

IBM Integration Bus 9.0.0
Version 1 Release 0

*IBM Integration Bus 9.0.0
Installing and Migrating*

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 671.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright IBM Corporation 1999, 2014.**

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




PDF books and IBM Knowledge Center

PDF books are provided as a convenience for printing and offline reading. For the latest information, see the online product documentation in IBM® Knowledge Center.

The PDF documentation is updated less frequently than the online product documentation in IBM Knowledge Center.

Links to product documentation topics outside this PDF book go to the online product documentation in IBM Knowledge Center. Links to targets outside a PDF book are marked by icons that indicate whether the target is a PDF book or a web page.

Table 1. Icons that prefix links to topics outside this book

Icon	Description
	A link to a page in the online product documentation in IBM Knowledge Center.
<i>Figure 1.</i>	
	A link to a web page.
<i>Figure 2.</i>	
	A link to a PDF book.
<i>Figure 3.</i>	

Contents

PDF books and IBM Knowledge Center iii

Figures vii

Chapter 1. Planning your IBM Integration

Bus solution 1

Finding the latest information 1

Understanding your processing requirements 2

Choosing your IBM Integration Bus edition and operation mode 4

Operation modes 5

Restrictions that apply in each operation mode 10

Planning an IBM Integration Bus environment. 14

Planning the high-level infrastructure. 15

Business planning 20

Cloud overview 22

Performance planning 27

Planning for security 30

Chapter 2. Installing and uninstalling 31

Installing 31

Installation Guide 32

Preparing for installation 34

Coexistence and migration 38

Preparing the system 43

Choosing what to install 59

Installing by using the Windows Launchpad 60

Installing IBM Integration Bus components and prerequisite products by using the sample script file 65

Installing the Integration Bus component 69

Installing the IBM Integration Toolkit. 79

Installing IBM Integration Explorer 84

Installing IBM Integration Administration for WebSphere Application Server 93

Verifying your IBM Integration Bus installation 104

Checking the broker operation mode and function level 111

Installing complementary products 114

Setting up a command environment. 119

Applying service 129

Uninstalling 150

Uninstalling the Integration Bus component 151

Uninstalling the IBM Integration Toolkit 159

Uninstalling the IBM Integration Explorer 164

Chapter 3. Migrating 171

Preparing for migration 172

Behavioral changes in Version 9.0 174

Supported migration paths 177

In-place broker migration 178

Parallel broker migration 180

Migration tasks. 182

Migrating development resources to IBM Integration Toolkit Version 9.0. 182

Migrating the Version 8.0 WebSphere Message Broker component. 188

Migrating the Version 7.0 WebSphere Message Broker component. 206

Migrating the Version 6.1 WebSphere Message Broker component. 225

Migrating the WebSphere Message Broker Explorer 250

Perform post-migration tasks 252

Reviewing technical changes in IBM Integration Bus Version 9.0. 253

Migrating deployable resources 259

Setting up a command environment. 262

Migrating a flow that contains JMS nodes 263

Migrating a flow that contains File nodes 264

Migrating a flow supporting ?wsdl queries 265

Migrating CMP applications 265

Updating error processing routines 266

Chapter 4. Configuring brokers for development environments 273

Creating a default configuration 274

Creating the default configuration 274

Removing the default configuration 275

Creating a development environment 277

Creating a broker for a development environment. 279

Configuring the Toolkit 280

Changing IBM Integration Toolkit preferences 281

Changing workbench capabilities. 282

Configuring CVS to run with the IBM Integration Toolkit. 283

Configuring the IBM Integration Toolkit to run Rational ClearCase 284

Creating a working set 285

Integrating the Rational Team Concert client with the IBM Integration Toolkit 287

Chapter 5. Configuring brokers for test and production environments 289

Customizing the z/OS environment. 290

Customizing UNIX System Services on z/OS 291

Resource Recovery Service planning on z/OS 295

Configuring WebSphere MQ on z/OS 295

Defining the started tasks to z/OS Workload Manager (WLM) 296

Automatic Restart Manager planning 297

Mounting file systems 298

Checking the permission of the installation directory 300

Customizing the version of Java on z/OS 301

Checking APF attributes of bipimain on z/OS 301

Configuring an integration server address space as non-swappable on z/OS	302
Configuring brokers	303
Creating a broker	304
Verifying brokers	330
Modifying a broker	331
Modifying a broker on Windows, Linux, and UNIX systems	332
Modifying a broker on z/OS	334
Configuring integration nodes using the IBM Integration Explorer	336
Configuring integration node properties in the IBM Integration Explorer	337
Using the IBM Integration Explorer to work with configurable services	345
Working with UserDefined configurable services	354
Changing IBM Integration Explorer preferences	356
Changing the operation mode of your broker.	356
Moving from Developer Edition	358
Checking the operation mode of your broker.	359
Advanced configuration	360
Configuring for applications	361
Web Services Reliable Messaging	469
WS-Security	473
Moving from IBM Integration Bus on a distributed system to z/OS.	535
Changing locales	536
Configuring for high availability	544
Using multi-instance brokers	544
Using a broker with an existing high availability manager	571
Using a broker with an existing Windows Cluster (Windows Server)	582
HTTP proxy servlet overview	584
Using external web servers with IBM Integration Bus	610
Configuring a broker as a WebSphere MQ service	614
Starting and stopping a broker as a WebSphere MQ service	614
Modifying the WebSphere MQ service for a broker.	616

Deleting the WebSphere MQ service for a broker	617
Reporting and displaying the status of a broker that runs as a WebSphere MQ service	618

Appendix. Installation 619

System requirements	619
Hardware requirements	620
Software requirements	625
License requirements	641
General industry standards supported by IBM	
Integration Bus	643
Installation packages	645
Packaging options	646
Package contents	648
Installation and uninstallation interfaces	652
How to install and uninstall the Integration Bus component	653
How to install and uninstall the IBM Integration Toolkit.	657
Installation wizard names	660
Installation and uninstallation authorization	661
Multicultural support.	663
Locales	664
System changes caused by installation and configuration	664
Start and main menu updates after installation	665
Registry changes created by installation and configuration	667
Environment variables after installation	668
Default WebSphere MQ resources created during installation and configuration	668

Notices 671

Programming interface information	673
Trademarks	673

Sending your comments to IBM . . . 675

Figures

1.	iii	10.	This graphic shows the interactions between message broker and server when the SOAP Request node is used.	494
2.	iii	11.	This graphic shows the interactions between message broker and server when the asynchronous SOAP nodes are used.	495
3.	iii	12.	Multi-instance queue manager	546
4.	Diagram showing the capacity and functionality of the four main modes of operation	6	13.	An Integration Registry configuration for multi-instance integration nodes	571
5.	State machine diagram showing how the presence of nodes and a defined coordination queue manager affect the state of the agent. This diagram summarizes the information in the preceding paragraph.	442	14.	HTTP proxy servlet deployed to a servlet container on the same machine as IBM Integration Bus	586
6.	The initial sender transmits a message from the application source to the reliable messaging source. The reliable messaging source transmits the message to the reliable messaging destination, which acknowledges receipt of the message. The reliable messaging destination then delivers the message to the application destination, and so it reaches the Application Destination.	469	15.	HTTP proxy servlet deployed to a servlet container that is on a different machine to IBM Integration Bus	587
7.	With InOrder set to false, all messages are processed when they are delivered.	470	16.	HTTP proxy servlet deployed to a servlet container that is on a different machine to IBM Integration Bus with a network load-balancer for distributing work to several brokers	589
8.	With InOrder set to True, message 3 is not processed because message 2 has not yet been delivered.	471	17.	WebSphere Application Server plug-in with application servers.	610
9.	This graphic shows the interactions between message broker and client when the SOAP Input and SOAP Reply nodes are used.	493	18.	WebSphere Application Server plug-in with IBM Integration Bus	611
			19.	Java compatibility logo	673

Chapter 1. Planning your IBM Integration Bus solution

Use the topics in this section to help you plan your implementation of IBM Integration Bus.

