obtain assistance from qualified services staff, either from IBM or from third parties, to implement Business Integrator. You must be prepared to use the documentation of the underlying products and technologies. (This documentation is provided with Business Integrator or otherwise from IBM.)

When you plan, install, and configure Business Integrator, you need to understand how to install and configure the underlying products and technologies that you use in your installation. Business Integrator provides the installation of most of the underlying products and technologies into its run-time environment. However, you might need to install and configure certain underlying products separately into either the build-time or run-time environment. You might also need to diagnose and correct installation problems with underlying products and technologies.

Before you design, develop, and publish solutions, you need to be:

- Generally familiar with system-integration techniques in a business environment.
- Prepared to use the tools of the underlying products and technologies that your solution requires.
- Familiar with the run-time behavior of the underlying products and technologies that your solution requires.
- Familiar with modeling concepts and techniques such as Unified Modeling Language and related tools, with state machine concepts, and with visual flow composition-modeling concepts and techniques.
- Familiar with Internet and Electronic Data Interchange (EDI) concepts and technologies, if required by your solution.
- Prepared to research the existing applications, systems, and networks that you integrate with Business Integrator.
  - Inside your enterprise, they can be known as legacy systems, back-end systems, enterprise applications, or endpoint applications.
  - Outside your enterprise, they can be known as trading networks, private EDI networks, or similar networks that your solution requires.

Before you deploy, run, manage, diagnose, and tune Business Integrator, you need to be prepared to use the management, trace, audit, exception handling, diagnostic and related tools of the underlying products and technologies that support your solution. You need to be prepared to understand the solution itself to the degree needed for these tasks.

---

## How to send your comments

IBM welcomes your comments. You can send your comments by any one of the following methods:

1. Electronically to the network ID listed below. Be sure to include your entire network address if you wish a reply.

idrcf@hursley.ibm.com

2. By FAX, use the following numbers:

UK: 01962-870229

Other countries: +44-1962-870229

3. By mail to the following address:

User Technologies  
Mail Point 095  
IBM United Kingdom Laboratories  
Hursley Park  
Winchester

Hampshire  
S021 2JN  
United Kingdom





---

# Part 1. Introduction and solution design

<b>Chapter 1. Overview</b> . . . . .	3	Solution Studio design tools. . . . .	20
Process integration flow . . . . .	3	Modeling . . . . .	21
The e-marketplace . . . . .	4	Business-process modeling . . . . .	21
Solutions . . . . .	4	Task modeling . . . . .	22
Solution design . . . . .	5	Information modeling. . . . .	22
Solution development . . . . .	5	Organization modeling . . . . .	23
Solution deployment . . . . .	5	Client modeling. . . . .	23
Role players . . . . .	6	Completing solution design. . . . .	24
Solution structure . . . . .	7		
 		<b>Chapter 4. Designs of the user-registration and purchase order (PO)-management sample templates.</b> . . . . .	25
<b>Chapter 2. Solution Studio</b> . . . . .	9	Overview . . . . .	25
Solution Studio overview . . . . .	9	User-registration solution . . . . .	25
Solution Studio and WebSphere Studio . . . . .	9	Business-process model . . . . .	25
Project view . . . . .	10	Task model . . . . .	26
Publishing view. . . . .	10	Approve registration . . . . .	28
ClearCase LT repository . . . . .	11	Information model . . . . .	29
Solution Studio wizards . . . . .	11	Organization model . . . . .	29
Solution Studio directory structure and usage . . . . .	12	Client model. . . . .	30
Substitution strings . . . . .	13	PO-management solution . . . . .	31
Solution Studio templates . . . . .	15	Business-process model . . . . .	31
Using Solution Studio. . . . .	16	PO-management overview . . . . .	31
Starting Solution Studio . . . . .	16	Create new PO . . . . .	32
Working with solution artifacts. . . . .	17	Modify PO . . . . .	32
Creating a solution artifact . . . . .	17	Cancel PO . . . . .	33
Modifying a solution artifact . . . . .	17	Approve PO . . . . .	33
Exiting from Solution Studio . . . . .	18	PO-approval task model . . . . .	34
 		Information modeling. . . . .	35
<b>Chapter 3. Solution design</b> . . . . .	19	Organization modeling . . . . .	35
Solution design overview . . . . .	19	Client modeling. . . . .	35
Business transformation . . . . .	19		
Role players in solution design. . . . .	20		

This part discusses the use of Solution Studio and provides overall solution design concepts. The creation of artifacts for solutions is discussed in “Part 2. Creating artifacts for solutions” on page 37.



---

## Chapter 1. Overview

The IBM WebSphere Business Integrator (referred to as Business Integrator throughout this documentation) Entry configuration and Enterprise configuration products provide a secure run-time environment in which business processes are fully integrated. Through a set of services, it provides a run-time environment in which customized solutions to business process problems can be deployed and executed. This run-time environment enables a business to easily install and integrate best-of-breed applications while leveraging its investment in legacy applications.

The IBM WebSphere Studio Business Integrator Extensions (referred to as Solution Studio throughout this documentation) provides a build-time environment in which customized solutions to business problems can be created. Solution Studio provides build-time support for constructing, modifying, and configuring solutions for deployment in the Business Integrator run-time environment. Solution Studio consists of a rich mix of tools that are used to create solutions based on customizing a prebuilt solution template.

This book describes how a solution can be designed, created, customized, and made ready for deployment in the run-time environment.

---

### Process integration flow

Every aspect of business operations is based on a set of business processes. In current business environments, these processes are often performed in many different ways. Information can come from a hard-copy catalog or an online database. Communications can occur by telephone, fax machine, or e-mail. Several software packages are often used to fulfill a single process. The activities for performing the process can be disjointed, in some cases highly manual, and often paper intensive. The potential for problems can exist at many points. Often the process does not run smoothly and can sometimes break down altogether. This can lead to higher costs of doing business, lost opportunities, and slower response time.

The Business Integrator platform automates the flow of business processes electronically and ties together the software packages that are used to perform each task. This integration of process flow results in more efficient processes and can dramatically reduce the time required to perform the process tasks.

---

## The e-marketplace

An electronic marketplace (or e-marketplace) is a trading community of businesses or enterprises (known as trading partners) engaged in e-business. The trading partners can participate in a variety of roles (such as buyers, vendors, and administrators). The trading partners participate in a community to exchange goods and services. Therefore, an e-marketplace is a virtual web of supply chains providing a market-based coordination mechanism across participating organizations.

For the e-marketplace, Business Integrator becomes a procurement network that brings together a number of enterprise vendors who offer goods and services to enterprise buyers. This procurement network matches buyers and vendors based on an efficient market mechanism that provides lower sourcing and transaction costs, access to a global market, reduced time-to-market cycle, and highly competitive pricing. This procurement network, based on Business Integrator services and solutions, offers the following advantages to members who are enterprise buyers and vendors:

- An intelligent market search for vendors, buyers, products, requests for quotes, and offers
- A negotiation-based business communications model
- A trusted, secure environment for e-commerce
- An integrated market catalog consisting of a variety of trading instruments (such as products, services, capacity, inventory, and promotions)
- Transaction management for extended business-to-business transactions
- Collaborative business processes to facilitate the integration of information and decisions between enterprises in the marketplace

The solutions created by using Business Integrator Solution Studio automate business processes and provide the advantages of these electronic processes to trading partners in the e-marketplace.

---

## Solutions

Solutions are designed, created, and made ready to deploy by using a variety of tools, many integrated into Business Integrator through Solution Studio. Usually, the solutions are tested in a run-time test environment before they are deployed to a run-time production environment.

Solutions follow a life cycle having the following three phases:

- Solution design
- Solution development
- Solution deployment

These phases are described in the following sections.

## **Solution design**

The solution-design phase can also be thought of as the business-transformation phase. In this phase, existing business processes are examined to develop a common understanding of the processes and to identify opportunities to streamline business functions that are both internal and external to the enterprise. Recommendations can then be made for improving the enterprise-wide information system. The business problem is clearly defined in terms of business processes, tasks, information, and organization. After this is completed, a solution is designed for the problem. The documented design is used in the next phase, solution development.

The design phase includes the following activities:

- Business-process modeling
- Task modeling
- Information modeling
- Organization modeling
- Client modeling

For more information on solution design, see “Chapter 3. Solution design” on page 19.

## **Solution development**

In the solution-development phase, the solution design is used to create a solution that can be deployed in a run-time environment. In this phase, Solution Studio tools are used to create and save the artifacts that compose a solution and to make those solution artifacts deployable in the run-time environment. Solution Studio is used to create the elements that integrate the business-process flow based on the solution templates that Solution Studio provides.

See “Chapter 2. Solution Studio” on page 9 for detailed information on Solution Studio and templates.

## **Solution deployment**

The solution-deployment phase activities take place in both the build-time and run-time environments. The solution is *published* in Solution Studio into a deployable solution package as the final development step in build time. Then the deployable solution package is moved to the run-time environment and the artifacts are installed and activated on the appropriate machines where Business Integrator services are running. This step is known as *deployment*, and because it takes place outside the build-time environment, the subject is

covered briefly in “Chapter 15. Solution deployment” on page 89. For more information on deploying a solution, refer to the *WebSphere Business Integrator Run Time* book.

---

## **Role players**

Many roles are involved in developing a solution during each phase of the solution life cycle. Typically, the solution is developed by a team with different skills and areas of expertise. In some cases, a team member can play more than one role, while other team members play a single role. The team can consist of one or more players performing the following roles for developing solutions:

### **Business transformation consultants**

Define and reengineer the processes, tasks, information, and organization of the business.

### **Business Integrator consultants**

Work with business transformation consultants to define business processes. Work with application/domain specialists to define workflows, message flows, and message sets based on the business processes, and also to define mapping and mediation requirements for messages.

### **IT architects/specialists**

Define system requirements and work with other role players to develop models. Work with application deployers to deploy Business Integrator solutions and to diagnose and resolve any problems during deployment.

### **Program managers**

Manage the development, deployment, and support of Business Integrator solutions.

### **Application/domain specialists**

Work with Business Integrator consultants to develop workflows, message flows, message sets, and message-mapping and mediation requirements; work with application developers to define business objects and interaction scenarios; and work with IT architects/specialists and application developers to define requirements for application adapters.

### **Application development administrators**

Install the tools that are used to develop the solution elements and artifacts; maintain the version-control system.

### **Application developers**

Develop artifacts that make up the solution; build adapters, enterprise

beans, JavaServer Pages™ (JSP files), servlets, and other artifacts and files that are specified in the solution design documents.

### **Deployers**

Configure the solution in the build-time environment, making it ready for deployment; deploy the solution in the run-time environment.

### **Support analysts**

Work with IT architects/specialists to support a deployed solution; work with system administrators to solve problems.

### **System administrators**

Work with IT architect/specialists to support a deployed solution; work with support analysts to solve problems.

During the solution-development phase, a typical team consists of an application development administrator who has overall responsibility for the technical management and development of the solution. Working with this person is a group of application developers numbering around ten tools specialists. For most effective use of Solution Studio on a single solution, a maximum team size of 20 is desirable.

---

## **Solution structure**

In the development phase, Solution Studio solutions have a hierarchical structure. From a top-down view, a solution hierarchy resembles the following:

Solution

Elements or components

Subcomponents

Artifacts

Artifact types and associated configuration scripts

A loose correspondence exists between the structure of a solution and the structure of a project as it is represented in the WebSphere Studio workbench based on the Solution Studio solution templates. For more information about Solution Studio and templates, see “Solution Studio overview” on page 9 and “Solution Studio templates” on page 15.





---

## Chapter 2. Solution Studio

This chapter provides an overview of Solution Studio. Online help provides assistance for using Solution Studio wizards to perform tasks such as creating solution artifacts and performing solution-development and management tasks.

---

### Solution Studio overview

Solution Studio consists of a rich mix of tools that are used to develop a Business Integrator solution by customizing existing solutions or creating new solutions. It consists of a user interface provided by WebSphere Studio, a group of wizards used to perform tasks such as starting tools and importing artifacts that are created with them, and a repository provided by ClearCase LT that provides file sharing and storage and enforces version control and change management.

### Solution Studio and WebSphere Studio

For Solution Studio, WebSphere Studio provides a graphical user interface (GUI) through which Business Integrator solution developers create, customize, and manage elements of a solution in the build-time environment. WebSphere Studio provides many tools that can be used to create solution artifacts. Solution Studio extends the integration of the tools that are used to create these solution elements and artifacts. An example of the WebSphere Studio workbench as used by Solution Studio is shown in the following figure. The figure shows both the project and publishing views for a user-registration solution.

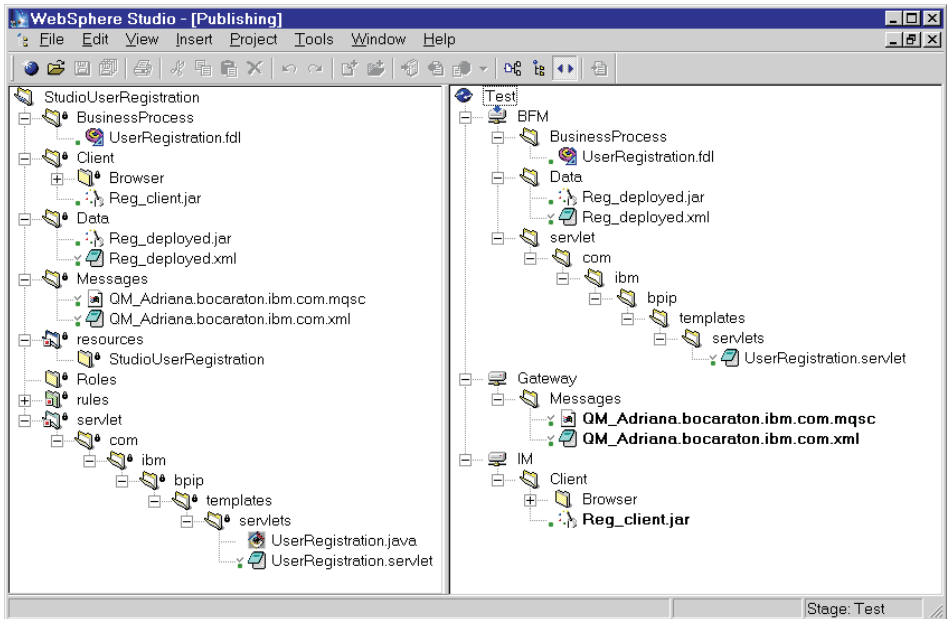


Figure 1. Project and publishing views of a Solution Studio user-registration solution

The WebSphere Studio workbench is divided into views or panes. The left pane provides a project view. The right pane provides a publishing view. Most wizards used to access the tools that are used to create a Business Integrator solution or configuration files are launched from the **Tools** menu in WebSphere Studio. The wizard used to create the deployable solution package (see “Creating a deployable solution package” on page 84) is launched from the publishing view.

### Project view

The project view provides a view of the ClearCase LT project directory structure. This directory structure contains artifacts and other information for the solutions you are working on.

### Publishing view

The right pane is the publishing view. It provides a logical view of the solution to be deployed to a test or production run-time environment. When you publish a solution, you create a deployable solution package that is reflected in this view. For more information about deployable solution packages, see “Chapter 14. Solution publishing” on page 81. For information on deploying solutions to a run-time environment, refer to the *WebSphere Business Integrator Run Time* book.

## ClearCase LT repository

The ClearCase LT repository is the storage place for solution artifacts. After the repository is set up, its file-based system can be accessed from within WebSphere Studio to check solution artifact files into and out of the repository. The repository is located on the ClearCase LT server, and these artifacts are accessed by ClearCase LT clients. ClearCase LT manages version control for all of the artifact files created for each solution.

Solution artifacts stored in the repository are organized in a hierarchy of folders. Each Business Integrator solution is set up as a WebSphere Studio project. A folder contains related solution artifacts that capture business information, such as process flows, data objects, messages, and organizational roles. The artifacts are arranged in a hierarchical fashion within a set of subdirectories in the repository.

Typically, an application development administrator sets up the ClearCase LT server and clients as part of the installation and configuration of Solution Studio. Information on installing and configuring Solution Studio is found in the *WebSphere Studio Business Integrator Extensions Installation Guide*. It includes information on setting up a ClearCase LT repository.

## Solution Studio wizards

Solution Studio provides wizards for use in developing your solutions. The wizards perform a variety of tasks. For instance, many of the tools (products such as MQSeries Workflow Buildtime and MQSeries Adapter Builder) that you use to create the artifacts for your solution are launched by using Solution Studio wizards. These wizards are also used to store the created artifacts in the project folders in the ClearCase LT repository. Solution Studio provides online help for using the wizards. Table 1 describes the wizards used in developing solutions.

Table 1. Solution Studio wizards

Wizard	Description
Wizard Registration	Used to register new wizards.
MQSeries Queue Definition	Used to create and configure MQSeries queues for messaging within the Business Integrator infrastructure, between the infrastructure and back-end applications, and between the infrastructure and gateways.
WebSphere Artifact Configuration	Used to create the configuration files for Web-based artifacts (for example, enterprise beans, servlets, HTML files, JSP files, and graphics files).
Solution Deployment	Used to create LDAP configuration files and command and map files for Policy Director.

Table 1. Solution Studio wizards (continued)

Wizard	Description
Application Configuration	Used to configure adapters for use with applications. This includes Java Message Service (JMS) configuration files.
Adapter Routing	Used to create and route message types. These messages work with your configured MQSeries adapters and enterprise worker beans. <b>Note:</b> This wizard can be opened only from the Application Configuration wizard's Application page.
MQSI	Used to launch MQSeries Integrator to create message mapping and translation. Creates message sets and exports. This wizard can be used only if MQSeries Integrator is installed on the client workstation; the wizard must be registered by using the Wizard Registration wizard.
MQ Adapter Builder	Used to launch MQSeries Adapter Builder to create application adapters and configuration files. This wizard can be used only if MQSeries Adapter Builder and is installed on the client workstation; the wizard must be registered by using the Wizard Registration wizard.
Workflow	Used to launch MQSeries Workflow to define the work or process flow and import the resultant FDL files. This wizard can be used only if MQSeries Workflow and is installed on the client workstation; the wizard must be registered by using the Wizard Registration wizard.

### Solution Studio directory structure and usage

Solution Studio defines a directory structure for storing artifacts. The  $x:\backslash bi\backslash$ Solution directory is the parent of the directories listed in Table 2, where  $x$  is the drive letter on which Solution Studio is installed and  $bi$  is the Solution Studio installation directory.

Table 2. Solution Studio directory structure and usage

Directory	Usage
lib	Contains .class files. This directory is listed in the run-time environment's CLASSPATH variable so that any programs placed here can be located by the run time.
bin	Contains .ddl files. This directory is listed in the run-time environment's PATH variable so that any programs placed here can be located by the run time.