About this task

The following topics describe the steps involved in planning your implementation of IBM Integration Bus:

- “Finding the latest information”
- “Understanding your processing requirements” on page 2
- “Choosing your IBM Integration Bus edition and operation mode” on page 4
- “Planning an IBM Integration Bus environment” on page 14
- “Planning the high-level infrastructure” on page 15
- “Business planning” on page 20
- “Performance planning” on page 27
- “Planning for security” on page 30

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.

“Performance planning” on page 27

When you design your IBM Integration Bus environment and the associated resources, the decisions that you make can affect the performance of your applications.

“Configuring databases” on page 361

Configure databases to hold application or business data that you can access from your message flows.

Finding the latest information

Access the latest information for IBM Integration Bus.

About this task

The following information is provided:

Requirements website

For the latest details of hardware and software requirements on all supported platforms, visit the IBM Integration Bus Requirements website.

readme.html

The product readme file is frequently updated and includes information about last minute changes and known problems and workarounds. The latest version is always on the product readmes web page; always check to see that you have the latest copy. The version of file that is included on the

product media, and which is installed when you install product components, contains a link to the latest version on the product readmes web page.

z/OS Program Directory for IBM Integration Bus for z/OS®

The program directory describes installation procedures for all components on z/OS. The latest version of the program directory is available in the IBM Publication Center publication number, GI13-3314.

Installation Guide

The Installation Guide is provided in the online information center. See “Installation Guide” on page 32 for details about how to access the guide online and where to find installation information in this information center.

Information center

The information center is installed with the IBM Integration Toolkit and the IBM Integration Explorer, and updates are typically included when you apply service to those components.

The information center is periodically updated independently of the code, and you can install the latest level from within the toolkit. For instructions about installing code and documentation updates, see “Applying service to the IBM Integration Toolkit” on page 141.

The information center is also available online at IBM Integration Bus Library web page.

Support information

The IBM Integration Bus support web page is regularly updated with the latest product support information. For example, if you are migrating from an earlier version, look under the heading “Solve a problem” for the document “Problems and solutions when migrating”.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Applying service to the Integration Bus component” on page 130

Apply maintenance updates and program fixes to the Integration Bus component.

“Applying service to the IBM Integration Toolkit” on page 141

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

Related reference:

“Installation Guide” on page 32

Installation information for IBM Integration Bus is provided in the online information center.

“Software requirements” on page 625

View the operating system, database, and other software requirements.

“Hardware requirements” on page 620

View the processor and related hardware requirements on all platforms.

Understanding your processing requirements

It is important to be clear about your processing requirements, and to understand what you want IBM Integration Bus to do for your business.

About this task

In a typical IT environment there are many interacting components, each with a role to play, and it is important that you are able to clearly define the role of IBM Integration Bus in the broader environment. A key design decision is how much business logic to implement within your message flows. IBM Integration Bus enables you to implement large amounts of processing within a message flow, and implementations can vary significantly from the simple routing of messages to complex validation and transformation. Some implementations also read data from a database and use that data to populate messages. Regardless of the amount of function that you decide to implement in a message flow, it is important to have clear boundaries between pieces of application processing.

Several different processing styles are commonly used with IBM Integration Bus, and it is important to understand the ones that are most relevant to you, such as:

Request Reply

This is the most common type of processing, and enables two applications to communicate, even if they use different data formats. For example, a message flow transforms a request message from Application 1 into a format that Application 2 can understand. The output of the first message flow is then sent to Application 2, which processes the request and issues a reply. The reply is processed in a second message flow, which converts the response message into a format that Application 1 can understand.

Aggregation

This type of processing is often used to invoke one or more back-end systems and coordinate replies. It is a more complex form of a Request Reply case, in which all the replies from the intermediate applications must be collected together before the reply message for the original request can be sent. For example, this type of processing could be used to book a holiday, in which a flight, hotel, and a car are required, and all must be successfully processed before the holiday confirmation can be sent.

Routing

Routing is used to redirect messages, and one or more copies of a message can be sent to one or more destinations.

Transformation

This type of processing involves the use of one or more transformation technologies such as Compute, JavaCompute, XMLT, or Mapping node. In this type of processing the input message is processed according to some business rules, and there might also be a change of message format or protocol.

File processing

The use of files is one of the most common methods of storing data. You can create message flows to process data in files, accepting data in files as input message data, and producing output message data for file-based destinations.

Database handling

You can configure your message flows to access and manipulate business data in databases.

Web services

IBM Integration Bus can be used as both a consumer and provider of web services.

Transport switching

You can use this type of processing to switch between transports such as HTTP and JMS.

For information about how to choose the appropriate edition of IBM Integration Bus to suit your requirements, see “Choosing your IBM Integration Bus edition and operation mode.” For more information about the capabilities provided by IBM Integration Bus, see IBM Integration features.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.



IBM Integration features

Some features of IBM Integration Bus have specific requirements, such as a specific mode of operation or the availability of an additional product, such as a database. Additional terms and conditions also apply to the use of some IBM Integration Bus features, including the requirement for other product licenses.

Related tasks:

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.

“Performance planning” on page 27

When you design your IBM Integration Bus environment and the associated resources, the decisions that you make can affect the performance of your applications.

“Planning the high-level infrastructure” on page 15

When you are planning to use IBM Integration Bus, consider your complete system of products as a whole and ensure that the flow of data throughout the system is carefully planned.

“Planning an IBM Integration Bus environment” on page 14

When you start to plan your IBM Integration Bus environment, you must first consider the design of your system infrastructure. A major consideration when you are planning to use IBM Integration Bus is how it will fit into your overall system architecture, and it is important to understand the movement of data around your whole system.

“Configuring databases” on page 361

Configure databases to hold application or business data that you can access from your message flows.

Choosing your IBM Integration Bus edition and operation mode

The capability and capacity provided by IBM Integration Bus vary according to the operation mode in which your integration nodes are running, and your entitlement to run in a particular mode depends on the edition of the product that you have purchased.

About this task

You can choose the mode of operation for your integration node based on the functionality and capacity that you require. If appropriate, you can install an edition that supports a restricted set of functions, if that subset fulfills your business requirements. You must ensure that your use and configuration of the

product conforms to the license agreement that you have purchased. Some features of IBM Integration Bus have specific requirements, such as a specific mode of operation or the availability of an additional product, such as a database.

The following topics describe the features that are available in each operation mode, together with additional requirements and restrictions:

- IBM Integration features
- “Operation modes”
- “Restrictions that apply in each operation mode” on page 10

For more information about the license requirements for each operation mode, see “License requirements” on page 641.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.

“Performance planning” on page 27

When you design your IBM Integration Bus environment and the associated resources, the decisions that you make can affect the performance of your applications.

“Configuring databases” on page 361

Configure databases to hold application or business data that you can access from your message flows.

Operation modes

The operation mode that you can use for your broker is determined by the license that you purchase.

The following modes are supported:

- “Express mode” on page 7. A limited set of nodes are enabled for use within a single integration server. Message flows are unlimited.
- “Scale mode” on page 7. A limited set of nodes are enabled for use within unlimited integration servers. Message flows are unlimited.
- “Standard mode” on page 7. All features are enabled for use with a single integration server. The number of message flows that you can deploy are unlimited.
- “Advanced mode” on page 8. All features are enabled and no restrictions or limits are imposed. This mode is the default mode, unless you have the Developer Edition.