Table 2. Solution Studio directory structure and usage (continued)

Directory	Usage
properties	Contains .properties files. This directory is listed in the run-time environment's CLASSPATH variable so that any files placed here can be located by the run time.
xml	Contains .xml and .dtd files. No environment variables are set to the location of this directory.
utils	Contains .bat files that are expected to be used often during the lifetime of the solution. No environment variables are set to the location of this directory.
config	Contains any type of file that can be used to configure the solution. These files are expected to be temporary. No environment variables are set to the location of this directory.
sql	Contains .ddl files that are used for database commands (for example, to create or modify database schemas). These files are expected to be temporary. No environment variables are set to the location of this directory.

### Substitution strings

In some solutions, you must provide values that are not available until deployment time. For instance, a solution can require the IP address of a machine that is not known until the solution is deployed. Solution Studio and the Business Integrator deployment tools provide special characters and strings that are substituted for the correct values at deployment time. The correct values are obtained from the topology repository and resolved during deployment. The substitution strings are listed in Table 3. Valid components and subcomponents, along with the types of artifacts they support, are listed in Table 4 on page 14. Supported product types are listed in Table 5 on page 14.

Table 3. Substitution specifications

Keyword	Substitution string
@{hostname(comp,subcomp)}	Host computer name
@{ipaddress(comp,subcomp)}	IP address

Table 3. Substitution specifications (continued)

Keyword	Substitution string
@{mqportnumber( <i>comp</i> , <i>subcomp</i> )}	Port number of first installed MQSeries queue manager
@{wsbiInstallDir( <i>comp</i> , <i>subcomp</i> )}	Business Integrator installation directory
@{installDir: <i>type</i> ( <i>comp</i> , <i>subcomp</i> )}	Installation directory of the product specified by <i>type</i> on the machine on which the <i>comp</i> , <i>subcomp</i> components are installed. See Table 5 for a list of valid product types.

See Table 4 for a list of valid component and subcomponent (*comp*,*subcomp*) pairings. The components listed in Table 4 are expanded in Table 14 on page 81.

Table 4. Components, subcomponents, and artifact types

Component	Subcomponent	Artifact types
BFM	WAS	Enterprise beans, adapters, queue configuration files
	WF	FDL files, adapters
IM	IM	HTML, GIF, and JPG files; JSP files and servlets
TAM	LDAP	XML files
IDM	DI	EIF, DTD, DIC, and DICMD files
	MQSI	MRP files
Gateway	PAM	PXML files
Endpoint	<i>endpoint_name</i>	Adapters, queue configuration files

Table 5. Supported types and corresponding products

Type	Product
b2bi_mgrs	WebSphere Business Integrator
csa	Common System Architecture
db2	DB2
db2rc	DB2 run-time client
dce	Distributed Computing Environment
di	DataInterchange

Table 5. Supported types and corresponding products (continued)

Type	Product
gwa	Gateway Adapter
http	IBM HTTP Server
hursley_sm	Solution Management
jdk	Java Development Kit
jmx	JMX Agent
ldap	SecureWay Directory Service
mq	MQSeries
mqak	MQSeries Adapter Kernel
mqjms	MQSeries with JMS support
mqpubsub	MQSeries with Publish/Subscribe support
mqsi	MQSeries Integrator
mqwf	MQSeries Workflow
pam	Extricity Gateway
pav	Partner Agreement View
pd_auth	Policy Directory Authorization
pd_console	Policy Directory Console
pd_mgmt	Policy Director Management Server
pd_netseat	Policy Director NetSeat
pd_runtime	Policy Director Runtime
was	WebSphere Application Server
was_personalization	WebSphere Personalization

---

## Solution Studio templates

When Solution Studio is installed on ClearCase LT clients, two solution templates are installed in the WebSphere Studio root directory's templates subdirectory. These templates provide a directory structure that is standard for most solutions you create. Figure 2 on page 16 shows the directory structure for the User Registration template.



Figure 2. The structure of the WebSphere Studio templates directory

The Solution Studio template provides your solution with directories and subdirectories where specific types of artifacts are stored. Expansion of the directory structure eventually leads to the artifacts themselves and any associated configuration scripts. The highest-level folder in the project view is the project folder, bearing the name of the solution (in this case, UserRegistration). It corresponds to the ClearCase LT repository.

---

## Using Solution Studio

Solution Studio enables you to create and publish solution artifacts. Online help is provided for using Solution Studio wizards to perform many tasks associated with creating artifacts, using tools, and creating a deployable solution package. Depending on how the products are installed on client machines, help can be available online or through product publications for the tools that are launched through Solution Studio.

**Note:** This section describes only tasks that pertain to creating a Business Integrator solution by using Solution Studio. It assumes that WebSphere Studio, Solution Studio, and the tools that are required for creating artifacts are installed, and that ClearCase LT is being used for version management.

### Starting Solution Studio

To start working with a solution in Solution Studio, perform the following steps:

1. Start WebSphere Studio.
2. In the project view of the WebSphere Studio workbench, select a project to work on.
3. To create or modify artifacts, use the wizards that integrate available products or tools. Any tools must be installed and registered so that they are available from the WebSphere Studio **Tools** menu.



## Working with solution artifacts

This section describes tasks you can perform when working with artifacts. Typically, the project you work on is already set up, and tasks include creating new artifacts or modifying existing artifacts.

### Creating a solution artifact

To create a new artifact for a solution, ClearCase LT must be running on the server and the client, and WebSphere Studio must be running on the client machine where the artifact is to be created. The tool that you use to create the artifact must be installed on the client machine. In most cases, the tool is a separate product that is integrated to work with Solution Studio. The wizard for this tool must be installed and registered.

To begin creating a new artifact, perform the following steps:

1. Expand the project directory structure in the project view.
2. Select the subfolder that is to contain the artifact you create.
3. With a project subfolder highlighted, click **Tools** → **Wizards**, then select the wizard for the tool you are using to create an artifact. This launches the wizard.

**Note:** You can start any wizard from any subfolder. This provides flexibility in creating and organizing the solution. The project subfolders can be used to help you structure your solution during development, or you can store all types of artifacts in the same subfolder in the project view. If you store all artifacts in the same subfolder, the task of publishing your solution can be more complex.

4. From the wizard, launch the tool to create the artifact or import artifacts.

When you are finished creating the artifact, use the wizard to store it in the subfolder you selected. This stores the artifact in the ClearCase LT repository, places it under version control, and makes it available for other Application Developers to work on.

### Modifying a solution artifact

Artifacts can require modification due to design changes or errors found during testing of the solution. While the solution is under development, artifacts can be in various states of completion and require constant modification as part of the work in progress. To modify an existing solution artifact, perform the following steps:

1. Expand the project directory structure to the subfolder and file level.
2. Select the artifact file you want to modify.
3. With the file highlighted, click **Tools** → **Wizards**, then the wizard used to create and store the artifact.

4. Modify the artifact as appropriate.

When you are finished modifying the artifact, use the wizard to store it in the appropriate subfolder of the project. Storing the artifact checks it into the ClearCase LT repository and places it under version control. Other Application Developers working on the same solution at other client machines can then check artifacts out of the ClearCase LT repository, work on them, and check them back in.

### **Exiting from Solution Studio**

To exit from Solution Studio, select **File** → **Exit** from the menu bar.

---

## Chapter 3. Solution design

This chapter describes the activities involved in designing a Business Integrator solution. It provides information on the modeling used to design various parts of the solution.

---

### Solution design overview

Creating an e-business solution begins with solution design. For Business Integrator, solution design begins with a definition of the business problem that the solution is to fix.

In the solution-design phase, the business problem is clearly defined in terms of business processes, tasks, information, organization, and client interfaces. This phase produces high-level design specifications that are used to create the solution and its artifacts. As it exists in Solution Studio, a solution is a work in progress. During the design phase, it is a set of models that can be document files or the output of one or more modeling tools. Some of the artifacts of this step are actually deployed with the solution to the run-time environment; however, most are not. The design phase results in all of the information that an Application Developer needs to create particular artifacts by using particular products or tools.

### Business transformation

This phase begins with business transformation. Business transformation defines and examines existing business processes to develop a common understanding of business requirements and to identify opportunities to streamline both internal and external business functions. A number of modeling tools can be used to capture the business processes of the problem. When the problem is clearly defined, the information can be used to model a solution to the problem.

The next step in the design phase primarily involves modeling the solution. Modeling the solution begins with modeling the e-business process. This step is the basis for modeling the following parts of a solution:

- Tasks
- Information
- Organization
- Client

The modeling exposes information about the design of the solution that can be translated into artifacts for the elements of a solution, including the following:

- Applications
- Business process
- Client
- Data
- Messages
- Roles
- Resources and rules (these facilities are provided by WebSphere Personalization and are available only in the Enterprise configuration)
- Servlets

These elements of the solution are reflected in the standard Solution Studio templates that are installed with Solution Studio and are available to anyone using Solution Studio for solution design and development.

### **Role players in solution design**

This phase involves the following role players:

- Business transformation consultants
- Business Integrator consultants
- IT architects/specialists
- Application/domain specialists

### **Solution Studio design tools**

Solution Studio provides an integrated development environment from which you can launch different tools. For instance, you can launch MQSeries Workflow to model processes or workflows. The MQSeries Workflow Buildtime tool creates artifacts that can be used by the MQSeries Workflow Runtime engine. The artifacts created as part of the design phase can then be part of the published and deployed solution.

Other tools can be used outside of Solution Studio for modeling and their artifacts imported into Solution Studio so that Application Developers can refer to them as they create solution artifacts. These can be test or other document files, graphs and charts, and diagrams. If they can be stored in electronic format, they can be part of the solution's design artifacts. These artifacts do not become part of the published solution.

---

## Modeling

Solution design can be approached from many angles. One way to approach it, and the method discussed in this book, is to model your solution to an e-business problem based on the flow of the transformed business process.

---

### Business-process modeling

The business-process model is created during the solution-design phase. *Business processes* are activities that are performed to meet the needs and requirements of a business. These activities start as a broad list of processes that is refined to a detailed list of tasks. From this task list, data entities, role-players who perform the tasks, and other information needed to design a solution can be identified. Figure 3 illustrates the steps and results in modeling your solution.

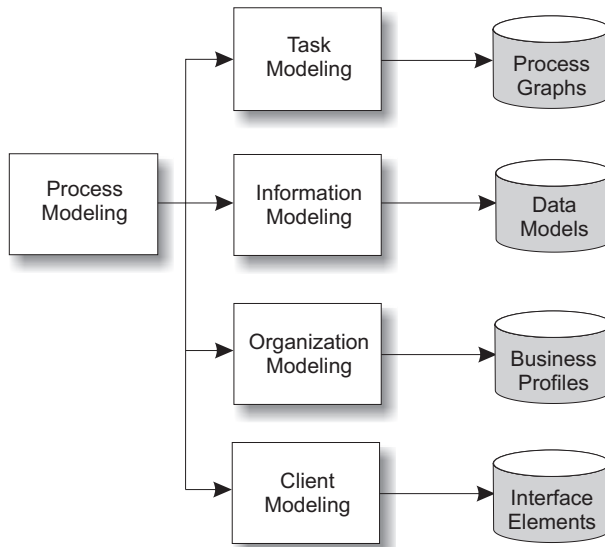


Figure 3. Overview of business-processing modeling

The following is a list of activities that are involved in defining business processes:

#### Process modeling

Captures the current and revised business processes, and defines improved processes.

#### Task modeling

Identifies and describes each task of the business process.

**Information modeling**

Identifies and describes the data exchanged during the business process.

**Organization modeling**

Identifies and describes the role players who perform the tasks.

**Client modeling**

Identifies the interface for Web-based and program-based clients.

The artifacts from modeling can be stored with the solution in the ClearCase LT repository so that Application Developers can refer to them while creating deployable solution artifacts.

**Task modeling**

Task modeling provides information about the tasks involved in the business-process flow. *Tasks* are the actual steps taken to complete a specific business process. These tasks can be performed by humans or be automated (executed by applications).

- Human tasks are executed through the client interfaces to browser views, which are dynamically rendered based on the role of the user and the process context.
- Automated tasks engage applications through Business Data Objects (BDOs).

**Information modeling**

*Information* refers to data that is exchanged as part of the progress of the business process. An example of data is information that a user keys into the system to generate results. For example, before you log on to a solution, you need to enter your user ID and password. The information model captures the pertinent data that the solution requires to perform the business processes.

For example, Table 6 on page 23 lists the data required by each task in a Request for Quote (RFQ) process model. As the table shows, data is directly related to the task being performed.

Table 6. Data required by the tasks in an RFQ process model

Tasks	Data
Create RFQ	RFQ number, process-start data
Enter RFQ Data	Creation date, creation time, address of requester, purchase order (PO) name, respond-by date, and request cost
Add Vendor	(no data)
View Vendor List	Vendor name, product list
Select Vendor	Vendor name, address
Add Item	Item description, item number
View Catalog	Internal part number, part number, quantity, description, price
Select Item	(no data)
Submit RFQ	Process-end information, log request data

## Organization modeling

The organization model includes concepts relating to enterprise roles (who does what) and groups.

Organizations are broken into *organizational units*. In an e-marketplace, the organizational unit can include internal groups (such as buyers and Web-portal hosts) and external groups or trading partners (such as vendors). Each member or user of the e-marketplace represents a specific role, such as vendor, buyer, or administrator.

In an e-marketplace example, member types include vendor-org, buyer-org, and portal-host. An e-marketplace therefore includes participants who are vendor organizations, buyer organizations, plus a Web portal host who administers and executes the e-marketplace.

A solution involves high-level business processes with specific roles. For instance, the following roles can be defined in the organization model for a PO-management solution:

- Buyer
- Financial approver
- Administrator

Users' roles determine the tasks that they perform or are permitted to perform in completing a business process.

## Client modeling

The client model describes the flow of screens in the process and the views that users are presented with at their browsers. This is typically the layout

and content of the HTML page on the browser. The views are role-based and activity- or task-based in that the design of a particular user's view is based on the user's role in the e-marketplace and the specific task that the user is performing at a given point in the business process.

The flow of screens is a branching tree diagram that shows where user interaction takes places. This refers to a diagram of views at particular interaction points with the process.

---

## **Completing solution design**

The solution-design phase is complete when all design documents are stored in the ClearCase repository and are available for Application Developers to refer to when they create artifacts for the solution.



---

## Chapter 4. Designs of the user-registration and purchase order (PO)-management sample templates

This chapter describes the designs of the user-registration and PO-management sample templates. It is based on the Business Integrator solution design process described in “Chapter 3. Solution design” on page 19.

---

### Overview

The user-registration and PO-management samples provide insight into solution creation. The design of each is described in this chapter. User registration is described first. The focus of the description is a specific subprocess, the registration approval process. PO management is described next with the focus of the description on the PO-approval subprocess.

---

### User-registration solution

The purpose of the user-registration solution is to register new users to participate in an existing e-marketplace in an automated manner.

#### Business-process model

The current business process for registering a new user is analyzed to reveal that an automated process for registering to use the e-marketplace can streamline the business process. The end-to-end process for registering a new user includes subprocesses that are assumed to be in place and functioning. For instance, it is assumed that the user-logon process includes the following two subprocesses:

- For existing users of the e-marketplace, a process to log on to perform work based on the user’s role.
- For new users, a process to register to use the e-marketplace (a one-time activity).

The end-to-end process for the user-registration solution from logging on to request registration to logging on as a user in an appropriate role is shown in Figure 4.

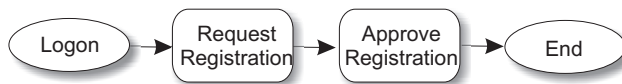


Figure 4. Process flow of the user-registration solution

The process has two subprocesses. In this example, a new user logon subprocess initiates the user-registration process.

The improved business-process model is as follows:

1. From a browser, a new user requests registration to the e-marketplace and participation with trading partners in the role of a buyer.
2. An administrator approves the new user in a guest role and provides the new user with a user ID and password.
3. The new user participates in the e-marketplace in the limited role of guest until he or she is added to the trading-partner services in the buyer role for which he or she registered.
4. The registered user logs on to use the e-marketplace and participates in the role for which he or she registered.

The user-registration solution focuses on the registration-approval process. This process is discussed in “Task model”.

The improved process eliminates many of the manual steps for the e-marketplace and trading-partner administrators, and involves the new user in the e-marketplace much more quickly in the guest role. Once the new user is approved in the role for which he or she registered, participation in the e-marketplace in that role is automatic at logon.

## **Task model**

The task model focuses on the approval process for the user-registration solution.

The MQSeries Workflow Buildtime tool is used to model the workflow for the approval process. The result is a Flow Definition Language (FDL) script that can be used by the MQSeries Workflow Runtime engine to act as a template for instances of the process flow.

The approval process has one start point and two end points. The process also has a subprocess for adding the registering user to services before the end points. The flow for the approval process is illustrated in Figure 5 on page 27.

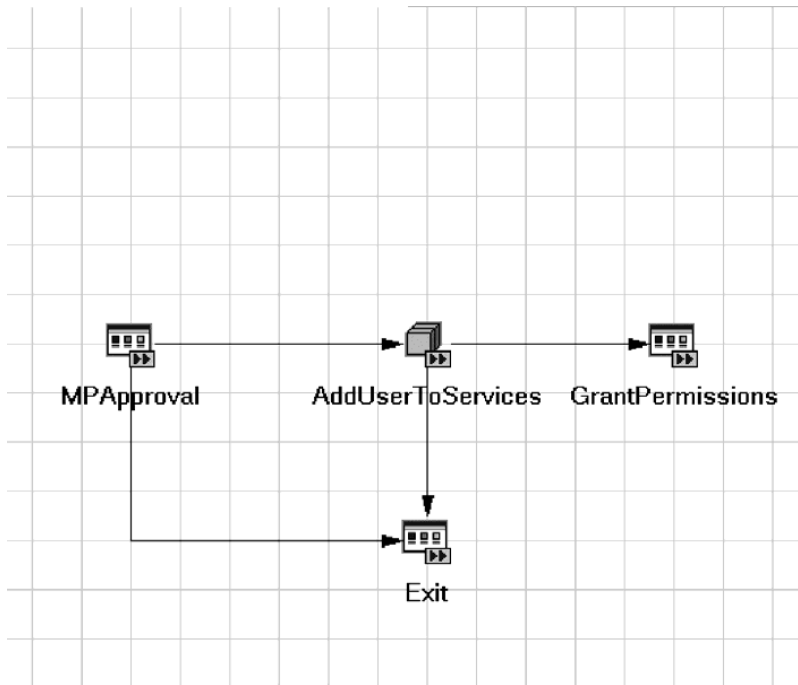


Figure 5. The approval process as modeled in MQSeries Workflow Buildtime

The subprocess has a single start point and a single end point. The end point of the subprocess leads to either of the two end points for the process flow. The subprocess is illustrated in Figure 6 on page 28.