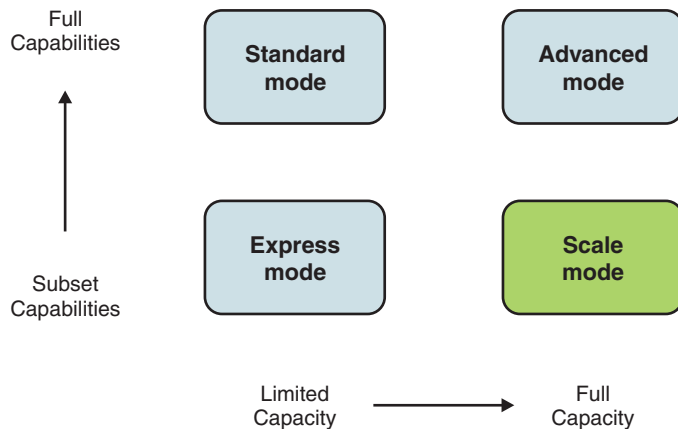


Figure 4. Diagram showing the capacity and functionality of the four main modes of operation

- “Developer mode” on page 8. All features are enabled, but you can use the product for evaluation, development, and test purposes only. Developer Edition is available on Windows x86-64 and Linux x86-64 operating systems and is limited to one message (transaction) per second at the message flow level.
- “Trial mode” on page 8. All features are enabled, but you can use the product for only 90 days after installation. Trial Edition is available on all distributed platforms in WebSphere® Message Broker Version 8.0.0.0, but has been replaced by the Developer Edition from Version 8.0.0.1 onwards.
- “Adapter mode” on page 9. Only adapter-related features are enabled, and the types of node that you can use, and the number of integration servers that you can create, are limited.

Operation modes typically restrict the number of nodes that are available, and integration server capacity, but many features are available across all modes of operation. For an overview of the main IBM Integration Bus capabilities, and the modes of operation in which they are available, see IBM Integration features.

In WebSphere Message Broker Version 8.0, Entry Edition was renamed Express® Edition, Starter Edition was renamed Standard Edition, and Enterprise was renamed Advanced.

You must ensure that your brokers are running in the operation mode for which you have purchased a license. You can set the operation mode when you create a broker by using the **mqsicreatebroker** command. On z/OS you can change the operation mode only after you create a broker; see “Changing the operation mode of your broker” on page 356.

If you have purchased a license for the full package, the Standard Edition, or the Remote Adapter Deployment mode, the broker is automatically created in advanced mode unless you specify the correct mode for your license.

Change the mode of your broker to conform to your license if necessary; see “Changing the operation mode of your broker” on page 356. You can also report the current mode of your broker; see “Checking the operation mode of your broker” on page 359.

The IBM Integration Toolkit remains the same in all modes. All the capabilities of the IBM Integration Toolkit are available in all modes. If you try to deploy too many message flows or integration servers for the mode, or try to use a node that is not valid in the mode, the operation is rejected, and an error message is displayed indicating the reason for the failure; see “Restrictions that apply in each operation mode” on page 10. Node restrictions for a given mode also apply to the use of message flows generated by IBM Integration Bus patterns.

Fix packs are identical for all modes, and can be applied without affecting the validity of the mode.

Express mode

In express mode, the broker operates with a limited set of nodes for use with a single integration server. Message flows are unlimited. For a list of nodes that can be deployed to a broker running in Express Edition mode, see “Restrictions that apply in each operation mode” on page 10.

Because the functions that are enabled and the number of integration servers that you can create are limited, not all samples and patterns work in Express Edition mode. If you want to run samples, see “Development and unit test” on page 9.

Scale mode

In scale mode, the broker operates with a limited set of nodes for use with unlimited integration servers. Message flows are unlimited. For a list of nodes that can be deployed to a broker running in Scale mode, see “Restrictions that apply in each operation mode” on page 10.

Because the functions that are enabled are limited, not all samples and patterns work in Scale mode. If you want to run samples, see “Development and unit test” on page 9.

WebSphere Enterprise Service Bus (WESB) customers can obtain a transfer license, which enables you to migrate from WESB to IBM Integration Bus, and to create and run brokers in Scale mode. As a current WESB customer, you can obtain the following transfer licenses:

- IBM Integration Bus WebSphere Enterprise Service Bus Transfer License
- IBM Integration Bus WebSphere Ent Svc Bus Transfer License Idle Standby

These licenses entitle you to obtain and use IBM Integration Bus in Scale mode in exchange for your WESB product licenses.

The license for IBM Integration Bus Standard Edition also entitles you to run brokers in scale mode, rather than standard mode, if you choose to. For more information about the features that are available in these modes, see IBM Integration features.

Standard mode

In standard mode, the broker operates with all features enabled. Use this edition if you expect to use all or most of the features that are available, but intend to configure a limited environment because of low capacity requirements.

You can use all the available functions, but are limited in the number of resources that you can create and maintain. You are limited to creating one integration

server; for more information, see “Restrictions that apply in each operation mode” on page 10. If you attempt to exceed the limits of this mode, the deployment is rejected.

You cannot use all the samples when your broker is in standard mode, because of the preceding restrictions. If you want to run samples, see “Development and unit test” on page 9.

Advanced mode

In advanced mode, the broker operates with all features enabled, and no operational limits on the creation of integration servers or on the number of flows that are deployed to an individual integration server are enforced. If you want to set up a full broker environment that uses most or all the features available, your brokers must operate in this mode, and you therefore require the full license. If you do not specify another mode, your brokers have the mode set to the default value advanced.

Trial mode

Trial Edition is available for WebSphere Message Broker Version 8.0.0.0, and has been replaced in Version 8.0.0.1 by the Developer Edition.

In trial mode, the broker operates with all features enabled. You can use all available function, and are not limited in the number of resources that you create and maintain. All capability is available for 90 days after installation.

You can download Trial Edition at no charge, from the following website: IBM Integration Bus Trial package.

Developer mode

In developer mode, the broker operates with all features enabled. You can use all available function, and you are not limited in the number of resources that you can create and maintain. IBM Integration Bus for Developers (Developer Edition) is available on Windows x86-64 and Linux x86-64 operating systems, and is provided for evaluative purposes only.

Developer Edition is limited to one message (transaction) per second at the message flow level. Each message flow is able to process one message per second, irrespective of the number of input nodes that are attached to the message flow, or the number of additional instances. When using Developer Edition, if a policy is deployed to increase the message rate throughput above one message per second, a message is reported in the syslog and event log stating that the message rate cannot be changed in Developer Edition. The policy that is deployed has no effect on the message rate.

You can download Developer Edition at no charge from the IBM Integration Bus web page, and you are free to use it for as long as you require, within the terms of the license. Unlike the Trial Edition that was available in previous versions of the product, there is no expiry period for the Developer Edition license.

If you move from WebSphere Message Broker Version 8.0 Trial Edition to IBM Integration Bus for Developers, your Trial mode brokers continue to work as before, but they are treated as Developer mode brokers in the Developer Edition, with no expiry. If you subsequently revert from Developer Edition to a Trial

Edition, your Trial mode brokers (which were treated as Developer mode brokers in the Developer Edition) revert to Trial mode, complete with the expiry period.

When you have installed Developer Edition, if you subsequently purchase a license and install the full version of IBM Integration Bus, any Developer mode brokers that are started with the full version are treated as Advanced mode brokers. In this case, you must also modify the operation mode of these brokers by using the `mqs imode` command to reflect the license that you have purchased. For more information, see “Changing the operation mode of your broker” on page 356.