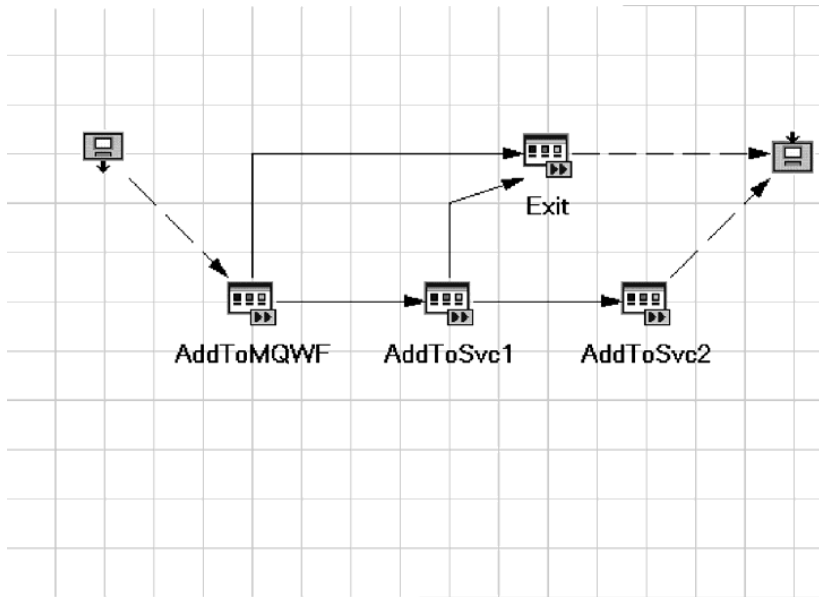


Figure 6. A subprocess as modeled in MQSeries Workflow Buildtime

### Approve registration

The approve-registration process has five tasks associated with it. Three are performed by humans and two are programmatic (that is, automatically performed by applications). For the user-registration solution, the administrator performs the first three tasks. The tasks are as follows:

1. When the administrator logs on to the e-marketplace, the Logon servlet authenticates the user ID and password through the Business Integrator Trust and Access Manager (TAM) and checks the authorization for the administrator based on the user's role. The Logon servlet then logs on to the Workflow Services session bean to determine whether there is any work for this user, based on his role; for instance, whether there are new users needing approval. The servlet presents the administrator with a screen that lists the following four options:
  - Approve/Reject new users
  - Register new user by proxy
  - Revoke registered user
  - Modify user data

Because a new user has requested to register with the e-marketplace, there is work for this administrator to perform. The servlet presents a screen with the Approve/Reject new user option displayed as an active link.

When the administrator selects this link, a request is sent to the UserReg servlet. The UserReg servlet requests a list of users with the status of Pending Approval from the Registration Services.

Registration Services returns the list of users, and the UserReg servlet provides this list as an HTML form with buttons such as **Approve** and **Reject** to the administrator. The administrator selects the new user and then selects the **Approve** button. This initiates a request to the UserReg servlet. The UserReg servlet sends an Approve User request to Registration Services. On the administrator's behalf, Registration Services initiates and completes the Workflow activity for approving the new user to the e-marketplace.

2. The administrator must add the user to the MQSeries Workflow Buildtime. This is a manual task in the approval subprocess. The administrator logs on to MQSeries Workflow and adds the user to the Buildtime. See the MQSeries Workflow documentation for details.
3. Next, the administrator confirms that the MQSeries Workflow Runtime is updated to reflect the new user. See the MQSeries Workflow documentation for details.
4. The new user is then added to the selected trading-partner services for which the user registered.
5. Finally, the user registration is approved and the new user can log on to the e-marketplace and perform work based on the user role.

### Information model

The data required by each task in the user-registration solution is listed in Table 7.

*Table 7. Information model*

Tasks	Data
New user request registration	User ID and password, process start data
Complete and submit form	New user name, address, phone, business name
Administrator logon	User ID and password
Select work item	(none)
Select new user	New user name, user ID and password, and role
Approve new user	(none)

### Organization model

The user-registration solution organization model is simple. There are two roles: new user and administrator. The new user's role in the e-marketplace

changes as part of the user registration process to one of the predefined roles for the e-marketplace dependent upon the privileges for which the new user is approved.

### Client model

The user-registration solution involves modifying and creating several different views that are presented to the various users, both by their role and by the tasks and subtasks they perform. The views for both the new user and the administrator change as the process progresses. The views change based on the tasks the users perform and the state of the process.

For the user-registration process, it is possible to identify the views listed in Table 8 to present in a user's browser based on his or her role.

*Table 8. Client model for user registration solution*

User role	Task	View	Type
New user	Request registration	Logon/Registration Welcome page	HTML form
	Submit registration	Registration form page	HTML form
		Successful submission acknowledgement	HTML page
		Unsuccessful submission	HTML form
		Registration status page	HTML page
	Log on in guest role	Logon/Registration Welcome page	HTML form
		Registration status page	HTML page
	Log on in approved role	Logon/Registration Welcome page	HTML form
		Tasks awaiting action based on role and user ID	HTML form
Administrator	Log on	Logon/Registration Welcome page	HTML form
	View pending work items	Tasks awaiting action based on role and user ID	HTML form
	Select work item	List of pending work items	HTML form
	Select new user and act on request	List of users requesting registration	HTML form

---

## PO-management solution

The purpose of the PO-management solution is to manage purchase orders for the enterprise in the e-marketplace.

### Business-process model

The current process for managing POs is analyzed to reveal that an improvement can be made to the PO-approval process. The overall PO-management business process is described in an overview. The PO-approval process is the focus of the design description and creation of artifacts for the PO-management solution.

### PO-management overview

The end-to-end process for managing POs is shown in Figure 7.

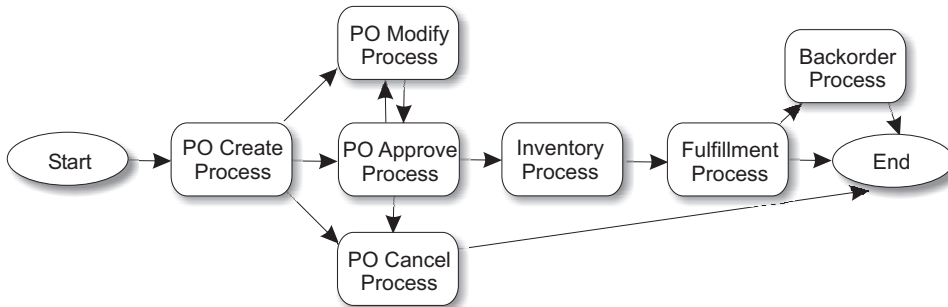


Figure 7. Process for managing purchase orders

The PO-management process has many subprocesses. In Figure 7, the process begins when a user of the e-marketplace logs on in a role that is authorized to participate in the PO-management process. For the solution, these roles can include the following:

- Buyer
- Administrator
- Financial approver

The logon servlet authenticates the user ID and password through TAM and checks the authorization for the user based on that role. The logon servlet presents the user with a screen to select the role the user wants to log on with. The user selects the buyer role, which sends a request to the POMgmt servlet that manages the user interface to the PO-management process for a list of actions that are available for the user to perform.

This servlet presents a screen with the following activities based on the buyer role:

- Create PO
- Modify PO
- Cancel PO

Simultaneously, the POMgmt servlet sends a request to the Workflow Services session bean to determine whether any work is pending for this buyer.

**Note:** The buyer role does not include approving POs. This action is reserved for administrators.

### **Create new PO**

If the buyer chooses to create a new PO, the POMgmt servlet presents the buyer with the Create Purchase Order HTML form page. The buyer fills in required trading-partner information, creates line items, and submits the PO by clicking **Submit PO** on the form page, which sends a create request to the POMgmt servlet.

The POMgmt servlet sends a create request to the PO Services session bean, which handles all requests for interaction with back-end persistent entity beans (for example, PO and POLineItem). PO Services creates an instance of the PO entity bean to create the PO as a persistent business data object (BDO). Because the request includes creation of PO line items, the PO entity bean sends a request to the POLineItem entity bean to create the line items as persistent BDOs.

The POLineItem bean returns the status of each line item to the PO bean, which returns an indication of whether the creation attempt was successful or unsuccessful to PO Services. PO Services returns the result to the POMgmt servlet, which returns the result to the buyer, updating the buyer's screen with the new PO. The PO Approval workflow process is started.

### **Modify PO**

If the buyer chooses to modify an existing PO, the POMgmt servlet requests a list of POs the buyer can modify from PO Services. PO Services returns the list to the servlet, which then presents the user with a list of POs that can be modified.

The buyer selects a PO to modify from the list; this sends a request to the servlet to display the PO with line items. The buyer selects the line item to modify, changes the quantity, and selects the Submit PO button on the HTML form. The HTML form is presented as a JSP file. Submitting the PO sends a modify request to the POMgmt servlet.



The POMgmt servlet sends a change request to PO Services to change the PO. PO Services delegates the modify PO request to the PO BDO to modify the PO. The current PO Approval workflow process is terminated. The PO BDO determines the POLineItem object or objects that need to be modified, then tells the POLineItem bean to make the changes. The POLineItem BDO then returns the status of the line items to the PO BDO, which returns a successful/unsuccessful indication to PO Services. PO Services returns the result to the POMgmt servlet, which changes the HTML page presented to the buyer. The PO Approval workflow process is restarted.

The buyer's view shows the modified PO. The buyer can then choose to modify other POs that are pending action or return to perform other activities that the buyer is authorized to perform.

### **Cancel PO**

If the buyer chooses to cancel a PO, the PO is selected from the list of POs the buyer's role has access to. This list is presented by the POMgmt servlet as described in "Modify PO" on page 32. The buyer selects the **Cancel PO** button on the HTML form, which sends a request to cancel the PO to the POMgmt servlet.

The POMgmt servlet sends this request to PO Services, which in turn initially delegates the cancel request to the PO BDO. The PO BDO tells the POLineItem BDO to delete the line items from the database. The PO BDO returns a successful/unsuccessful indication to PO Services, then removes the PO. These results are returned to the POMgmt servlet, which renders a revised HTML page.

### **Approve PO**

The PO-approval process is the process discussed for the PO management solution. See "PO-approval task model" on page 34 for details. The role that can perform this action is the administrator role. The financial-approver role can also perform this action if the total cost of the PO is greater than \$10,000.00. The buyer role cannot approve a purchase order in the PO-management solution.

The POLineItem bean checks the current inventory level of each line item by sending a request to the Inventory BDO entity bean for a quantity of each item. The Inventory bean checks inventory to determine whether enough inventory is available for each line item to satisfy the request and returns the status of each line item to the POLineItem bean.

When asked to make changes, the POLineItem BDO sends a request to the Inventory BDO to process the modification. The Inventory BDO determines whether or not it can process the request based on the current inventory, and returns the status to the POLineItem BDO.

## PO-approval task model

The PO-approval process is composed of several activities identified in the workflow. The process from logon to approval can involve the following steps:

1. An administrator enters the URL for the e-marketplace's welcome screen.
2. The Logon servlet renders the logon screen.
3. The administrator enters the appropriate user ID and password and is authenticated and authorized.
4. The Logon servlet prompts the user to select the role in which he or she wants to log on.
5. The user selects the administrator role.
6. The Logon servlet presents the administrator with the welcome screen for the administrator role.
7. The administrator selects Purchase Order Management and Review PO from the menu.
8. The request is sent to the POMgmt servlet.
9. The servlet logs on to the Workflow Services session bean to determine whether there is any work outstanding, such as POs that require approval. If outstanding work exists, a list of POs is returned to the servlet.
10. The servlet renders an HTML form that lists any POs that are awaiting action from an administrator.
11. The administrator selects the PO to approve from the list.
12. A request is sent to the POMgmt servlet for the particular PO (identified by a PO identifier in the BDO) that is forwarded to PO Services. This also initiates the preliminary validation subprocess in the PO Approve process.
13. PO Services forwards the PO data to the POMgmt servlet, which renders an HTML form of the PO with the options of approving or rejecting it and the option of canceling any action on the form and returning to the list of POs.
14. The administrator approves the PO, which sends a request to PO Services to complete the preliminary validation subprocess.
15. The next activity, Get PO Amount, is programmed to occur automatically.
16. An MQSeries message named User Process Execution Services (UPES) is sent to the Java Message Service (JMS) Listener's worker to call a method on the PO Services bean.
17. This method initiates the Test PO Amount activity, retrieves the cost amount of the PO, and passes this amount to the activity by completing it.

18. Logic in the Test PO Amount activity determines whether the amount is greater or less than \$10,000.00. If the amount of the PO is less than \$10,000.00, the PO Approve Success activity is initiated. If the amount is greater than \$10,000.00, the FA Approval activity is initiated; this places the PO in a state for the financial approver to review and manually act on by using the FA Approve form.
19. Assuming the amount is less than \$10,000.00, the PO Approve Success activity sends a UPES message that results in a method on PO Services to set the state of the PO to approved, and the PO-approval process completes.

### Information modeling

The data required by each task in the PO-management solution's approve-PO process is displayed in Table 9.

*Table 9. Information model*

Tasks	Data
Preliminarily validate	(none)
Get PO amount	PO amount
Test PO amount	(none)
PO approve	(none)
Approve success	(none)

### Organization modeling

The organization model in the PO-approval process is simple. A buyer within the enterprise creates and modifies purchase orders. An administrator approves or rejects each purchase order.

### Client modeling

The client model for the PO-approval process involves modifying and creating several different views that are presented to the administrator. The views vary as the process progresses and by the tasks the administrator performs.

The views that can be generated for the PO approver in the approve-PO process of the PO-management solution are shown in Table 10 on page 36.

Table 10. Client model for PO-management solution approve-PO process

User role	Task	View	Type
Administrator	Log on	Logon/Registration Welcome page	HTML form
	View outstanding work items	Tasks awaiting action based on role and user ID	HTML form
	Select work item	List of pending work items	HTML form
	Select review PO	List of POs requiring action	HTML form
	Select PO	PO selected for approval	
	Preliminarily validate PO	Automatic approval/rejection tasks	
Financial approver	Log on	Logon/Registration Welcome page	HTML form
	View outstanding work items	Tasks awaiting action based on role and user ID	HTML form
	Select work item	List of pending work items	HTML form

---

## Part 2. Creating artifacts for solutions

<b>Chapter 5. Creating solution artifacts</b> . . . . .	39	Overview . . . . .	57
Overview . . . . .	39	Artifact types . . . . .	57
Artifacts for the solution templates . . . . .	39	Supported tools . . . . .	58
Creating artifacts . . . . .	40	Creating the artifacts . . . . .	58
Registering tools for use . . . . .	41	MQSeries Integrator . . . . .	58
		DataInterchange. . . . .	59
<b>Chapter 6. Applications</b> . . . . .	43	Partner Agreement Manager (PAM) . . . . .	59
Overview . . . . .	43	Overview of creating and editing public processes . . . . .	60
Artifact types . . . . .	44	MQSeries Queue Definition . . . . .	61
Supported tool . . . . .	45	Java Message Service . . . . .	63
Business Integrator considerations . . . . .	45		
Creating the artifact . . . . .	46	<b>Chapter 11. Roles.</b> . . . . .	69
Artifact example . . . . .	46	Overview . . . . .	69
		Artifact types . . . . .	69
<b>Chapter 7. Business Process</b> . . . . .	47	Supported tool . . . . .	69
Overview . . . . .	47	Business Integrator considerations . . . . .	69
Artifact types . . . . .	47	Creating the artifacts . . . . .	69
Supported tools . . . . .	47	Artifact example . . . . .	71
Business Integrator considerations . . . . .	48	LDAP configuration file . . . . .	71
Creating the artifacts . . . . .	48	Policy Director commands file . . . . .	72
Creating work flows . . . . .	48	Policy Director map file . . . . .	72
Artifact examples . . . . .	49	Policy Director configuration file . . . . .	72
<b>Chapter 8. Clients</b> . . . . .	51	<b>Chapter 12. Servlet</b> . . . . .	73
Overview . . . . .	51	Overview . . . . .	73
Artifact types . . . . .	51	Artifact types . . . . .	73
Supported tools . . . . .	51	Supported tool . . . . .	73
Creating the artifacts . . . . .	51	Business Integrator considerations . . . . .	74
Artifact example . . . . .	53	Creating the artifact . . . . .	74
Browser user registration JSP file . . . . .	53	Creating the servlet artifact file. . . . .	74
		Artifact example . . . . .	75
<b>Chapter 9. Data</b> . . . . .	55		
Overview . . . . .	55	<b>Chapter 13. Resources and rules</b> <b>(Personalization)</b> . . . . .	77
Artifact types . . . . .	55	Overview . . . . .	77
Supported tool . . . . .	55	Artifact types . . . . .	78
Creating the artifact . . . . .	55	Supported tool . . . . .	78
Artifact examples . . . . .	56	Business Integrator considerations . . . . .	78
Registration Services . . . . .	56	Creating the artifacts . . . . .	78
Workflow Services . . . . .	56		
Registration BDO . . . . .	56		
<b>Chapter 10. Messages</b> . . . . .	57		

This part describes how to create solution artifacts.



---

## Chapter 5. Creating solution artifacts

This chapter provides an overview of creating artifacts for Business Integrator solutions. It provides information about the artifact types and the supported tools for creating the artifacts.

---

### Overview

You create solution artifacts in Solution Studio by using WebSphere Studio workbench, the installed and supported products and tools, and the Solution Studio wizards. For information about using Solution Studio, see “Chapter 2. Solution Studio” on page 9 and the online help. Unlike many of the artifacts that are created during the solution-design phase, which are not publishable or deployable, most of the artifacts you create during the solution-development phase are eventually deployed to a run-time production environment and used as part of an e-business process.

Each instance of a type of artifact is created by using a specific tool that is part of the integrated set of tools.

**Note:** The rest of the chapters in this part of the book detail different solution elements (for example, Applications or Business Process) and provide information on the supported tools for creating the associated artifacts.

Although other products or tools can create the same or similar type of artifact as those created by supported tools, these tools are not supported for Solution Studio solutions. Artifacts created by using these other tools can still be part of your solution. See “Chapter 14. Solution publishing” on page 81 for more information on how to incorporate the artifacts not created by using tools that are integrated with Solution Studio into your solution.

For instance, the Business Process solution element comprises two types of artifacts, the workflow and the microflow. You use a different tool to create each type of artifact. For the workflow (the flow of the business process), you use the MQSeries Workflow Buildtime tool. For the microflow, you use MQSeries Adapter Builder.

---

### Artifacts for the solution templates

This section identifies the types of artifacts that you can create for any solution that is developed by using the Solution Studio solution templates. The list of artifacts is based on the template identified in “Solution Studio templates” on page 15. Table 11 on page 40 shows the types of artifacts you

can create for any solution that can be deployed to an Enterprise configuration Business Integrator run-time environment.

*Table 11. Types of artifacts for solutions based on the folder of the Solution Studio template*

Folder	Artifact types
Applications	<ul style="list-style-type: none"> <li>• Adapters that connect the Business Integrator system with legacy applications by mapping inbound and outbound messages to Business Integrator-standard messaging. This is done by using the interchange schema defined by the message sets.</li> <li>• Microflows that act as interfaces between session and entity beans and a Java-enabled back end. Microflows can also act as state machines.</li> </ul>
Business Process	Workflows for the business process.
Client	Servlets that manage the interaction of the business process and the user view of the process. JSP files that provide the HTML renderings of information in the user view.
Data	Business data objects (BDOs).
Messages	Message configuration files, business object documents (BODs), electronic data interchange (EDI), message queues, Partner Agreement Manager public processes, message sets, Java Message Service (JMS) Listener configurations.
Resources	Java objects (enterprise beans, JavaBean™ components (beans), and JSP files) used by the WebSphere Personalization product.
Roles	Organization and roles information for configuring LDAP and Policy Director.
Rules	Java objects (enterprise beans, beans, and JSP files) used by the WebSphere Personalization product.
Servlets	Servlets.

---

## Creating artifacts

Use specific supported tools to create artifact files. Some artifact files are created by using Solution Studio wizards. Use wizards to start many of the products or tools that you use to create artifact files and to import and export files to and from tools and Solution Studio. Use wizards to create the configuration files needed by many of the artifacts for effective deployment to run-time test and production environments. The chapters that follow describe the use of the tools and wizards to create the artifacts based on their use to create solutions.



---

## Registering tools for use

The artifacts you create for your solution are created by using specific products or tools integrated into the Solution Studio. Before you can use a product or tool from Solution Studio, it must be registered with Solution Studio. To register the tool, use the Wizard Registration wizard.

In most cases, the products or tools that are used by Application Developers are already installed and registered by the Application Development Administrator. The person who installed the product or tool used the Wizard Registration wizard to configure the tool for use within Solution Studio.

Assuming that the product or tool has been installed but not registered, use the Wizard Registration wizard to register it. To register the product or tool, do the following:

1. With WebSphere Studio running, click **Tools** → **Wizards** → **Wizard Registration Wizard**. This starts the wizard, as shown in Figure 8.



Figure 8. The Wizard Registration wizard

2. Consult the online help or the *WebSphere Studio Business Integrator Extensions Installation Guide* book for procedures for registering the product or tool. The online help provides field help for the wizards.

If the product or tool is not installed, ask your Application Development Administrator whether the product or tool can be installed on the workstation or whether the product or tool is available on another workstation.

---

## Chapter 6. Applications

This chapter describes the creation of application adapters. Online help is available for the wizards that are used to create solution artifacts.

---

### Overview

*Application adapters* integrate best-of-breed legacy applications, such as SAP, into the Business Integrator platform by converting data and associated semantics to Business Integrator-standard messages. These messages are formatted in an application-neutral format known as extensible markup language (XML). This allows disparate applications to interact with one another through the exchange of XML messages.

Application adapters enable communications between applications (including business objects, native applications, and legacy applications) and the application adapter infrastructure. They perform the following tasks:

- Convert source data to an application-neutral format.
- Determine how to get data to the target.
- Direct or route the data through the Business Integrator infrastructure.
- Convert data from an application-neutral format into the format required by the target.

Two kinds of application adapters can be created:

**Push** The Business Integrator infrastructure is responsible for initiating and managing delivery of messages to the target application. This type of adapter typically does not require changing the target application to support the Business Integrator infrastructure.

**Pull** The target application is responsible for managing delivery of messages to it. This model requires changing the target application to support the Business Integrator infrastructure.

**Note:** To create a *push* application adapter, the application must have an external interface that enables the Business Integrator infrastructure to get data into and out of the application's environment.

Application adapters are built by using MQSeries Adapter Builder, which is launched from the MQSeries Adapter Builder wizard. For more information on application adapters, see the MQSeries Adapter Builder and MQSeries Adapter Kernel documentation.

## Artifact types

Three types of artifacts are created for the Applications element:

- Target adapters
- Source adapters
- Java Service adapters

**Note:** Java Service adapters are the microflows used to manage business-process flows. A Java Service adapter can provide a state machine for a process.

The tool enables you to create target and source adapters and Java Service adapters. After you have created the adapters by using the tool, the MQSeries Adapter Builder wizard compiles the adapter's source code for use in the solution. It also compresses the compiled adapters and moves these compressed files into the Applications folder in the project view.

The following table illustrates the relationship of MQSeries Adapter Builder to the MQSeries Adapter Builder wizard. The file-naming convention and the file types for the different adapters are provided in Table 12.

Table 12. Relationship between MQSeries Adapter Builder and the MQSeries Adapter Builder wizard

Tool	Adapter type			
	Target	Source	Java Service (non-EJB)	Java Service (EJB)
MQAB (source code)	<i>adapter.java</i> <i>adapter.c</i> <i>adapter.h</i> <i>adapter.def</i> <i>adapter.mak</i>	<i>adapter.c</i> <i>adapter.h</i> <i>adapter.def</i> <i>adapter.mak</i>	<i>adapter.java</i> <i>adapter_flow.xml</i> (XML configuration file)	<i>adapter.java</i> <i>adapter_flow.xml</i> (XML configuration file)

Table 12. Relationship between MQSeries Adapter Builder and the MQSeries Adapter Builder wizard (continued)

Tool	Adapter type			
	Target	Source	Java Service (non-EJB)	Java Service (EJB)
Wizard	Compiles source files to .class, .lib, .dll, and .exp files; compresses the compiled files into single file named a <i>adapter_TA.adapter</i> ; and imports the file into the Applications folder in the project view of WebSphere Studio.	Compiles source files to .lib, .dll, and .exp files; compresses the compiled files into single file named a <i>adapter_SA.adapter</i> ; and imports the file into the Applications folder in the project view of WebSphere Studio.	Compiles source file to a .class file; compresses the compiled file to a file named <i>adapter_JA.adapter</i> ; and imports the files into the Applications folder in the project view of WebSphere Studio.	Imports additional files for LDAP and WebSphere Application Server; produces the following files:  <i>adapter_CS.jar</i> <i>adapter_CS.xml</i> <i>adapter_Inst.XML</i> <i>adapter_JA.jar</i> <i>adapter_JetAce.xml</i>  and imports the files into the Applications folder in the project view of WebSphere Studio.

### Supported tool

The supported tool for creating the adapters is MQSeries Adapter Builder. This product provides an GUI for building an adapter for virtually any type of application. Although a visual interface hides much of the coding for building adapters, the output of the product is compiled source-code files that can be saved to the Applications folder. The procedures for building adapters are provided by an online help system. Information is also provided in the MQSeries Adapter Builder *Using the Control Center* book.

The Solution Studio MQSeries Adapter Builder wizard launches the tool and provides added functionality, such as compiling and compressing files.

The tool must be installed on the machine and registered for use by using the Wizard Registration wizard.

### Business Integrator considerations

For the adapter builder and the wizard to work together, store the adapter files you build for a project in a specific directory that you can then identify to the wizard. This enables you to locate, browse, and select the adapters directory quickly when you are ready to compile, compress, and store them in the Applications folder of the solution.

Source and target adapters are sourced in C. Target adapters and Java Service adapters are compiled to Java class files by the wizard. In addition to the class

file for the target adapter, the wizard compiles source-code files to .lib, .dll, and .exp files. These are the compiled file types for the source adapters.

A source adapter is needed for the application that is sending messages, and a target adapter is needed for the application that is receiving messages. Both are built by using MQSeries Adapter Builder. Applications and adapters are configured to work together by using the Solution Studio Application Configuration wizard.

---

## Creating the artifact

The adapter artifact is created by using MQSeries Adapter Builder. Refer to the online help available with the product for procedures for building adapters. Start MQSeries Adapter Builder from the MQSeries Adapter Builder wizard. To start the wizard, do the following:

1. From WebSphere Studio, select **Tools** → **Wizards** → **MQSeries Adapter Builder Wizard**.
2. From the wizard, check the box next to the type of adapter to build. Types include source adapters, target adapters, and Java Service adapters.
3. Click the **Start MQSeries Adapter Builder** button to start MQSeries Adapter Builder.
4. Use the tool to build your adapter.
5. When you finish building the adapter, you are ready to compile the source code into a class file, compress it, and move both the compressed file and the XML configuration file created by the adapter builder to the Applications folder in the WebSphere Studio project view.
6. The adapters can then be published as part of your solution. The target and source adapters are published to the applications that they service. The Java Service adapters that form microflows are published to the Business Flow Manager (BFM) server.

---

## Artifact example

The output of the MQSeries Adapter Builder and the wizard for a Java Service adapter microflow is a Java class file and an XML configuration file.

---

## Chapter 7. Business Process

This chapter describes the creation of artifacts for the Business Process element.

---

### Overview

Business-process elements include both workflows and microflows. These implement the business processes at the subprocess and task levels.

#### Artifact types

The artifact types for the Business Process element are the workflow (or business-process flow) and the microflow.

The workflow provides the overall flow of the business process. This artifact can be created as part of the design phase by using the MQSeries Workflow Buildtime tool. Although created as part of the design, this artifact is deployed to the run-time environment, where it provides a template for many instances of this business process.

The microflow models the flow of data between various components and Java objects such as session beans and entity beans.

#### Supported tools

You use different tools to create each type of artifact. The supported tool for creating workflows is the MQSeries Workflow Buildtime tool. MQSeries Workflow provides a GUI for building workflows by dragging and dropping elements from a palette of activities to the work space and connecting them to show the relationships among activities. Online help provides information on using the tool and procedures for creating the flows at the task level.

The MQSeries Workflow Buildtime tool is used to create Flow Definition Language (FDL) files that are used by the MQSeries Workflow Runtime engine to implement instances of processes and tasks.

MQSeries Adapter Builder is used to create microflows that are used by MQSeries Adapter Kernel and WebSphere Application Server to route messages between applications. MQSeries Adapter Builder provides a GUI for building microflows. The Solution Studio MQSeries Adapter Builder wizard launches the tool and provides added functionality, such as compiling and compressing source-code files. The tool must be installed on the machine and registered for use by using the Wizard Registration wizard.

## Business Integrator considerations

This section details special considerations for creating artifacts for the Business Integrator run-time environment.

The FDL files that you create during solution design or development are deployed to the run time for use by the MQSeries Workflow Runtime engine.

The microflows that you create during solution design or development are deployed to the run time for use by MQSeries Adapter Kernel and WebSphere Application Server.

---

## Creating the artifacts

To create the artifacts, you use different Solution Studio wizards and integrated tools.

### Creating work flows

To create a business process or workflow, do the following:

1. From WebSphere Studio, with a subfolder such as Business Process in the project view highlighted, select **Tools** → **Wizards** → **Workflow Wizard**. This launches the Workflow wizard, which is shown in Figure 9.

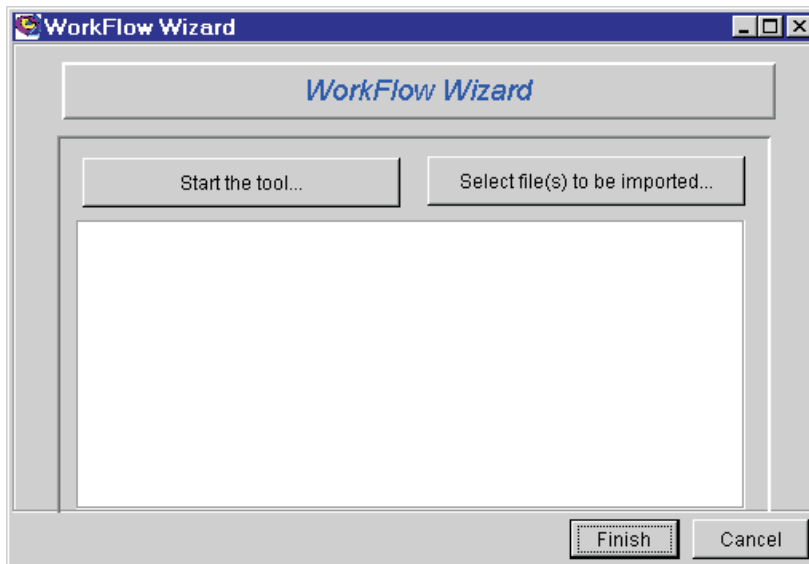


Figure 9. Workflow wizard

From the wizard you can start the tool and select files to import.

2. Select **Start the Tool** to start the MQSeries Workflow Buildtime tool.



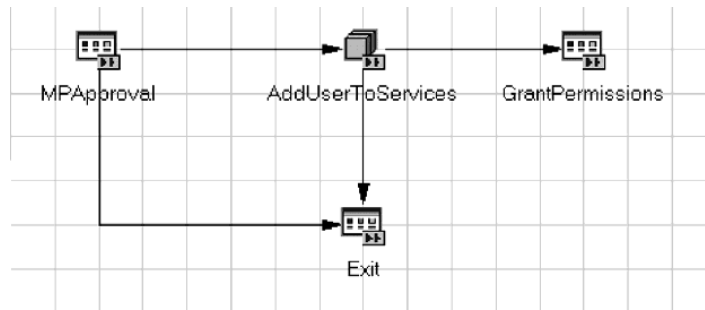
Refer to the MQSeries Workflow documentation for procedures for creating workflows. The artifact that you create is an FDL file that is imported to Solution Studio.

---

## Artifact examples

The following is an example of the user-registration sample template's administrator-approval process in the MQSeries Workflow Buildtime tool by using the GUI to define the process:

Figure 10. Buildtime GUI for creating FDL files



For information on microflows, refer to the MQSeries Adapter Builder documentation.



---

## Chapter 8. Clients

This chapter describes the creation of artifacts for the Clients element.

---

### Overview

The artifacts for the following two types of clients are created for the client element of a solution:

#### **Browser**

Provides the external interfaces or views to the Business Integrator run time that enable users to interact with business processes through a standard Web browser. It provides access to business-process content and activities based on the user's role.

**Client** Provides the internal interfaces to the business processes through enterprise beans.

### Artifact types

The artifact types for the browser are JSP files and HTML pages. Those created for the Client are XML files and enterprise beans (Java archive (JAR) files).

### Supported tools

The supported tool for creating JSP files and HTML pages is WebSphere Studio Page Designer. The supported tool for creating WebSphere Application Server XML files is the WebSphere Artifact Configuration wizard. The supported tool for creating enterprise beans is VisualAge for Java.

---

### Creating the artifacts

To create the artifacts for the browser, use the WebSphere Studio Page Designer tool. This tool enables you to create HTML and JSP files. Refer to the WebSphere Studio documentation for information on using the page designer.

Enterprise beans are distributed as JAR files. These artifacts require configuration files that you create by using the WebSphere Artifact Configuration wizard. The wizard enables you to create configuration files for several artifacts during a single use by clicking the **Save Configuration** button after you create each file. To create the configuration file, do the following:

1. From WebSphere Studio, select the JAR file for which you are creating the configuration file from the Client folder in the project view.

2. Select **Tools** → **Wizards** → **WebSphere Artifact Configuration Wizard** to start the wizard, which is shown Figure 11.

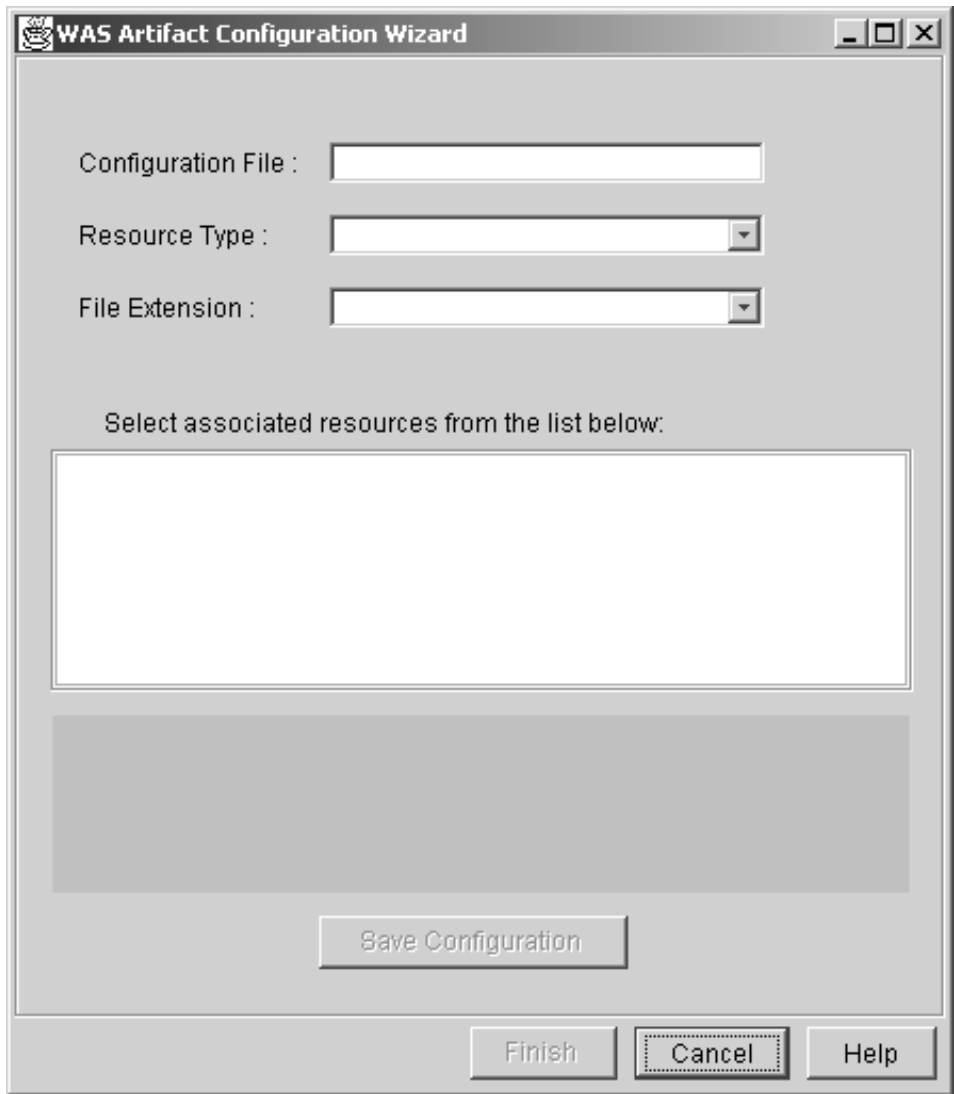


Figure 11. The WebSphere Artifact Configuration wizard

3. From the **Resource Type** menu, select **Enterprise Bean** to get a list of EJB artifacts from which to select.
4. Check the box corresponding to the artifact for which you are creating the configuration file.

5. In the **Container** field, enter the name of the container to which the enterprise bean is to be deployed. In the **Home** field, enter the name of the enterprise bean's home interface.
6. To create more configuration files for other artifacts, click the **Save Configuration** button after creating each file. Create and save other configuration files for other artifacts.
7. When you are finished creating files, click the **Finish** button. The wizard creates an XML configuration file for each artifact you saved a configuration for, then stores that XML file in the Client folder in the project view.