Adapter mode

In adapter mode, the broker operates with a limited set of nodes available for deployed flows. Use this edition if you expect your typical use of the broker to be integration with Enterprise Information Systems (EIS). This edition supports the subset of development resources that provide EIS interaction. For a list of nodes that can be deployed to a broker running in Remote Adapter Deployment mode, see “Restrictions that apply in each operation mode” on page 10.

You can create up to two integration servers, with no limit on the number of deployed message flows in each of these integration servers; see “Restrictions that apply in each operation mode” on page 10. If you attempt to exceed the limits of this mode, the deployment is rejected.

You cannot use all the patterns and samples when your broker is in adapter mode, because of the preceding restrictions. If you want to run samples, see “Development and unit test.”

Development and unit test

Your license also covers use of the product for development and unit test purposes, but check the license to ensure that you conform to any restrictions for development and unit test. You can view the license for IBM Integration Bus by visiting the Software license agreements search website. Search for "IBM Integration Bus" and choose the license that applies to the version you are using.

Contact your IBM representative if you want further details about license agreements, or if you want to purchase additional licenses or change the type of license that you have purchased.

Integration with Tivoli® License Manager

If you use IBM Tivoli License Manager to control and manage your licensed software products, you must ensure that you choose the correct license for the IBM Integration Bus edition that you have purchased. For more information, see “Installing IBM License Metric Tool” on page 115.

Related concepts:



IBM Integration features

Some features of IBM Integration Bus have specific requirements, such as a specific mode of operation or the availability of an additional product, such as a database. Additional terms and conditions also apply to the use of some IBM Integration Bus features, including the requirement for other product licenses.

Related tasks:

“Checking the operation mode of your broker” on page 359
Run the **mqs imode** command to report the mode that is being used by your broker.

“Changing the operation mode of your broker” on page 356
Change the operation mode in which your broker is working by using the **mqs imode** command.

Related reference:

“Restrictions that apply in each operation mode”
The operation mode in which your broker is working defines how many integration servers you can use, and which nodes are available.



mqs createbroker command

Use the **mqs createbroker** command to create a broker and its associated resources.



mqs imode command

Use the **mqs imode** command to configure and retrieve operation mode information.

“License requirements” on page 641
Use the reference information in this section to understand license requirements.

Restrictions that apply in each operation mode

The operation mode in which your broker is working defines how many integration servers you can use, and which nodes are available.

Developer mode

All features are enabled for an unlimited time, but you can use the product for evaluation purposes only (within the terms of the license). IBM Integration Bus for Developers, which is also known as Developer Edition, is available on Windows x86-64 and Linux x86-64 operating systems only.

Developer Edition is restricted to processing one message per second. For more information, see “Operation modes” on page 5.

Trial mode

All features are enabled, but you can use the product for only 90 days after installation. Trial Edition is available for WebSphere Message Broker Version 8.0.0.0, and has been replaced in later versions by Developer Edition.

Standard mode

All features are enabled. You can create one integration server, but the number of message flows that you can deploy is unlimited.

Not all samples work in Standard mode because the sample import and deploy wizards create one or more integration servers, and the number of integration servers is limited in this mode.

You can run a sample in Standard mode if the default integration server is deleted after you have run the Default Configuration wizard, but before you install the sample. For example, to run the Pager sample:

1. Run the Default Configuration wizard.

2. Delete the default integration server on the IB9NODE that is created by the wizard.
3. To import and deploy the Pager sample, click Set up the Pager samples.
4. Follow the documented steps to run the Pager samples.

If you want to run samples to explore and understand the features of the product, you can install them on your development and unit test computers; see “Development and unit test” on page 9.

Express mode

The functions that are enabled are limited, and you can create only one integration server. Therefore, not all samples work in Express mode.

The following message flow nodes can be used in Express mode (plug-in nodes are also supported):

<ul style="list-style-type: none"> • AggregateControl node • AggregateReply node • AggregateRequest node • Check node • Database node • DatabaseRetrieve node • DatabaseRoute • EmailOutput node • FileInput node • FileOutput node • FlowOrder node • HTTPAsyncRequest node • HTTPAsyncResponse node • HTTPHeader node • HTTPInput node • HTTPReply node • HTTPRequest node • Input node • JavaCompute node • JMSHeader node • JMSInput node • JMSMQTransform node • JMSOutput node • JMSReceive node 	<ul style="list-style-type: none"> • JMSReply node • Label node • Mapping node • MQGet node • MQHeader node • MQInput node • MQJMSTransform node • MQOutput node • MQReply node • .NETCompute node • .NETInput node • Output node • Passthrough node • PHPCompute node • Publication node • ResetContentDescriptor node • RouteToLabel node • SOAPAsyncRequest node • SOAPAsyncResponse node 	<ul style="list-style-type: none"> • SOAPEnvelope node • SOAPExtract node • SOAPInput node • SOAPReply node • SOAPRequest node • TCPIPClientInput node • TCPIPClientOutput node • TCPIPClientReceive node • TCPIPServerInput node • TCPIPServerOutput node • TCPIPServerReceive node • Throw node • TimeoutControl node • TimeoutNotification node • Trace node • TryCatch node • Validate node • XSLTransform node
--	---	---

Scale mode

There is no limit to the number of integration servers that you can create in Scale mode, but the available nodes are limited. Therefore, not all samples work in Scale mode.

Plugin nodes can be used in Scale mode. The following message flow nodes can be used in Scale mode:

<ul style="list-style-type: none"> • AggregateControl node • AggregateReply node • AggregateRequest node • Check node • Database node • DatabaseRetrieve node • DatabaseRoute • EmailInput node • EmailOutput node • EndpointLookup node • FileInput node • FileOutput node • FlowOrder node • HTTPAsyncRequest node • HTTPAsyncResponse node • HTTPHeader node • HTTPInput node • HTTPReply node • HTTPRequest node • Input node • JavaCompute node • JDEdwardsInput node¹ • JDEdwardsRequest node¹ • JMSHeader node • JMSInput node • JMSMQTransform node • JMSOutput node • JMSReceive node • JMSReply node 	<ul style="list-style-type: none"> • Label node • Mapping node • MQGet node • MQHeader node • MQInput node • MQJMSTransform node • MQOutput node • MQReply node • .NETCompute node • .NETInput node • Output node • Passthrough node • PeopleSoftInput node¹ • PeopleSoftRequest node¹ • Publication node • RegistryLookup node • ResetContentDescriptor node • Route node • RouteToLabel node • SAPInput node¹ • SAPReply node¹ • SAPRequest node¹ • SCAAsyncRequest node 	<ul style="list-style-type: none"> • SCAAsyncResponse node • SCAInput node • SCARReply node • SCARRequest node • SiebelInput node¹ • SiebelRequest node¹ • SOAPAsyncRequest node • SOAPAsyncResponse node • SOAPEnvelope node • SOAPExtract node • SOAPInput node • SOAPReply node • SOAPRequest node • Throw node • TimeoutControl node • TimeoutNotification node • Trace node • TryCatch node • TwineballInput node • TwineballRequest node • Validate node • XSLTransform node
---	--	--

Notes:

1. A separate license for adapters is required before you can use the WebSphere Adapters nodes in Scale mode.

Remote Adapter Deployment mode

Only adapter-related features are enabled in Adapter mode, and the types of node that you can use, and the number of integration servers that you can create, are limited. You can create up to two integration servers, with no limit on the number of deployed message flows in each of these integration servers.

Not all samples work in this mode. If you want to run samples to explore and understand the features of the product, you can install them on your development and unit test computers; see “Development and unit test” on page 9.