Online help is available for using this wizard. Field help is also available.

---

## Artifact example

The following are examples of a JSP file for the user-registration sample template.

### Browser user registration JSP file

The following is an example of a JSP file created for the user-registration sample template by using WebSphere Studio Page Designer. Two views are presented. Figure 12 on page 54 shows the page as it is created in Page Designer; Figure 13 on page 54 shows the page as it is previewed in a browser.

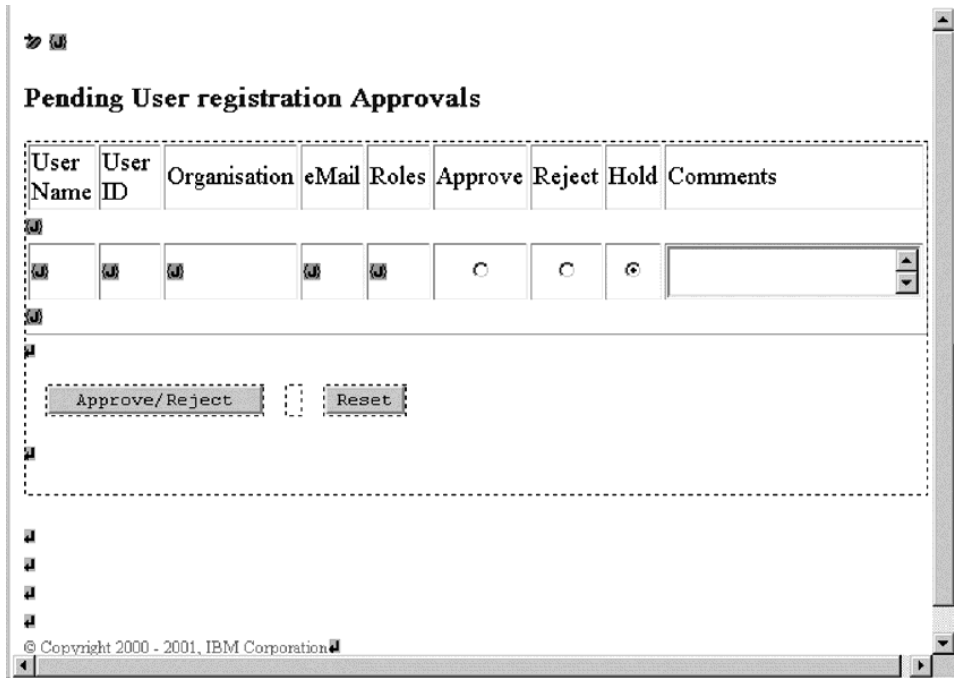


Figure 12. Sample JSP file as shown in WebSphere Studio Page Designer



Figure 13. Sample JSP file as resolved on a browser

---

## Chapter 9. Data

This chapter describes the creation of server-side enterprise beans and other business objects.

---

### Overview

The Data element for your solution comprises server-side enterprise beans. These business objects provide the data necessary to populate messages.

### Artifact types

The artifacts are server-side enterprise beans.

### Supported tool

The supported tool for creating any type of enterprise bean in Solution Studio is VisualAge for Java. The tool must be installed on the machine and configured to operate with WebSphere Studio. If it is not installed, you must install it and then configure it to operate with WebSphere Studio. For information on configuring WebSphere Studio and VisualAge for Java to work together, refer to the *WebSphere Studio Business Integrator Extensions Installation Guide*, or refer to the configuration information in the WebSphere Studio and VisualAge for Java documentation.

---

### Creating the artifact

You create enterprise beans by using VisualAge for Java. Refer to the VisualAge for Java documentation for information about creating enterprise beans.

For the User Registration solution, use VisualAge for Java to create three enterprise beans as follows:

- Two session beans:
  - Registration Services
  - Workflow Services
- One entity bean:
  - Registration Business Data Object (BDO).

The function of these beans is to take requests from servlets, manage the workflow, interact with back-end applications, and create persistent BDOs.

After you create the enterprise beans, store them in the Data subdirectory of the project. Because VisualAge for Java does not support versioning of

enterprise beans with an external versioning control system such as ClearCase LT, you must store them in a different form. Use VisualAge for Java to create deployable JAR files containing the enterprise beans, then import the JAR files into Solution Studio. When the JAR files are imported into Solution Studio, the artifacts are placed under ClearCase LT version control in the ClearCase LT repository.

---

## Artifact examples

This section discusses artifacts that were created for the user-registration sample template.

### Registration Services

The Registration Services is a session bean. When a new user makes a registration request, the UserReg servlet requests that the Registration Services session bean create a persistent BDO. The persistent BDO is maintained in the Trusted Zone until the new user registration is complete. The new user's request for registration is either approved or rejected by the administrator.

### Workflow Services

Workflow Services is a session bean that manages the tasks in the user-registration sample template.

### Registration BDO

The Registration BDO is an entity bean created by the Registration Services session bean for the new user. For each new user who makes a registration request, an instance of the Registration BDO is created by Registration Services. This instance persists until the user registration is completed by either approval or rejection.



---

## Chapter 10. Messages

This chapter describes the creation of configuration and other artifact types for the various types of messages that can be part of the solution.

---

### Overview

The Messages element for your solution comprises configuration and other types of files, including the following:

- MQSeries Integrator
- Electronic Data Interchange (EDI)
- Partner Agreement Manager (PAM)
- MQSeries message queues
- Java Message Service (JMS) message queues

Different wizards and tools are used for creating the artifacts, for launching the tools, and for storing the artifacts in the solution element before publishing them with the solution.

**Note:** The Messages folder is not automatically created by Solution Studio. You must manually create this folder in your solution by using WebSphere Studio.

### Artifact types

Various types of artifacts can be created for the Messages element of your solution. This element includes artifacts of the configuration file type, but also includes the creation and configuration of message queues. Table 13 shows the relationship between the type of message, the Business Integrator artifact, and the tool used to create the artifact.

*Table 13. Artifact type and tool for message element*

Messages element subfolder	Artifact type	Tools
MQSI	An MQSeries Integrator message set. <b>Note:</b> The wizard does not create a configuration file for the message set. See "MQSeries Integrator" on page 58 for information on configuring MQSeries Integrator in the run-time environment.	MQSI wizard and MQSeries Integrator

Table 13. Artifact type and tool for message element (continued)

Messages element subfolder	Artifact type	Tools
EDI	Export/Import files (.eif), Dictionary files (.dic), and PERFORM command files (.dicmd).	DataInterchange
PAM	Extensible markup language (.pxml).	PAM Public Process Editor
MQSeries	MQSeries queues (.mqsc) and queue configuration files (.xml).	MQSeries Queue Definition wizard
JMS	Java Message Service configuration file (.xml).	Application Configuration wizard

## Supported tools

The supported tools for creating artifacts for the Messages element are listed in the third column (**Tools**) of Table 13 on page 57.

---

## Creating the artifacts

Different wizards and products or tools are used to create the various artifacts for the Messages element of the solution. During the solution-design phase, you can find that your solution requires a variety of message types to integrate the business process with public processes and back-end applications through the Business Integrator Information Delivery Manager (IDM).

---

## MQSeries Integrator

This section describes creating MQSeries Integrator artifacts for a solution by using the MQSI wizard. The MQSI wizard enables you to launch the MQSeries Integrator tool, with which you create message sets. It also enables you to import the created message set into Solution Studio, where it can be part of your solution. Unlike some other artifacts, the message set cannot be configured for automatic deployment in the run-time environment.

The message set can be published and deployed to the run-time environment as part of solution publishing and deployment; however, after it is deployed to the MQSeries Integrator machine, it must be manually imported to the MQSeries Integrator run time. You must use MQSeries Integrator to perform the manual import.

Refer to the MQSeries Integrator documentation for information about creating message sets and for procedures for manually importing them into the run time. Online help is also available for creating message sets and for importing them into the run time.

---

## DataInterchange

The DataInterchange product acts as Business Integrator's Electronic Data Interchange gateway. Its function is to reformat data for electronic transmission. DataInterchange is installed, configured, deployed, and operated by itself, although it can interoperate with the rest of Business Integrator. The artifacts are created by using the DataInterchange Client and the DTD Conversion Utility on any machine on which you have DataInterchange licenses; this machine does not have to have Business Integrator or Solution Studio installed. DataInterchange artifacts are deployed to, undeployed from, and redeployed to Business Integrator machines by using DataInterchange's native facilities. DataInterchange does not use any Solution Studio facilities for audit, exception, or trace; you cannot manage DataInterchange facilities through the Business Integrator consoles. Refer to the *DataInterchange Client User's Guide* for details on using DataInterchange to create the artifacts and store them in Solution Studio for use in your solution.

---

## Partner Agreement Manager (PAM)

This section describes the creation of PAM artifacts.

Solutions can include PAM public processes. These processes, which are sometimes called intra-enterprise processes, control the flow of work between two partners outside the Business Integrator private processes. The processes are created by using the PAM Public Process Editor. The editor is launched from Solution Studio whenever a public process is required. The artifact is then stored in the repository, making it available for the solution.

A wide variety of public-process connectors are available in the PAM Public Process Editor, enabling a large number of combinations and interactions. However, only the most useful and common connection scenarios are supported at present. Generating a private process beyond the scope of the following scenarios can produce unpredictable results. The following ways of connecting public processes are supported:

- SIMPLE—A process generates a single business object, which is then passed on serially to the next public-process step.
- BRANCH—A public process in which one or more of a number of business objects are produced.
- LOOP—A business object is routed as an input business object to a previously defined public-process step.

Complicated flows can be set up with nested branching, looping, and simple steps. Branching and looping sections of public processes can follow serially from one another. Only the following connection scenarios are supported:

- Any number of simple steps with terminal and initiating process nodes.
- Branching with any number of simple steps inside or outside the branch.

- A branch in which one of two business objects, determined by the logical XOR operator, is produced.
- Looping around any number of simple steps.
- Branching and looping following each other serially when separated by at least simple step.

**Note:** You can define both public and private processes with the Public Process Editor. Both of these are encapsulated within the same PXML file. For Business Integrator, define public processes and not private ones; the private processes are autogenerated when the solution is deployed. However, if you define your own private processes, your private processes are deployed without autogeneration.

## Overview of creating and editing public processes

For complete information on public processes, including creating or editing them, refer to the Extricity publications. The general procedures for creating a public process are as follows:

1. From Solution Studio, click **New** → **Public Process Template**. Give the new file the name of the public process you are creating.
2. Edit this new file by double-clicking it in the Business Process folder in the project view.
3. Enter the user ID and password for the PAM machine that Solution Studio is configured to use for process authoring. If the file is empty, the launcher assumes that the file is a new process to be created and uses the name of the file as an editable default process name.
4. Successful logon launches the Public Process Editor. This is known as a session.
5. Create or edit the public process.
6. When you have completed creating or editing the public process, save it to the PAM database and close the editor. Closing the editor causes the artifact to be exported from PAM and imported to Solution Studio, where it is stored in the repository.

Editing an existing public process is similar to creating a new public process. The general procedure is as follows:

1. From within Solution Studio, double-click the file to edit in the Messages folder of the project view.
2. Enter the user ID and password for the PAM machine that Solution Studio is configured to use for process authoring. If the file contains valid XML, the public process name is retrieved and displayed and imported into PAM from storage in Solution Studio.
3. Successful logon launches the Public Process Editor. This is known as a session.

4. Edit the public process.
5. When you have completed creating or editing the Public Process, save it to the PAM database and close the editor. Closing the editor causes the artifact to be exported from PAM and imported to Solution Studio, where it is stored in the repository.

---

## MQSeries Queue Definition

The Solution Studio MQSeries Queue Definition wizard is the supported tool for defining and configuring queues for your solution. If you create queues outside of Solution Studio, you must manually configure the queues during the deployment of your solution in the run-time production environment. This wizard, like other Solution Studio wizards, automates the process of creating and configuring, thereby speeding the process of creating and deploying solutions.

To define and configure queues, do the following:

1. With the Messages folder in the project or development view of WebSphere Studio highlighted, click **Tools** → **Wizards** → **MQSeries Queue Definition Wizard**.

The MQSeries Queue Definition wizard is displayed, as shown in Figure 14 on page 62.

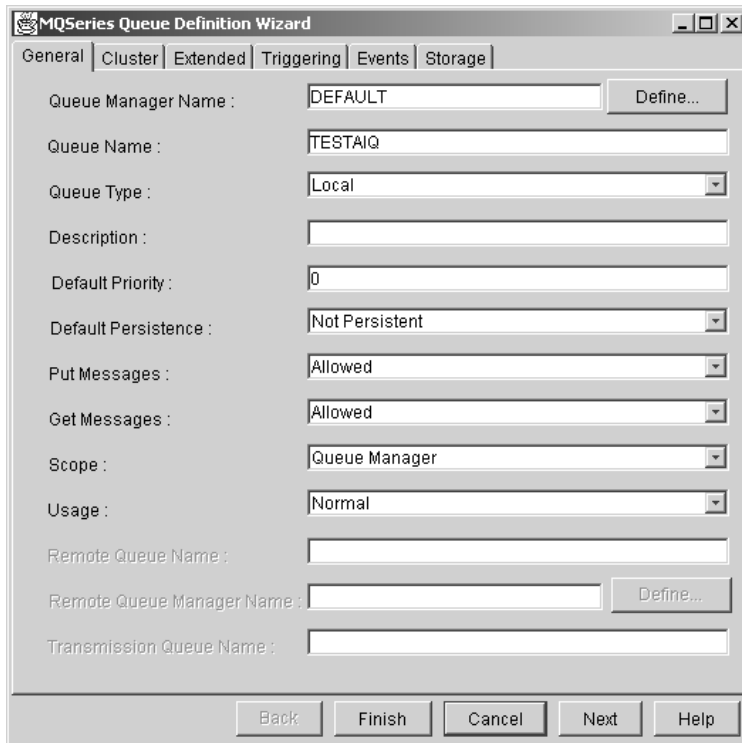


Figure 14. MQSeries Queue Definition wizard

2. Begin defining the queue by providing a queue name. You can specify other options on this page or accept the defaults. You can define two types of queues, local and remote. Click other tabs on the wizard to define various options for the queues:

- Cluster
- Extended
- Triggering
- Events
- Storage

The last four options in the list are specific to local queues only. See the MQSeries documentation for additional information.

3. When you have finished defining your queue, click **OK**. The wizard creates the queue definition in the form of a queue definition (.mqsc) file and stores the file in the Messages folder. The wizard also creates a configuration file that contains data used in creating the queue when the solution is deployed to the run-time environment.

Use the wizard to define all the queues that your solution requires. The wizard automates the process and prevents you from having to manually define and configure queues during deployment of your solution.

Online help is available for using the wizard and for fields on the various tabs.

---

## Java Message Service

This section describes the Java Message Service (JMS) artifacts you can create for your solution by using the Application Configuration wizard. JMS artifacts are supported in the run time by MQSeries Adapter Kernel, WebSphere Application Server, and JMS Listener. The JMS Listener performs routing of messages according to message types. It sends each message to the particular worker-message bean that handles that message's type. Upon receiving the message, the worker-message bean calls the appropriate component and delivers the message. See the *WebSphere Business Integrator Run Time* book and the MQSeries Adapter Kernel documentation for more information.

To create a JMS artifact, perform the following steps:

1. With the Messages folder in the project or development view of WebSphere Studio highlighted, click **Tools** → **Wizards** → **Application Configuration Wizard**. The Application Configuration wizard is displayed, as shown in Figure 15 on page 64.

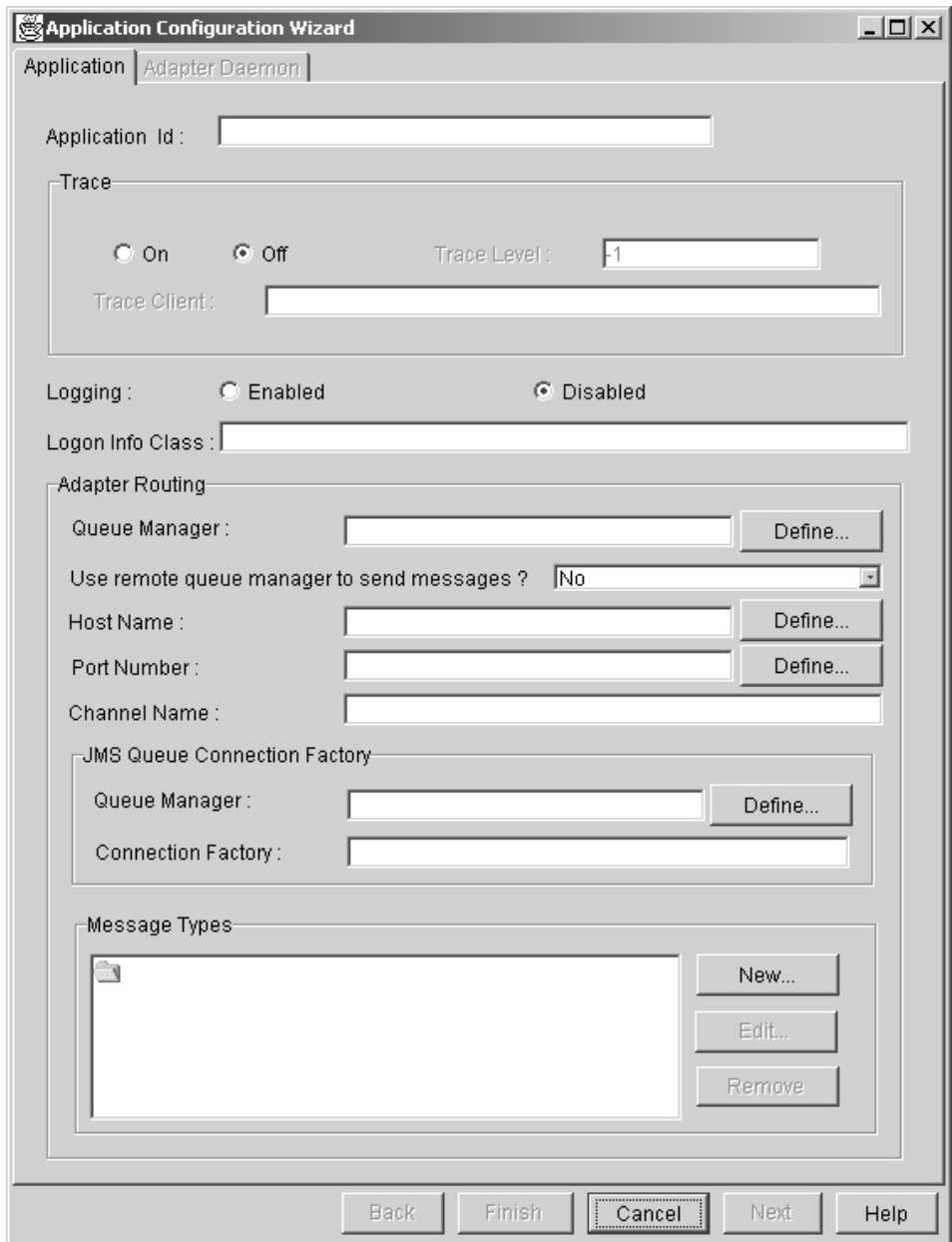


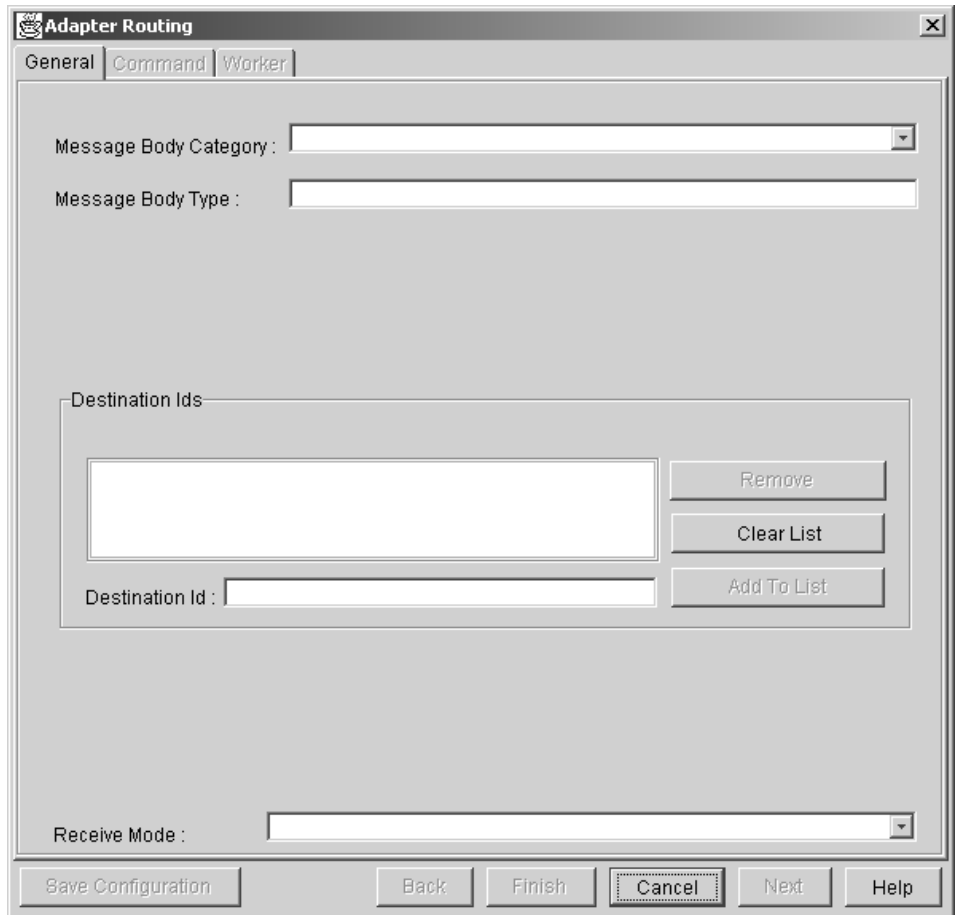
Figure 15. The Application Configuration wizard

Ensure that you are working on the **Application** page of the Application Configuration wizard; if not, click the **Application** tab.

2. Type the name of the application for which JMS is to be configured in the **Application Id** field.



3. Type the name of the JMS queue manager and JMS queue connection factory in the appropriate fields under **JMS Queue Connection Factory**.
4. Optionally, provide trace, logging, and logon information as appropriate in the wizard's other fields.
5. Select the message type you want to configure from the **Message Types** field and click **Edit**, or click **New** to create a new message type. This launches the Adapter Routing wizard, as shown in Figure 16.



The screenshot shows the 'Adapter Routing' wizard window with the 'General' tab selected. The window has a title bar with a close button. Below the title bar are three tabs: 'General', 'Command', and 'Worker'. The 'General' tab contains the following fields and controls:

- 'Message Body Category': A dropdown menu.
- 'Message Body Type': A text input field.
- 'Destination Ids': A list box containing no items, with three buttons to its right: 'Remove', 'Clear List', and 'Add To List'.
- 'Destination Id': A text input field.
- 'Receive Mode': A dropdown menu.

At the bottom of the window are five buttons: 'Save Configuration', 'Back', 'Finish', 'Cancel', and 'Help'. The 'Cancel' button is highlighted with a dashed border.

Figure 16. The Adapter Routing wizard

6. On the **General** page of the Adapter Routing wizard, enter the message's body category in the **Message Body Category** field and the message's body type in the **Message Body Type** field. Example values of default for these fields are shown in Figure 17 on page 66.

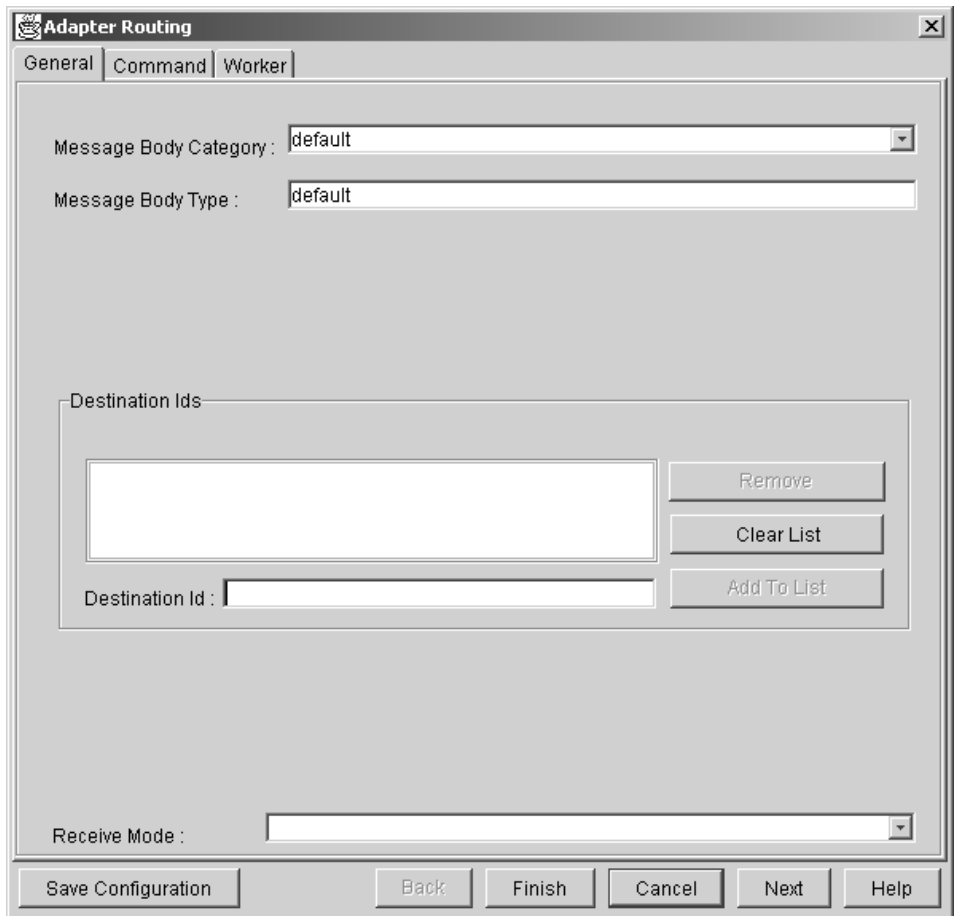


Figure 17. The **General** page of the Adapter Routing wizard

7. As necessary, provide information in the **Command** and **Worker** pages of the Adapter Router wizard, as shown in Figure 18 on page 67 and Figure 19 on page 68, respectively. See the online help for details.

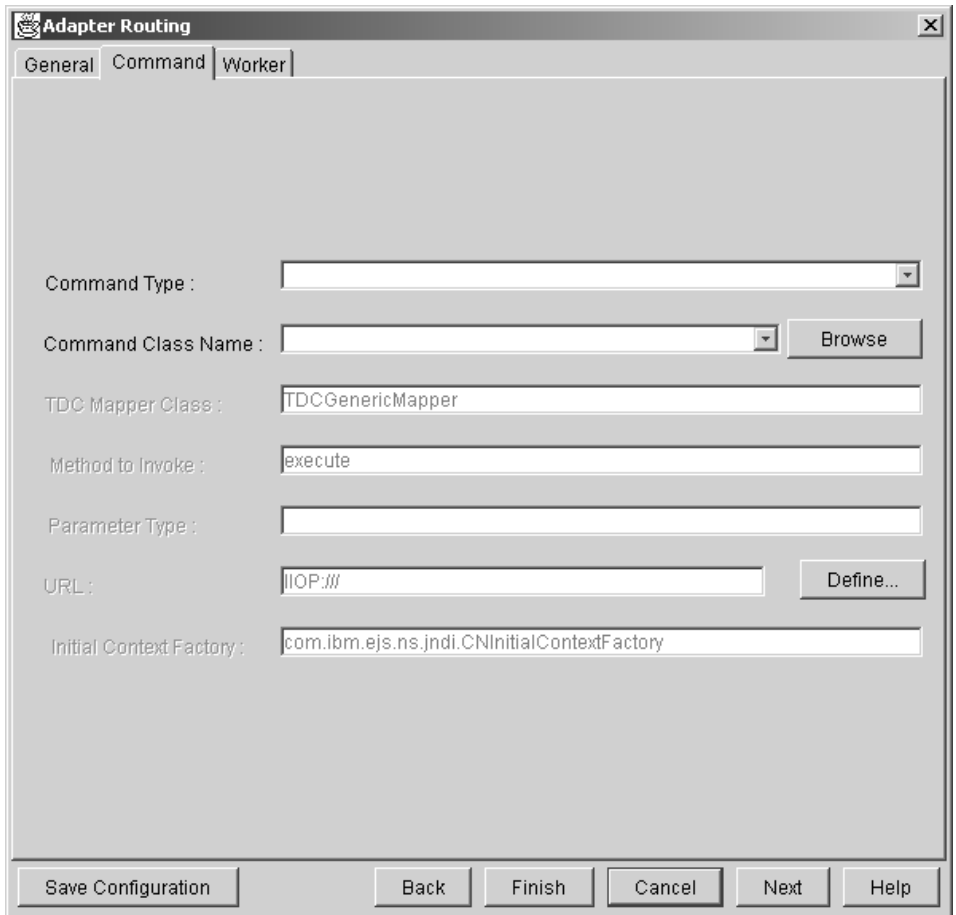


Figure 18. The **Command** page of the Adapter Routing wizard

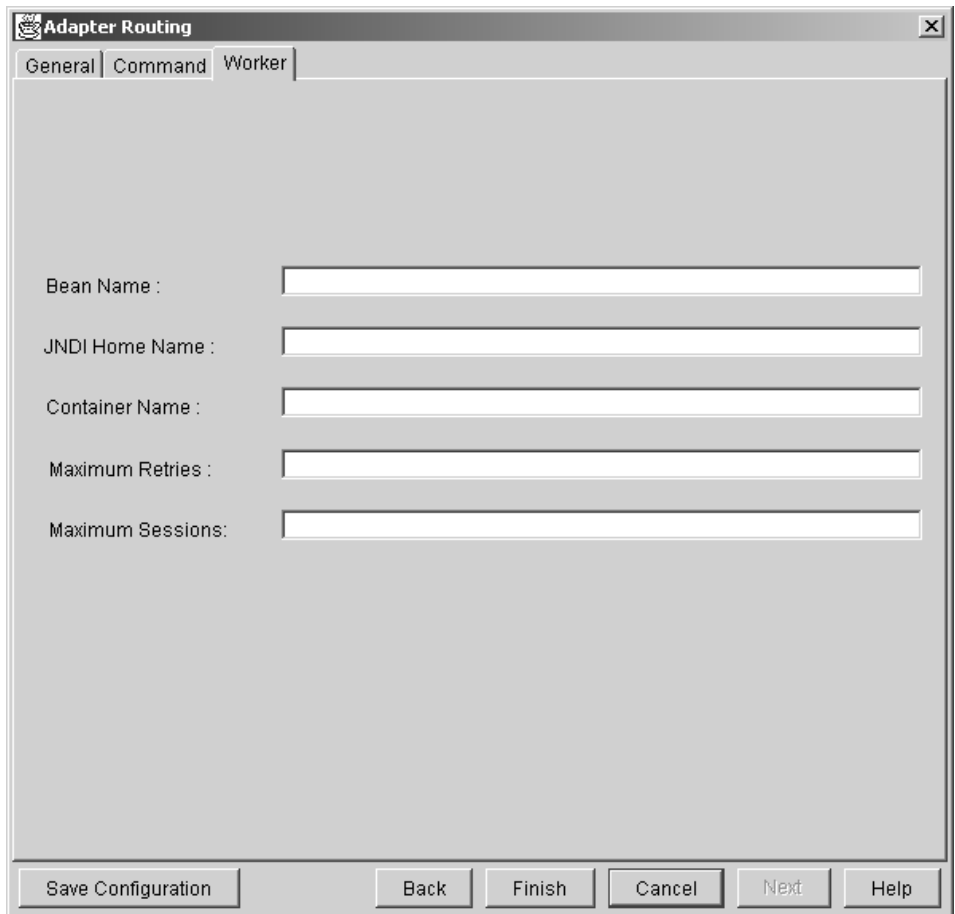


Figure 19. The **Worker** page of the Adapter Routing wizard

8. Click **Finish**. The wizard saves the configuration file in the Messages folder of the project view.

See the online help for complete information about using these wizards to configure JMS artifacts.

---

## Chapter 11. Roles

This chapter describes the creation of the Lightweight Directory Access Protocol (LDAP) configuration and Policy Director files for solutions.

---

### Overview

Roles are the means by which authenticated users are authorized to access solutions or Business Integrator private processes. For the solutions you create, the types of tasks a user performs and the views that a user is presented with are role based. Users' roles determine what activities they perform for a running solution. The task or activity that users can perform at any given time is determined by the users' roles.

LDAP and Policy Director are part of the Trust and Access Manager (TAM) services in the Business Integrator run-time environment. TAM provides centralized security, authentication, authorization, and nonrepudiation. TAM uses DB2 Universal Database™ for storage of configuration data.

### Artifact types

For Business Integrator Enterprise configuration, the following four kinds of artifacts are created for user roles:

- LDAP configuration files
- Policy Director command files
- Policy Director map files
- Policy Director configuration files

### Supported tool

The Solution Studio Solution Deployment wizard is the supported tool for configuring LDAP and Policy Director files for the solution.

### Business Integrator considerations

The Solution Deployment wizard is the last tool you use before publishing your solution. You use it to create the LDAP configuration and Policy Director files that modify the run-time directories after you have created the other artifacts for the solution.

---

### Creating the artifacts

To create the artifacts, do the following:

1. From WebSphere Studio, launch the Solution Deployment wizard, which is shown in Figure 20.

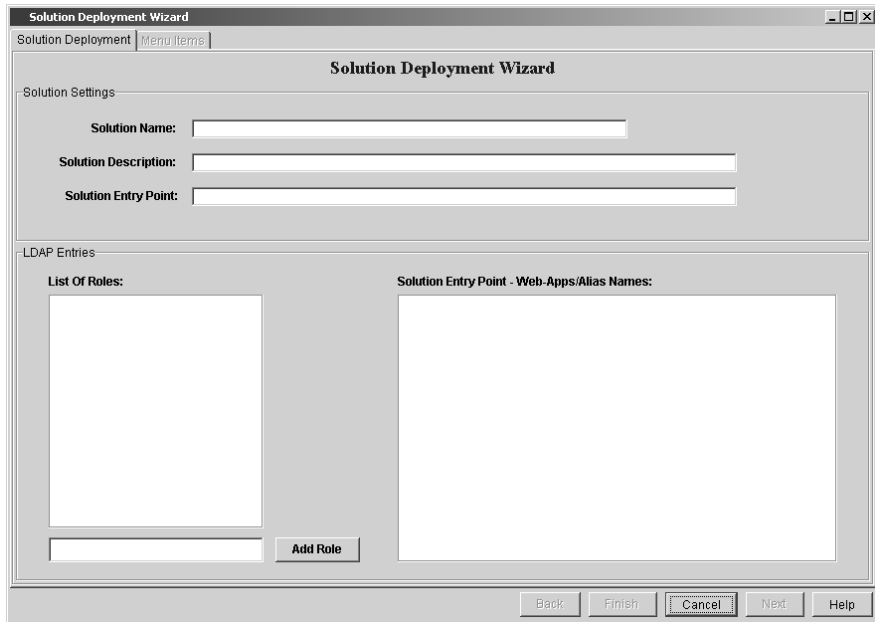


Figure 20. Solution Deployment wizard

2. Give the solution a name. It is recommended that this be the same as the project name.
3. Provide a description of the solution in the Solution Description field.
4. Identify the entry servlet for the solution. There is only one entry servlet for each solution. You can browse to locate it or you can enter the name.
5. Add new roles and highlight all the roles to be used in the solution from the list.
6. Click **Build Files** to build the LDAP and Policy Director configuration files for the solution.
7. Click **Finish** to complete building the files and to store them in the Roles folder, which is shown in Figure 21 on page 71.

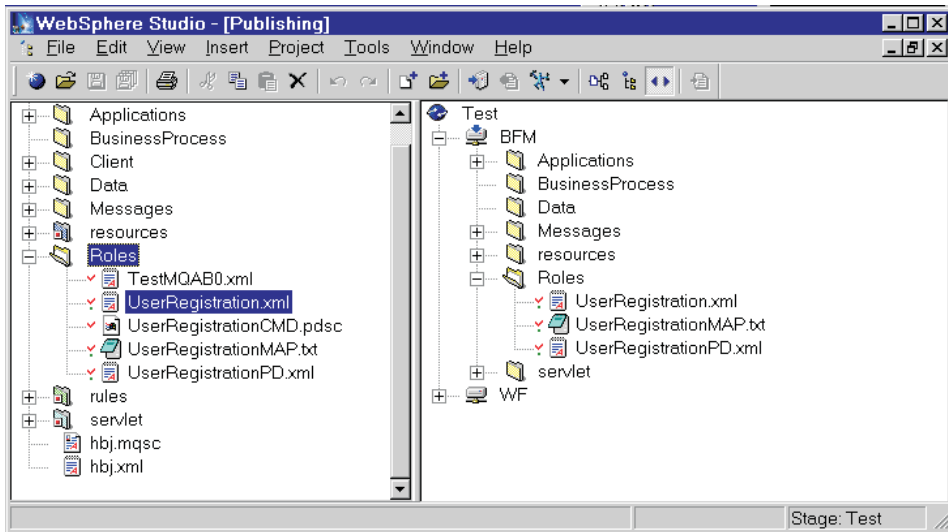


Figure 21. Roles folder containing wizard-built and stored configuration files

The files are used when you create the deployable solution package by using the Deployment wizard. Online help is available for the wizards.