The following message flow nodes can be used in Adapter mode (plug-in nodes are also supported):

<ul style="list-style-type: none"> • CDInput node • CDOOutput node • CICSRequest node • CORBARequest node • DatabaseInput node • EmailInput node • EndpointLookup node • FileInput node • FileOutput node • FTEInput node • FTEOutput node • HTTPAsyncRequest node • HTTPAsyncResponse node • HTTPInput node • HTTPReply node • HTTPRequest node • IMSRequest node • Input node • JavaCompute node • JDEdwardsInput node • JDEdwardsRequest node 	<ul style="list-style-type: none"> • JMSInput node • JMSOutput node • JMSReceive node • MQGet node • MQInput node • MQOutput node • .NETCompute node • .NETInput node • Output node • PeopleSoftInput node • PeopleSoftRequest node • PHPCompute node • Real-timeInput node • Real-timeOptimizedFlow node • RegistryLookup node • SAPInput node • SAPReply node • SAPRequest node • SCAAsyncRequest node • SCAAsyncResponse node 	<ul style="list-style-type: none"> • SCASInput node • SCASReply node • SCASRequest node • SecurityPEP node • SiebelInput node • SiebelRequest node • SOAPInput node • SOAPReply node • SOAPRequest node • TCPIPClientInput node • TCPIPClientOutput node • TCPIPClientReceive node • TCPIPServerInput node • TCPIPServerOutput node • TCPIPServerReceive node • TimeoutControl node • TimeoutNotification node
---	--	---

Advanced mode

All features are enabled and no restrictions or limits are imposed.

Related concepts:

“Operation modes” on page 5

The operation mode that you can use for your broker is determined by the license that you purchase.

IBM Integration features

Some features of IBM Integration Bus have specific requirements, such as a specific mode of operation or the availability of an additional product, such as a database. Additional terms and conditions also apply to the use of some IBM Integration Bus features, including the requirement for other product licenses.

Samples

The IBM Integration Toolkit provides samples that show the features that are available in IBM Integration Bus, and how to use them. This topic provides links to the information about the individual samples.

Related tasks:

“Checking the operation mode of your broker” on page 359

Run the **mqsimode** command to report the mode that is being used by your broker.

“Changing the operation mode of your broker” on page 356

Change the operation mode in which your broker is working by using the **mqsimode** command.

Deciding which nodes to use

IBM Integration Bus includes many message processing nodes that you can use in your message flows.

Creating a message flow

Create a message flow to specify how to process messages in the broker. You can create one or more message flows and deploy them to one or more brokers.

Related reference:



What the Default Configuration wizard creates

A table of the components that are created by the wizard, details of how to resolve problems, and how to view errors.

Planning an IBM Integration Bus environment

When you start to plan your IBM Integration Bus environment, you must first consider the design of your system infrastructure. A major consideration when you are planning to use IBM Integration Bus is how it will fit into your overall system architecture, and it is important to understand the movement of data around your whole system.

About this task

It is also important to understand the high-level goals of the system that you are building, and to be clear about the functional requirements. For example, IBM Integration Bus has extremely powerful routing and transformation capabilities, but it is not designed to function as an application server whose primary function is data processing that involves a high number of complex calculations.

Another important consideration is whether you require a highly available (HA) system that can withstand the failure of an individual execution group or integration node. For more information about high availability, see “Configuring for high availability” on page 544.

The capability and capacity of your brokers is partly determined by the operation mode in which the brokers are running, and this is determined by the IBM Integration Bus license that you have purchased. For example, Express Edition entitles you to run a broker in Express mode, in which a subset of capabilities are enabled for use with a single integration server, whereas the full edition enables you to run brokers in Advanced mode with all features enabled and with no limits to the number of resources that you can create and maintain. For more information, see IBM Integration features and “Operation modes” on page 5.

The following topics describe these factors:

- “Cloud overview” on page 22
- “Planning the high-level infrastructure” on page 15
- “Choosing your IBM Integration Bus edition and operation mode” on page 4
- “Performance planning” on page 27
- “Planning for security” on page 30
- “Considering resource naming conventions” on page 18

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.



Routing using publish/subscribe applications

You can route your messages to applications using the publish/subscribe method of messaging.

Related tasks:

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your

choice.

“Performance planning” on page 27

When you design your IBM Integration Bus environment and the associated resources, the decisions that you make can affect the performance of your applications.

“Configuring databases” on page 361

Configure databases to hold application or business data that you can access from your message flows.

Planning the high-level infrastructure

When you are planning to use IBM Integration Bus, consider your complete system of products as a whole and ensure that the flow of data throughout the system is carefully planned.

About this task

It is important to understand how IBM Integration Bus will fit into the overall system of products in your business, and to be clear about the movement of data around the whole system. It is also important to understand the high-level goals of the system and your functional requirements, so that you can ensure that you are planning to use IBM Integration Bus in a way that will gain the maximum benefit. For example, IBM Integration Bus has extremely powerful routing and transformation capabilities, but it is not designed to function as an application server whose primary function is data processing with a high number of complex calculations. Therefore, it is important to fully understand the role of the product within your overall IT system, in order to use it to its full advantage.

The following topics describe some additional factors to consider when planning your implementation of IBM Integration Bus:

- “Designing the WebSphere MQ infrastructure” on page 16
- “WebSphere MQ resources for brokers” on page 17
- “Considering resource naming conventions” on page 18
- “Performance planning” on page 27
- “Planning for security” on page 30

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.

“Performance planning” on page 27

When you design your IBM Integration Bus environment and the associated resources, the decisions that you make can affect the performance of your applications.

“Configuring databases” on page 361

Configure databases to hold application or business data that you can access from your message flows.

Designing the WebSphere MQ infrastructure

You must create and manage the WebSphere MQ resources that are required to support your brokers, and the applications that connect to them to supply or receive messages.

Brokers

A broker uses WebSphere MQ messages to provide information, status, and instructions about its internal operations. Connections are also required by each IBM Integration Toolkit with each broker with which it communicates.

Some of these resources that are required are created for you when you create brokers and integration servers. The requirements associated with the broker are described in “WebSphere MQ resources for brokers” on page 17.

Applications and message flows

Your applications exchange messages and other data by communicating with message flows that are running in the broker. You can connect your applications to the broker by using one of the supported communications methods. If your applications are written to use WebSphere MQ, the requirement for the channels or client connections are determined by the types of nodes that you include in your message flows. These resources are application-specific, and you must create these resources yourself.

The following nodes might require WebSphere MQ resources:

- MQInput and MQOutput
- MQReply and MQGet

For more information about creating resources, see the *Intercommunication* section of the WebSphere MQ Version 7 product documentation online.

Related concepts:

 The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

 Connecting client applications

Connect your client applications to the broker by using one or more of the supported protocols from other resources and software servers in your network.

 Developing integration solutions

IBM Integration Bus provides a flexible environment in which you can develop integration solutions to transform, enrich, route, and process your business messages and data. You can integrate client applications that use different protocols and message formats.

Related reference:



Message flows

Use the reference information in this section to develop your message flows and related resources.

Related information:



WebSphere MQ Version 7 product documentation

WebSphere MQ resources for brokers:

Each broker depends on a number of WebSphere MQ resources: some are required, others are optional, and depend on your environment and requirements. Some of these resources are created for you, but others you must define for yourself.

WebSphere MQ resources created for you

When you create a broker, the following WebSphere MQ resources are created for you:

- On distributed systems, the queue manager for the broker. Each broker must be associated with a dedicated queue manager. Specify a queue manager name when you create the broker. If this queue manager does not exist, it is created for you.
- Fixed-name queues on the queue manager that hosts this broker. The broker uses these queues to send messages to provide information, status, and instructions about its internal operations

WebSphere MQ resources that you must create yourself

Depending on the setup of your broker, you might need to create some WebSphere MQ resources yourself. You might need some or all the following resources:

- If you create a broker on z/OS, you must create the queue manager. See “Creating a broker on z/OS” on page 313 for more details.
- Define listener connections on the broker queue manager. You must define one listener connection for every protocol that your applications use; for example, TCP/IP.

Create WebSphere MQ resources by using one of the following commands and utilities:

- **runmqsc**
- The PCF interface
- WebSphere MQ Explorer
- IBM Integration Explorer

For more information about creating resources, see the *Intercommunication* section of the WebSphere MQ Version 7 product documentation online.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

| **Related reference:**

| “Designing the WebSphere MQ infrastructure” on page 16
| You must create and manage the WebSphere MQ resources that are required to
| support your brokers, and the applications that connect to them to supply or
| receive messages.

| **Related information:**

|  [WebSphere MQ Version 7 product documentation](#)

| **Considering resource naming conventions:**

| When you plan a new IBM Integration Bus network, one of your first tasks must
| be to establish a convention for naming the resources that you create within this
| network.

| Information is provided about the following resources:

- Product component naming conventions
- WebSphere MQ naming conventions
- Database naming conventions

| **Related reference:**

| “Designing the WebSphere MQ infrastructure” on page 16
| You must create and manage the WebSphere MQ resources that are required to
| support your brokers, and the applications that connect to them to supply or
| receive messages.

| *Naming conventions for brokers and associated resources:*

| Establish a naming convention for all the IBM Integration Bus resources in your
| broker environment to ensure that names are unique, and that users creating new
| resources can be confident of not introducing duplication or confusion.

| Consider the names that you use for your brokers and resources:

| **Brokers**

| When you create a broker, give it a name that is unique within your broker
| environment. You must use the same name for that broker when you create
| it on the system in which it is installed, using the command
| **mqsicreatebroker**. Broker names are case sensitive except on Windows
| platforms.

| **Integration servers**

| Each integration server name must be unique within a broker.

| **Message flows and message processing nodes**

| Each message processing node must be unique within the message flow it
| is assigned to. For example, if you include two MQOutput nodes in a
| single message flow, provide a unique name for each one.

| Message flow names must be unique within the broker. All references to
| that name are always to the same message flow. If you assign the same
| message flow to more than one broker, you must ensure that you maintain
| unique message flow names across those brokers.

| **Message sets and messages**

Each message name must be unique within the message set to which it belongs.

Message set names must be unique within the broker. All references to that name are always to the same message set. If you assign the same message set to more than one broker, you must ensure that you maintain unique message set names across those brokers.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

Related reference:



mqsicreatebroker command

Use the **mqsicreatebroker** command to create a broker and its associated resources.



Naming conventions for IBM Integration Bus for z/OS

Decide upon a naming convention for your IBM Integration Bus for z/OS broker to make customizing, operating, and administering easier.

WebSphere MQ naming conventions:

All IBM Integration Bus resources have dependencies on WebSphere MQ services and objects. You must therefore also consider what conventions to adopt for WebSphere MQ object names. If you already have a WebSphere MQ naming convention, use a compatible extension of this convention for IBM Integration Bus resources.

When you create a broker, you must specify a queue manager name. This queue manager is created for you if it does not already exist. Every broker must have a dedicated queue manager.

Ensure that every queue manager name is unique within your network of interconnected queue managers, whether or not every queue manager is in your IBM Integration Bus network. This ensures that each queue manager can unambiguously identify the target queue manager to which any given message must be sent, and that IBM Integration Bus applications can also interact with basic WebSphere MQ applications.

WebSphere MQ supports a number of objects defined to queue managers. These objects (queues, channels, and processes) also have naming conventions and restrictions.

In summary, the restrictions are:

- All names must be a maximum of 48 characters in length (channels have a maximum of 20 characters).
- The name of each object must be unique within its type (for example, queue or channel).

- Names for all objects starting with the characters SYSTEM. are reserved for use by IBM.

Additionally, there is a restriction to the length of the user identifier on each platform:

- On AIX®, UNIX systems, and z/OS, the maximum length is eight characters.
- On Windows, the maximum length is 20 characters.

There are a few restrictions for naming resources: see Naming conventions for IBM Integration Bus for z/OS.

Related tasks:

“Creating a broker” on page 304

You can create brokers on every platform that is supported by IBM Integration Bus. The broker runs as a 64-bit application on all platforms except Linux on x86 and Windows on x86.

Related reference:



Naming conventions for IBM Integration Bus for z/OS

Decide upon a naming convention for your IBM Integration Bus for z/OS broker to make customizing, operating, and administering easier.

Database naming conventions:

Consider the naming conventions you use for databases that you create for application use.

Ensure that the databases that you use for application data, which are accessed through your deployed message flows, are uniquely named throughout your network, so that confusion and errors are avoided by all your users.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.

Business planning

Consider the role of IBM Integration Bus in your overall business solution, and understand the potential benefits and limitations based on your functional and business requirements.

About this task

It is important to understand the capability and capacity that you require from your system. The capability and capacity of your integration nodes are partly determined by the operation mode in which they are running, and this is determined by the IBM Integration Bus license that you have purchased. For example, Express Edition entitles you to run a broker in Express mode, in which a subset of capabilities are enabled for use with a single integration server, whereas the full edition of the product enables you to run in Advanced mode with all features enabled and with no limits to the number of resources that you can create

and maintain. Alternatively, you might choose to use the Developer Edition, which you can download at no charge and use for evaluative purposes. For more information about the features and modes of operation provided in each edition, see “Choosing your IBM Integration Bus edition and operation mode” on page 4.

It is also important to consider the most suitable technology for your business; this varies between projects and organizations, and there are various factors to consider when assessing your requirements:

- The pool of development skills available. For example, if your developers are already skilled in the use of ESQL, you might want to continue using this transformation technology. However, if you have strong Java skills, the JavaCompute node might be the better choice, rather than having to teach your Java developers ESQL.
- The ease with which you can teach developers a new development language. For example, if you have many developers it might not be viable to teach all of them ESQL, and you might choose instead to educate only a few people who can then develop common functions or procedures.
- The skills of individual people. If your developers have relatively little experience with IBM Integration Bus, you might find that the graphical interface provided by the Mapping node is more suitable than other transformation technologies.
- Asset reuse. If you are an existing user of IBM Integration Bus, you might already have large amounts of ESQL code that you would like to reuse. You might also have other resources that you want to reuse, such as Java classes for key business processing, style sheets, or mappings. You might want to use these as the core of message flow processing and extend the processing with one of the other technologies.
- Performance. If message throughput is a prime consideration, you might choose to use ESQL or Java. For more information about designing your system for optimum performance, see Message flow design and performance and Code design and performance.
- Solution scalability, cost and flexibility. You might have an existing network infrastructure that you want to use, or you might want to take advantage of IBM Integration Bus cloud capabilities. For more information on the benefits of cloud, see “Cloud overview” on page 22.

For more information about factors to consider when planning your IBM Integration Bus solution, see the following topics:

- “Planning an IBM Integration Bus environment” on page 14
- “Planning the high-level infrastructure” on page 15
- “Understanding your processing requirements” on page 2
- “Performance planning” on page 27
- “Planning for security” on page 30

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.

“Performance planning” on page 27

When you design your IBM Integration Bus environment and the associated resources, the decisions that you make can affect the performance of your applications.

“Configuring databases” on page 361

Configure databases to hold application or business data that you can access from your message flows.

“Planning for security” on page 30

It is important to be aware of security issues when you are planning to install and use IBM Integration Bus.