## Artifact example

This section contains examples of artifacts that were created for the user-registration sample template, including the following:

- LDAP configuration file (in this case, UserRegistration.xml)
- Policy Director files:
  - UserRegistrationCMD.pdsc
  - UserRegistrationMAP.txt
  - UserRegistrationPD.xml (the Policy Director configuration file).

### LDAP configuration file

The following is an example of an LDAP configuration file for the user-registration solution:

```
<?xml version="1.0"?>
<Epic o="ePIC">
<ePICSolutions o="ePICSolutions">
<ePICSolution cn="UserRegistration">
<description>MP administrator authorization of new user</description>
<epicpdobjectspace>/UserRegistration/BtoBi</epicpdobjectspace>
<epicsolutionentry>com.ibm.bpip.studio.wizards.ExtractSubComponentFiles
</epicsolutionentry>
<epiclistofroles>
<value>Vendor</value>
```

```

<value>Buyer</value>
<value>EPIC_ASSIGN_TO</value>
<value>Supplier</value>
</epiclistofroles>
</ePICSolution>
</ePICSolutions>
</Epic>

```

## Policy Director commands file

The following is an example of the Policy Director commands file for the user-registration solution:

```

group create UserRegistrationGrp "cn=UserRegistrationGrp, o=ePICGroups, o=ePIC"
UserRegistrationGrp
group create UserRegistrationBuyerGrp "cn=UserRegistrationBuyerGrp, o=ePICGroups,
o=ePIC" UserRegistrationBuyerGrp

acl create UserRegistrationAcl
acl create UserRegistrationBuyerAcl

acl modify UserRegistrationAcl set group UserRegistrationGrp Tvr
acl modify UserRegistrationBuyerAcl set group UserRegistrationBuyerGrp Tv
acl attach "UserRegistration/BtoBi/UserRegistration" UserRegistrationAcl
acl attach "UserRegistration/BtoBi/UserRegistration/Buyer"

UserRegistrationBuyerAcl

```

## Policy Director map file

The following is an example of a Policy Director map file for the user-registration solution:

```

/BtoBi/UserRegistration/Buyer/com.ibm.bpip.studio.wizards.ExtractSubComponentFiles

```

## Policy Director configuration file

The following is an example of a Policy Director map file for the user-registration solution:

```

<?xml version="1.0"?>
<instructions>
<instruction>
<type>PDUtility</type>
<method>updateMapFile</method>
<parameter type="String" value="solName"></parameter>
<parameter type="String" value="mapFilename"></parameter>
</instruction>
<instruction>\
<type>PDUtility</type>
<method>execPDAdmin</method>
<parameter type="String" value="PDCmdsfile"></parameter>
</instruction>
</instructions>

```



---

## Chapter 12. Servlet

This chapter describes the creation of servlets and servlet configuration artifacts.

---

### Overview

The Servlet element for your solution comprises the many servlets you create for it. These servlets are stored in the servlet folder in the project view until published. Each Solution Studio solution has one entry servlet that is the initial point of contact between the user and the solution within the Trusted Zone. A solution typically has many more servlets; the number depends on the complexity of the solution you create.

Servlets are used by the solution in the Business Integrator run-time environment to perform a variety of tasks. For instance, in the user-registration sample template, the Logon and UserReg servlets perform many tasks based on a user's role and the activities that a user can perform in the business process, including the following:

- Returning an HTML form page as the result of a request for new registration
- Performing initial verification of the user ID
- Requesting that Registration Services (a session bean) create a persistent Business Data Object (BDO)
- Recognizing the user ID as belonging to a particular role in the solution
- Checking with Workflow Services (a session bean) for work that is pending for a particular user based on the user's role
- Returning an HTML form page with activities that can be performed based on the user's role
- Responding to requests from the user to perform an activity by returning another HTML form page that is based on the user's role and the activity the user is to perform

These servlets can create the browser client views for the user.

### Artifact types

The artifacts are servlets or Java class files.

### Supported tool

The supported tool for creating any type of servlet in Solution Studio is VisualAge for Java. The tool must be installed on the machine and configured

to operate with WebSphere Studio. If it is not installed, you must install it and then configure it with WebSphere Studio. For information on configuring WebSphere Studio and VisualAge for Java to work together, refer to the *WebSphere Studio Business Integrator Extensions Installation Guide*, or refer to the configuration information in the WebSphere Studio and VisualAge for Java documentation.

## Business Integrator considerations

Store all servlets in the servlet folder of the project view. Because the servlets control the user's interaction through a browser with the private processes in the run-time production environment, the servlets for a solution are published to the Interaction Manager (for Enterprise configuration) or the Web Portal (for Entry configuration).

---

## Creating the artifact

You create servlets by using VisualAge for Java. Refer to that product's documentation for information about using VisualAge for Java to create servlets and other Java artifacts for your solution.

### Creating the servlet artifact file

Each servlet requires a configuration file. To create a configuration file, perform the following steps:

1. With the appropriate folder (typically the servlet folder) selected in the project view of WebSphere Studio, click **Tools** → **Wizards** → **WebSphere Artifact Configuration Wizard**. The wizard opens.
2. In the **Configuration File** field, specify the name of the XML file that contains the configuration data.
3. In the **Resource Type** drop-down list, click **Servlet**. The wizard displays all servlet artifacts that are available for you to work with.
4. Click the radio button next to the servlet for which you want to create configuration data.
5. Enter the appropriate information in the **Persistence**, **Web Application**, **Class Name**, and **Alias** fields.
6. If you want to provide configuration information for more than one servlet, click **Save Configuration** to save the configuration information for the current servlet.
7. When you have finished creating all configuration information for all of the affected servlets, click **Finish**. Solution Studio creates an XML configuration file for each artifact, stores it in the repository, and closes the wizard.

See the online help for additional information.

---

## Artifact example

The following is an example of Admin.servlet, the XML configuration file of the Admin servlet:

```
<?xml version="1.0"?>
<servlet>
  <page-list>
    <default-page>
      <uri>/SCM/SCMDesktop.jsp</uri>
    </default-page>
    <page>
      <uri>/UR/UserRegApproval.jsp</uri>
      <page-name>PendingApprovals</page-name>
    </page>
    <page>
      <uri>/UR/MPAdminWelcome.html</uri>
      <page-name>AfterApprovals</page-name>
    </page>
  </page-list>
  <code>com.ibm.b2bi.templates.servlets.Admin</code>
  <description>eMarketplace</description>
</servlet>
```



---

## Chapter 13. Resources and rules (Personalization)

This chapter provides an overview of WebSphere Personalization for Business Integrator. This is an optional feature that can be used to provide role-based content for users. It is available only in the Enterprise configuration.

---

### Overview

For Business Integrator's Enterprise configuration, WebSphere Personalization can provide the user with content and navigation based on the user's role. Personalization has both a run-time and a build-time (or development) environment.

For Business Integrator, the WebSphere Personalization product uses three components:

- WebSphere Application Server Advanced Edition
- WebSphere Personalization Resource engine
- WebSphere Personalization Rules engine

These components must be installed when the WebSphere Personalization product and the Business Integrator Enterprise configuration product are installed. WebSphere Application Server Advanced Edition is the foundation for the run time. The rules engine provides the execution environment for the rules developed in the development environment. The resource engine defines users and content in the run-time environment.

WebSphere Studio and VisualAge for Java provide the development environment for personalization. WebSphere Studio resource-definition wizards and the Rules Editor together enable you to define and match users and content.

The WebSphere Personalization Resource Console enables you to specify resource collections and create resource hierarchies. Resources can be both users and content. The console can also be used to categorize users or content into groups and to associate properties with those groups. These user and content resources can be part of multiple groups with hierarchical relationships that enable you to change group properties without modifying the database or changing the schema.

The WebSphere Studio Rules Editor provides a GUI for creating rules based on user and content resources created by using the Resource Console. The rules specify how to match users with content.

The user views (Web pages) can be created by using WebSphere Studio Page Designer, which implements the rules.

### **Artifact types**

Artifact types include JSP files, beans, and enterprise beans. Personalization resource files are in XML format. For more information on WebSphere Personalization files, refer to the WebSphere Personalization documentation.

### **Supported tool**

The supported tool for creating any personalization for a user's browser view is WebSphere Personalization. For information on using WebSphere Personalization, refer to the WebSphere Personalization documentation.

### **Business Integrator considerations**

When you consider using WebSphere Personalization for your Business Integrator solution, consider how personalization is to be integrated with your solution and which Business Integrator service is to be used to provide the personalization.

Personalization templates are provided as part of Solution Studio. If your solution is targeted for an Enterprise configuration Business Integrator run-time environment, you can modify these templates for use in your solution.

---

## **Creating the artifacts**

Refer to the WebSphere Personalization documentation for information on creating the rules and resources artifacts for your solution.

---

## Part 3. Solution deployment

<b>Chapter 14. Solution publishing</b> . . . . .	81	Creating a deployable solution package	84
Overview . . . . .	81	<b>Chapter 15. Solution deployment</b> . . . . .	89
Before creating the deployable solution		Deploying the solution . . . . .	89
package . . . . .	82	Deploying the solution package . . . . .	89
Deployable solution package . . . . .	83		
Solution deployment model . . . . .	83		
Creating the solution metadata file . . . . .	84		

This part provides a description of the final phase of the solution life cycle, solution deployment. This phase of the solution life cycle spans both the build-time and the run-time environments. Solution publishing is performed in build time by using Solution Studio wizards and WebSphere Studio. Solution deployment is the process of physically moving the solution elements and artifacts to the physical machine where they run; it occurs during run time.





---

## Chapter 14. Solution publishing

The solution-publishing step of the deployment phase is performed as the last phase of build time. It involves preparing a deployable solution package. This chapter provides an overview of solution publishing and describes the process of creating a deployable solution package. Read the entire chapter before publishing your solution. Online help is available for using the Solution Studio Publishing Wizard.

---

### Overview

The publishing step produces a *deployable solution package*. This package contains all solution artifacts and associated configuration scripts logically grouped and targeted for a particular run-time server in the run-time environment. A run-time server is a component or manager in the run-time environment (for example, the Business Flow Manager (BFM) or the Trust and Access Manager (TAM)) that is already installed and ready to execute a solution. See Table 14 for a list of managers and the types of artifacts they manage.

Table 14. Types of solution artifacts deployed to Business Integrator managers

Business Integrator manager	Artifact types
Business Flow Manager (BFM)	Workflows, FDL files, business data objects, microflows, state machines
Information Delivery Manager (IDM)	Messages: BODs and message maps, EDI
Interaction Manager (IM)	JSP files, servlets, Java classes, HTML
Trust and Access Manager (TAM)	Organization and roles information
Gateway	Public trading processes

Grouping artifacts and associating them with logical run-time servers is called *logical mapping*. Logical mapping has a corresponding *physical mapping* that occurs as part of the run-time step of the solution-deployment phase.

After all of a solution's artifacts and corresponding configuration scripts are created by Application Developers, someone in the role of a deployer performs the publishing step by using the Solution Studio Deployment wizard and WebSphere Studio's publishing feature. This involves manually constructing the Business Integrator run-time servers in the WebSphere Studio publishing view, then moving the artifacts that comprise solution elements from their project folders in the project view to the appropriate server in the publishing view. The person publishing the solution must know which

Business Integrator manager handles which artifact files or folder of the solution. For example, an MQSeries Workflow FDL file in the Business Process folder in the project view must be moved to the BFM server in the publishing view because the BFM manages workflows in the run-time environment.

Figure 22 is a snapshot of WebSphere Studio showing the project and publishing views as it is being used to publish a solution. The person publishing the solution drags the artifact files from a project folder in the project view and drops them on the appropriate server in the publishing view. There is not necessarily a one-to-one correspondence between folders in the project view and components in the publishing view. Not all artifacts in a project folder are published to a server. For instance, design artifacts can be stored in a project folder for reference but are not moved to a server because they do not contribute to the execution of the solution.

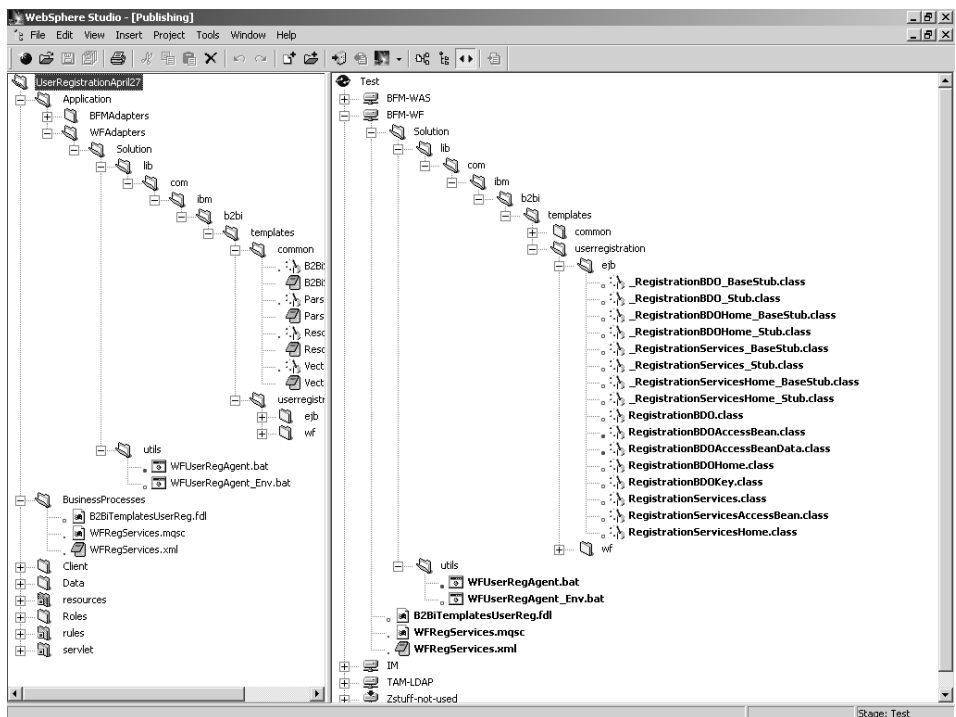


Figure 22. Publishing the Business Integrator solution by using WebSphere Studio

## Before creating the deployable solution package

Before you create the deployable solution package, ensure that the artifacts for the solution are complete and checked into the ClearCase LT repository. In addition, enterprise beans and servlets require configuration data to be successfully deployed and configured to operate in the run-time environment.

This configuration data is stored in XML files. For an enterprise bean, the associated XML file contains the name and location of the EJB container. For servlets, this file contains the name of the servlet engine to use, the name of the Web application associated with the servlet, and the name of the alias.

Use the WebSphere Artifact Configuration wizard in Solution Studio to create XML files. Information about using this wizard is available in the online help.

When you publish the project, both the artifact and its configuration file are included in the deployable solution package.

## **Deployable solution package**

The deployable solution package is a zip file that contains the solution's artifact files and their configuration scripts in the form of zip files and a solution metadata file (`metadata.xml`). The metadata file defines where the artifact files and their configuration or deployment scripts are stored in the zip file.

### **Solution deployment model**

The solution-deployment model organizes any Business Integrator solution according to a hierarchy. A Business Integrator solution is made up of components that are made up of subcomponents that contain artifacts. The deployment model has the following hierarchy:

#### **Solution**

This identifies the solution to be deployed, including a name and a version number. It also lists the components of the solution.

#### **Component**

This identifies a particular Business Integrator manager. A component consists of an arbitrary number of subcomponents. See Table 4 on page 14 for a listing of valid components and subcomponents.

#### **Subcomponent**

This identifies the software that the manager is running on. A typical subcomponent is WebSphere Application Server. A subcomponent defines a group of artifacts in the following way:

**zip** The name of the zip file containing all artifacts of the subcomponent.

**script** The name of an XML file containing all the functions for installing, activating, and uninstalling the subcomponent.

#### **manual**

The name of a file containing a set of manual instructions used when deploying the subcomponent to the run-time environment.

## Artifact

Anything that can be contained in a file. Each artifact is detailed within this structure by its name and type. For instance, an artifact can be an enterprise bean's JAR file, an FDL file, or a script.

The following example of a deployable solution package when it is unzipped illustrates the hierarchical structure of the zip file:

```
UserRegistration\metadata.xml
UserRegistration\BFM\was\bfm.zip
UserRegistration\WM\wf\flow.zip
UserRegistration\WP\was\webserver.zip
```

The manual file in the package contains instructions for configuring a run-time component after the element has been installed. For instance, when installing an FDL file, you must run MQSeries Workflow's import facility to initialize it with the new process model. Include instructions for performing this activity as part of the deployable solution package.

## Creating the solution metadata file

The solution metadata file is an XML file named metadata.xml that defines where the solution artifacts are stored in the deployable solution package's zip file along with their dependencies and deployment control scripts. This file is created in Solution Studio by using the Deployment wizard. For deployment in the run-time environment, the wizard places this file in the root directory of the deployable solution package's zip file. Online help is available for using the Deployment wizard.

## Creating a deployable solution package

Complete procedures for creating the deployable solution package are available in the Solution Studio online help. The following is an overview of how to create a deployable solution package. It assumes that all the artifacts and any associated artifact configuration scripts are already created. To create a deployable solution package, you must also be familiar with the run-time topology to which the solution is to be deployed. The servers that are created during publishing are physically mapped to specific machines as part of the deployment process.

1. With WebSphere Studio running and the project folder selected in the project view as shown in Figure 23 on page 85, use WebSphere Studio to create the Business Integrator servers for the solution in the publishing view.

WebSphere Studio creates a local host server in the publishing view and creates subfolders that match the project view folders for your solution. Create new servers by clicking **Insert** → **Server** from the main menu. Keep the Insert Server panel open until you have finished creating all the servers needed for your solution based on your topology.

Define all of your servers in the publishing view.