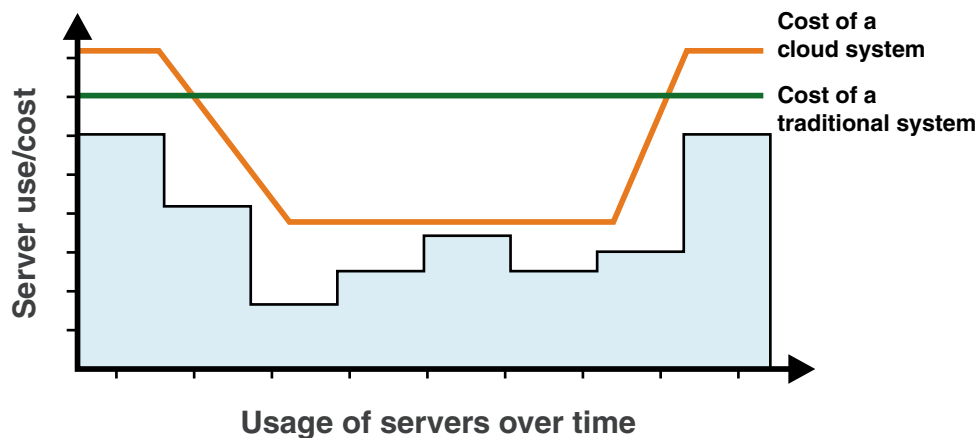
Cloud overview

When you are deciding on your environment for IBM Integration Bus, consider your requirements and how you might want to implement a cloud solution.

IBM Integration Bus Version 9.0 introduces the option of installing IBM Integration Bus in the cloud.

Depending on your requirements, you might benefit from reduced operating costs and deployment flexibility for IBM Integration Bus when compared to a static installation.

Using a cloud vendor usually has a higher operating cost than using a static network-based solution, but less upfront capital is required and parts of the infrastructure are automated. Review the following graph as an example of how this can reduce costs overall:



While the cost equivalent in computing power is slightly higher than using your own server, you pay for as much resource as you use, thereby reducing the average cost over a period of time. There are many different financial models for cloud vendors that offer different services.

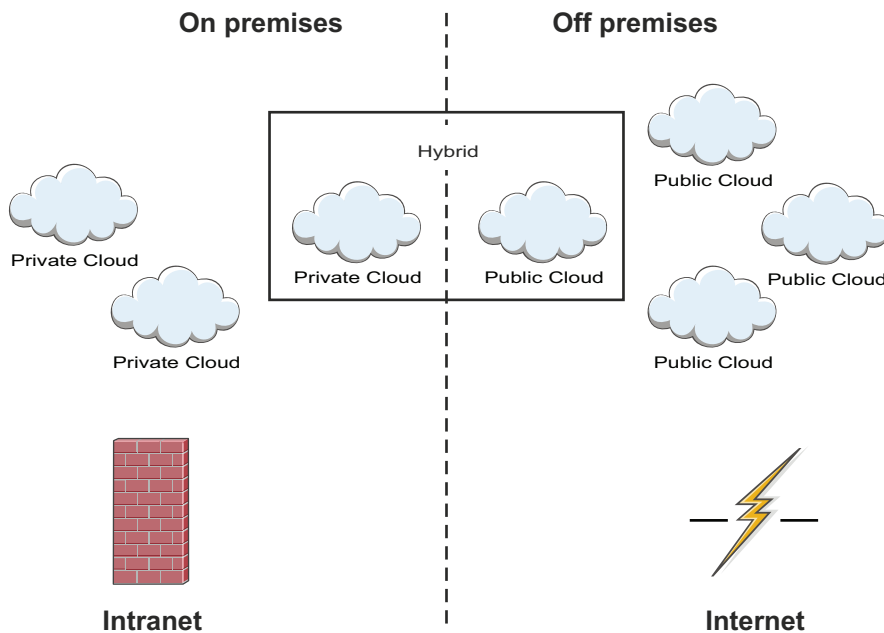
You can use your own cloud solution, if you have the appropriate skills within your organization to ensure maximum cost-efficiency of your resource use.

Types of cloud

You can choose how much control you retain over your cloud installation by deciding whether you want your solution to be *on premises* or *off premises*, and whether that cloud is public or private.

- An on-premises (or private) cloud is accessed and used by solely one organization. If you use a private cloud, you maintain absolute control of the maintenance and configuration, but setting up a private cloud requires significant technical skills, scrutiny of security implementation, time, and resources.
- An off premises (or public) cloud is a cloud that is available to multiple organizations, who each use parts of the cloud privately or publicly. There is little to no architectural difference to public or private clouds, except for extra and different security considerations for your services. The maintenance, configuration, and security of the hosting servers are controlled by the hosting vendor.
- A hybrid cloud service contains both on premises and off premises elements. For example, you might host client-sensitive data on premises in a private cloud, but connect to an application that is hosted in a public cloud that processes some of that data.

See the following diagram for a visual representation of the relationship between these concepts:



You can use IBM Integration Bus in any of these configurations. For example, you might have a traditional installation of IBM Integration Bus privately on premises, but deploy your integration nodes in the cloud.

Cloud technologies

By using cloud technologies, you can choose to abstract parts of your network setup so that the physical components are instead virtual. You can then manipulate

the non-physical components to meet the needs of your solution without the associated cost overheads of a physical change, such as changing physical servers or operating system.

This abstraction is collectively known as *XaaS* or *Everything as a Service*, and you can create any part of XaaS at different levels of system architecture.

For a cloud installation of IBM Integration Bus, you can benefit from:

- Infrastructure as a Service (IaaS). IaaS is used and maintained by system architects, or the equivalent in your organization.
- Platform as a Service (PaaS). PaaS is used and maintained by application developers, or the equivalent in your organization.

Review these topics to learn about each level of XaaS and the advantages of using each layer with IBM Integration Bus.

Note: Each level of XaaS is independent. You do not have to use any of these concepts together in an installation.

You can still install IBM Integration Bus by using the provided installation images to a private file system. You might want to do this if you have a small-scale solution that is unlikely to change in size. To review what is required for a standard installation, see “Installing” on page 31.

Related concepts:

“Cloud overview” on page 22

When you are deciding on your environment for IBM Integration Bus, consider your requirements and how you might want to implement a cloud solution.



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Infrastructure as a Service”

Learn what Infrastructure as a Service (IaaS) is, and how you might benefit from using IaaS with IBM Integration Bus.

“Platform as a Service” on page 26

Learn what Platform as a Service (PaaS) is, and how you might benefit from using PaaS with IBM Integration Bus.

Related tasks:

“Business planning” on page 20

Consider the role of IBM Integration Bus in your overall business solution, and understand the potential benefits and limitations based on your functional and business requirements.

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.

“Performance planning” on page 27

When you design your IBM Integration Bus environment and the associated resources, the decisions that you make can affect the performance of your applications.

Infrastructure as a Service

Learn what Infrastructure as a Service (IaaS) is, and how you might benefit from using IaaS with IBM Integration Bus.

Infrastructure as a Service acts as the equivalent to computing hardware by using virtual machines. An IaaS configuration reduces the cost overhead of managing many servers, whether on premises or off premises, as you do not need to manually configure each system. For example, you can apply security patches to all virtual machines simultaneously by using a script.

The advantage of using IaaS with IBM Integration Bus is to create a scalable solution that does not require increasing configuration as it grows in size. By configuring scripts, you can scale and change attributes as required, instead of manually recoding the system infrastructure.