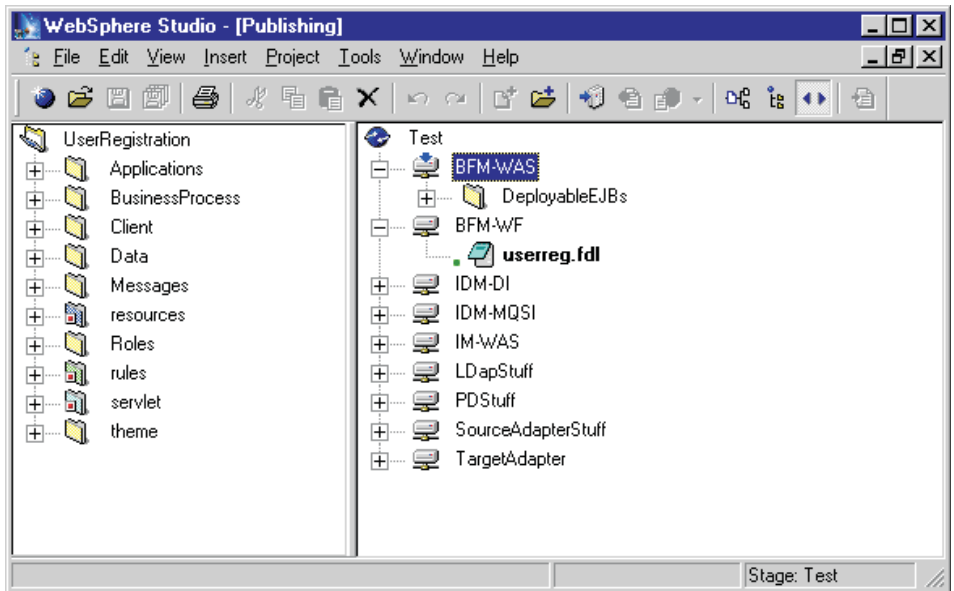


Figure 23. Publishing the solution in WebSphere Studio

2. By using WebSphere Studio's drag and drop feature, move the artifacts and configuration files from the project-view folders to the appropriate servers in the publishing view. This maps the artifacts from the project view to the publishing view. You can move the entire contents of project folders to servers by dragging the folder to the appropriate server and dropping it there. For instance, the entire contents of the Roles folder can be moved to the LDAP server in the publishing view. Alternatively, you can expand the project folder and move selected artifacts to the appropriate servers.

When you move some artifacts or files to the servers in the publishing view, WebSphere Studio can potentially not recognize them as publishable (for instance, LDAP configuration files (.ldif)). You are prompted whether or not to publish them. Click **Yes** in the dialog box for all files you know to be publishable. Primarily, it is design artifact files (except FDLs) that you do not want to publish.

3. When you have all the artifacts for your solution moved to the appropriate servers, delete the local host server that WebSphere Studio created.
4. Right-click the project in the publishing view, then click **Publish whole project**. When you publish the project, the Deployment wizard starts. The left side of the wizard lists all the possible servers available in the run-time environment based on the selected topology. Your solution does

not have to use all possible servers. The wizard enables you to create manual instructions for each server. You can create the instructions once for each particular server. Each time you create a server, click **Finish** to move on to create the next server.

The wizard steps through the process of creating the deployable solution package. It compresses the contents of the servers into individual zip files that contain the artifact files, their XML deployment scripts, and any additional files required by the deployment scripts. A sample zip file and its contents are shown in Figure 24.

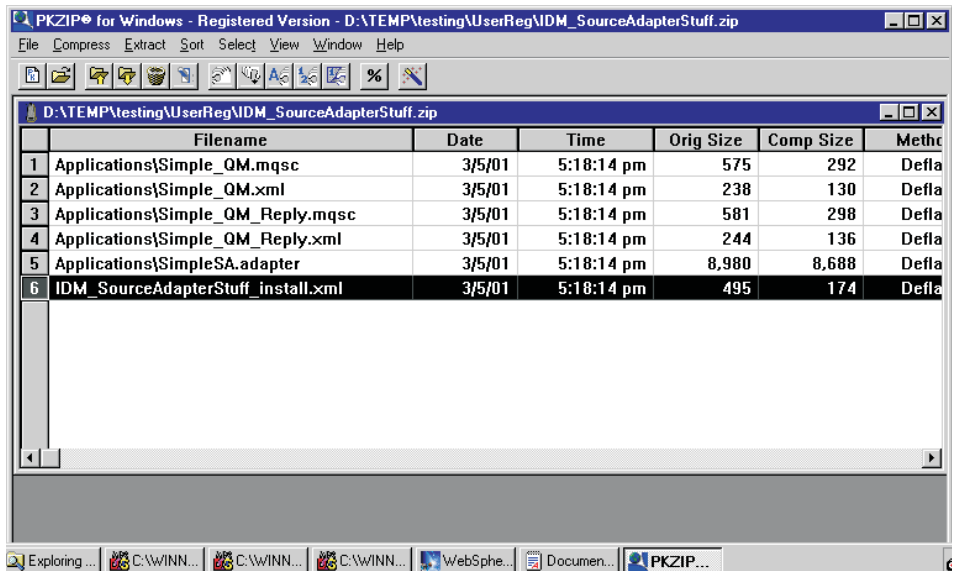


Figure 24. Compressed solution elements for the deployable solution package

The wizard creates the metadata.xml file, then compresses the individual zip files and the metadata file into one zip package, as shown in Figure 25 on page 87.

	Filename	Date	Time	Orig Size	Comp Size	Method	Attr
1	BFM_BFM-WAS.zip	3/5/01	5:19:06 pm	76,672	76,525	DeflatedN	w
2	BFM_BFM-WF.zip	3/5/01	5:19:06 pm	3,876	3,787	DeflatedN	w
3	IDM_IDM-DI.zip	3/5/01	5:19:06 pm	27,170	26,045	DeflatedN	w
4	IDM_IDM-MQSI.zip	3/5/01	5:19:06 pm	32,967	26,032	DeflatedN	w
5	IDM_SourceAdapterStuff.zip	3/5/01	5:19:08 pm	10,654	9,903	DeflatedN	w
6	IDM_TargetAdapter.zip	3/5/01	5:19:08 pm	11,826	11,141	DeflatedN	w
7	IM_IM-WAS.zip	3/5/01	5:19:08 pm	156,360	145,134	DeflatedN	w
8	metadata.xml	3/5/01	5:19:08 pm	3,019	463	DeflatedN	w
9	TAM_LDapStuff.zip	3/5/01	5:19:08 pm	1,585	1,286	DeflatedN	w
10	TAM_PDapStuff.zip	3/5/01	5:19:08 pm	1,411	1,078	DeflatedN	w

Figure 25. Deployable solution package file

The deployable solution package is stored on the drive where WebSphere Studio is installed. The compressed file is given the file name of the solution with a .zip extension. It is stored in a folder that has the same name as the project name. If the folder does not exist, the Deployment wizard creates it and stores the package there. Figure 26 on page 88 shows the solution package.

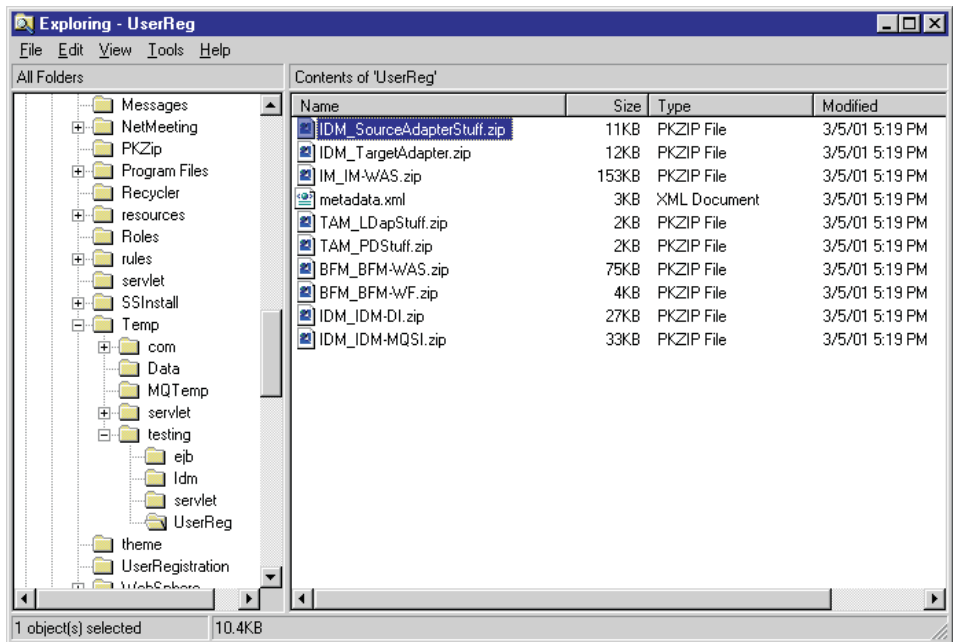


Figure 26. Storing the deployable solution package for movement to the run-time test or production environment

After the package is created, it is ready for the run-time test or production environment. The deployable solution package can be moved to the run-time environment by any of the following methods:

- Diskette, CD-ROM, or other portable media
- Mapped network drive
- FTP

The deployable solution package is ready for the next step in the solution-deployment phase. For a brief overview of the deployment process, see “Chapter 15. Solution deployment” on page 89. For details on deploying a solution, refer to the deployment chapter in the *WebSphere Business Integrator Run Time* book.



---

## Chapter 15. Solution deployment

This chapter provides a brief description of the deployment of a solution package to the run-time environment. For details of deploying a solution to the run time, refer to the *WebSphere Business Integrator Run Time* book.

---

### Deploying the solution

Deploying the solution takes place in the run-time environment after the solution is packaged into a zip file. Typically, an administrator familiar with the particular run-time environment—that is, the physical view of the topology in which the solution is to run—deploys and installs the solution by using the Solution Deployment wizard from the Platform Console. Some solutions can require additional configuration at the various servers. Complete procedures for deploying the solution and using the Solution Deployment wizard are provided in the *WebSphere Business Integrator Run Time* book.

In the run-time environment, the solution's logical component (for example, the Business Flow Manager) is mapped to the physical system. The solution elements are targeted to specific machines, and the install paths for the solution elements are specified. The solution elements are transported to the target machines, and the elements are installed and activated by using the XML files packaged with the solution elements. During deployment, the topology is updated to add solution artifacts so that they can be managed.

The Solution Deployment wizard provides automated deployment of solution components. It outlines the content of the solution for confirmation and updates the topology repository throughout the deployment process. If a software product does not support automated deployment, the wizard instructs the person deploying the solution to perform the required steps manually and confirm completion.

---

### Deploying the solution package

The following is an overview of how to deploy a solution in the run-time environment. For complete procedures for deploying the solution and using the Solution Deployment Wizard from the Platform Console, refer to the *WebSphere Business Integrator Run Time* book.

A solution is deployed to an installed run-time environment. The installation of the solution occurs on machines on which Business Integrator is already installed. Deploying the solution begins with a deployable solution package and involves several steps:

1. Moving the deployable solution package to a run-time environment
2. Unpacking the solution package
3. Creating the physical mapping of solution elements to specific machines in the run-time environment
4. Distributing the solution elements to individual machines in the run-time environment and configuring them as needed
5. Installing the solution elements on individual machines by using the artifact configuration scripts
6. If necessary, configuring some run-time components manually
7. Activating the solution

---

## Part 4. Appendixes



---

## 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 on 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.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**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 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.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Limited,  
Intellectual Property Department  
Hursley Park  
Winchester SO21 2JN  
United Kingdom

Such information may be available, subject to appropriate terms and conditions, including, in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained here in was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measures may have been made on development-level systems, and there is no guarantee that these measurements will be the same on generally available system. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the application data of their specific environment.

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 claim related to non-IBM products. Questions on capabilities of non-IBM products should be addressed to the suppliers of those products.

Any performance data contained here in was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measures may have been made on development-level systems, and there is no guarantee that these measurements will be the same on generally available system. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the application data of their specific environment.

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 claim related to non-IBM products. Questions on 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 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 illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purpose of developing, using, marketing or distributing application programs conforming to IBM's application programming interfaces 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

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

- DB2
- IBM
- MQSeries
- Process Choreographer
- SecureWay
- VisualAge

- WebSphere

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

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

Rational and ClearCase are registered trademarks of Rational Software Corporation.

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



---

## Bibliography

This bibliography lists the books in the IBM WebSphere Business Integrator and associated libraries.

---

### IBM WebSphere Business Integrator library

The Business Integrator library consists of the following books:

- *WebSphere Business Integrator Concepts and Planning*, GC34-5960  
This book introduces the Business Integrator system, providing a high-level system overview, defining the system capabilities, and describing its value to e-businesses. This book also provides the information that you need to plan the installation of Business Integrator.
- *WebSphere Business Integrator Installation Guide*, GC34-5961  
This book is a guide to installing and configuring Business Integrator. It contains information about:
  - Selecting your required topology
  - Installing and configuring the base products and software components of Business Integrator on each machine in the topology
  - Installing and configuring firewalls and proxies
- *WebSphere Studio Business Integrator Extensions Installation Guide*, SC34-5962  
This book is a guide to installing and configuring Solution Studio. It also contains information about setting up clients and servers, and creating projects.
- *WebSphere Business Integrator Run Time*  
This book is a comprehensive guide to the Business Integrator runtime system, providing the following information:
  - Detailed conceptual information about the run-time components of Business Integrator
  - Deployment of solutions to the run-time system
  - System administration, such as starting and stopping software components and base products, defining users, and using the Exception Console
  - General problem determination information, including how to trace and debug, and information on obtaining help from technical support

- *WebSphere Business Integrator Messages*  
This book lists the error messages that are produced by Business Integrator and provides references to the documentation for the messages of base products.
- *WebSphere Studio Business Integrator Extensions Developer's Guide*  
This book describes how to create a Business Integrator solution, beginning with the solution design phase, to the solution implementation phase, and finally the solution deployment phase by using a sample business problem. This book also provides procedures for assembling a Business Integrator solution in the run-time environment and a description of how to use Solution Studio for solution design and implementation.
- *WebSphere Business Integrator DataInterchange for Windows NT User's Guide, SC34-5963*  
This book is a guide to installing and using DataInterchange in the Business Integrator environment.

You can find the latest versions of the books at the following Web site:

<http://www-4.ibm.com/software/webservers/btobintegrator/>

This site contains links to the Web sites of the underlying products of IBM WebSphere Business Integrator.

## **Related documentation**

The utilities subdirectory on the Documentation CD contains documentation about utilities that can prove useful in building and running solutions. This documentation is not available on the IBM WebSphere Business Integrator Web site.

WebSphere Business Integrator also provides a number of external application programming interfaces (APIs). HTML documentation that is generated by using the Javadoc tool is provided for these APIs. For a list of the APIs, refer to the *WebSphere Business Integrator Run Time* book.

---

## WebSphere Partner Agreement Manager library

The Partner Agreement Manager Version 2 Release 1 library consists of:

- *Partner Agreement Manager Installation Guide*, GC34-5964
- *Partner Agreement Manager Administrator's Guide*
- *Partner Agreement Manager User's Guide*
- *Partner Agreement Manager Adapter Developer's Guide*
- *Partner Agreement Manager Script Developer's Guide*
- *Partner Agreement Manager API Guide*
- *Partner Agreement Manager Adapters for MQSeries User's Guide*
- *Partner Agreement View User's Guide*, GC34-5965
- *WebSphere Partner Agreement Manager Business Process Integration Adapter Guide*

---

## DataInterchange library

The DataInterchange Version 3 Release 1 library consists of:

- *DataInterchange Client User's Guide*, SB34-2010
- *DataInterchange Administrator's Guide*, SB34-2002
- *DataInterchange Installation Guide*, GB09-8070
- *DataInterchange Messages and Codes*, SB34-2000
- *DataInterchange Programmer's Reference*, SB34-2001

---

## Other libraries

You can find important information in the libraries of the following products:

- DB2<sup>®</sup> UDB
  - *IBM DB2 Universal Database Quick Beginnings Version 6.1*, S10J-8149
- MQSeries<sup>®</sup>
  - *MQSeries for Windows NT Quick Beginnings*, GC34-5389
  - *MQSeries System Administration*, SC33-1873
  - *MQSeries Using Java*, SC34-5456
  - *MQSeries MQSC Command Reference*, SC33-1369
  - *MQSeries Queue Manager Clusters*, SC34-5349
  - *MQSeries Integrator Introduction and Planning*, GC24-5599
  - *MQSeries Workflow Getting Started with Buildtime*, SH12-6286
  - *MQSeries Workflow Getting Started with Runtime*, SH12-6287
  - *MQSeries Adapter Kernel for Multiplatforms: Quick Beginnings*, GC34-5855
  - *MQSeries Adapter Kernel: Problem Determination Guide*, GC34-5897

- *MQSeries Adapter Builder for Windows NT: Using the Control Center, GC34-5882*
- SecureWay®
  - *SecureWay Policy Director Up and Running, SCT6-3KNA*
  - *SecureWay Policy Director Base Administration Guide*
  - *SecureWay Firewall User's Guide, CG31-8658*
- VisualAge®
  - *VisualAge Java, Enterprise Edition Getting Started*
  - *VisualAge C++ Professional for Windows NT Getting Started*
- WebSphere® Application Server
  - *Introduction to WebSphere Application Server, SC09-4430*

---

# Index

## A

- administrators 6
- application
  - specialists 6
- application/domain specialists 19
- artifact
  - editing 17

## B

- Business Integrator
  - consultants 6
  - system overview 3
- Business Integrator consultants 19
- business process
  - modeling 5
  - transforming 25
- business-process modeling 19, 21
- business transformation 5, 19, 25
  - consultants 6
- business-transformation consultants 19

## C

- ClearCase repository 11
- client modeling 5
- consultants
  - business transformation 6
- consultants, Business Integrator 6
- creating
  - solution artifact 17
- customizing Business Integrator
  - components 5

## D

- defining Business Integrator
  - platform requirements 5
- distributing the solution 5
- domain specialists 6

## E

- e-marketplace 4
- editing an artifact 17
- exiting Solution Studio 18
- exiting WebSphere Studio 18

## F

- flow
  - integration of process 3

## I

- information
  - modeling 5
- information modeling 19
- integration process flow 3
- IT architect/specialists 6, 19

## L

- life cycle of a solution 4

## M

- modeling
  - business transformation 19
  - client 5, 19
  - information 5, 19
  - organization 5, 19
  - process 21
  - task 5, 19

## O

- organization
  - modeling 5
- organization modeling 19
- overview
  - Business Integrator system 3
  - Solution Studio 9

## P

- phase
  - solution design 5
  - solution development 5
- phases of a solution life cycle 4
- players, role 6
- process flow
  - integration of 3
- process modeling 21
- programmers 6
- project managers 6
- project view 10
- publishing view 10

## R

- repository
  - ClearCase 11
  - Solution Studio 11
- role players 6

## S

- solution artifact
  - creating 17

- solution artifact (*continued*)
  - editing 17
- solution assemblers 6
- solution deployment
  - description of 5
- solution design
  - activities 19
  - activities of phase 5
  - description of 5
  - role players 19
- solution development
  - description of 5
- solution elements 5
- solution lifecycle
  - description of 4
  - phases of 4
- Solution Studio 9
  - editing a solution artifact 17
  - exiting 18
  - overview 9
  - repository 11
  - starting 16
  - using 16
- Solution Studio wizards 11
- starting Solution Studio 16
- starting the WebSphere Studio workbench 16
- starting WebSphere Studio 16
- supply chain 4
- support analysts 6
- system
  - administrators 6
  - programmers 6

## T

- task modeling 19
- tasks
  - modeling 5
- trading community 4
- trading partners 4
- transforming business processes 25

## U

- using Solution Studio 16
- utilities 98

## W

- WebSphere Studio
  - exiting 18
- Wizards, Solution Studio 11





File Number: BIZAAF00



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.