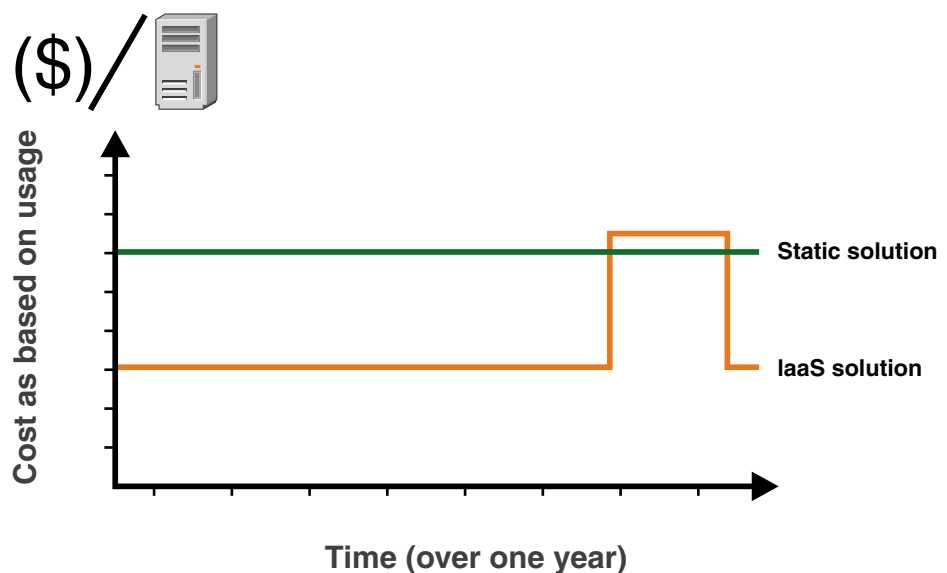
You can use IaaS on premises or off premises, but the primary benefit of IaaS is the ability to use a vendor to reduce the costs of maintaining and configuring the physical machines. If you want to use IaaS on premises, consider the skill availability of system architects in your organization to configure your architecture, and the associated costs.

The benefits of using an IaaS configuration are:

- Flexible (or elastic) hosting. An IaaS provider can provision new virtual machines for you quickly.
- If you use an IaaS provider, remove the responsibility and associated effort of applying fixes, security patches, and other upgrades.
- Removes or reduces capital expenditure on hardware and human resources.

Considerations for using IaaS are:

- If you use IaaS without PaaS, the lack of dynamic scalability: you must predict your peak usage in advance.
- Choosing an IaaS vendor with the right cost model for your usage.
- The capability of the vendor to meet your ongoing business requirements.
- Security and data compliance, if you have restrictions for data use and sharing.
- Dependency on the IaaS vendor for uptime and potentially recovery.
- Requires new and different security measures.



Related concepts:

“Chef”

Chef software is an open source configuration management tool that you can use to create parts of an Infrastructure as a Service (IaaS). You can use Chef scripts to provision an IBM Integration Bus installation.

Chef:

Chef software is an open source configuration management tool that you can use to create parts of an Infrastructure as a Service (IaaS). You can use Chef scripts to provision an IBM Integration Bus installation.

Chef scripts, which are called *recipes*, are made of reusable definitions that are written in the Ruby programming language. Chef recipes that perform related functions are grouped in a single container, called a cookbook. Cookbooks and recipes automate common infrastructure tasks. The recipe definitions describe what your infrastructure consists of, and how each part of your infrastructure is deployed, configured, and managed. Chef applies these definitions to servers, to produce an automated infrastructure.

By using Chef, you can create scalable infrastructure with minimal configuration to maintain. Chef scripts form part of a recipe for your infrastructure configuration that is understood by multiple systems, services, and languages. By using Chef to configure your infrastructure deployment, you do not have to rewrite code if you change part of the underlying infrastructure, such as changing data service provider.

To learn more about Chef concepts, see the online Chef documentation [About Cookbooks - Chef Docs](#).

Related information:

 [Chef Documentation - About Cookbooks](#)

Platform as a Service

Learn what Platform as a Service (PaaS) is, and how you might benefit from using PaaS with IBM Integration Bus.

A PaaS cloud configuration automates the execution environment and solution deployment. For example, acting as an operating system that supports IBM Integration Bus. A typical PaaS provider consists of many servers that each provide customized environments, and can dynamically allocate resources to that environment where and when they are required.

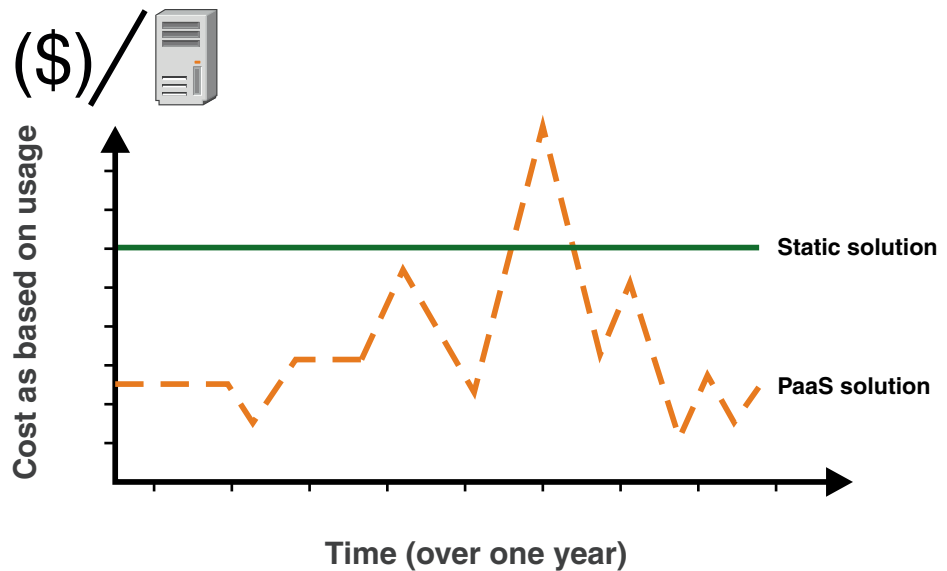
Applications are deployed into this environment, and are often charged by usage or capacity, which means that you pay only for as much as you use or specify, which can greatly reduce operating costs.

The benefits of using a PaaS configuration are:

- Reconfiguring and maintenance of the environment is faster and requires significantly less effort.
- If you use a PaaS provider, maintenance and optimization are provided, requiring less developer time.
- Dynamically scalable. PaaS automatically increases or decreases resources, which are based on demand. You usually pay only for the resources that are used.

Considerations for using PaaS are:

- Choosing a PaaS vendor with the right cost model for your usage.
- Whether the PaaS provider has any restrictions on what type of content and applications are allowed.
- What frameworks, languages and databases are supported.
- Vendor lock-in, and choosing standard or proprietary technologies. If your provider uses proprietary technologies, you might find it harder to switch vendors.
- Security and data compliance, if you have restrictions for data use and sharing.
- Dependency on the PaaS vendor for aspects of security and uptime.
- Requires new and different security measures.



Performance planning

When you design your IBM Integration Bus environment and the associated resources, the decisions that you make can affect the performance of your applications.

About this task

If you are planning to create and maintain a small, stand-alone system with limited requirements for availability and performance, it is possible to achieve this relatively quickly. However, if you are planning a large or complex system, or if you have specific requirements for high availability or performance, it is worth taking time to understand the factors that can influence your design and the techniques that you can use to optimize it. As a starting point, consider the following aspects of your system and understand how these factors can influence performance:

Message flows

A message flow includes an input node that receives a message from an application over a particular protocol; for example, WebSphere MQ. The message must be parsed by the input node, and the performance impact of this parsing varies according to the parser that is used and the number of parses required. You can reduce the impact of parsing by using some

optimization techniques such as parsing avoidance or partial parsing. For more information about parsing, see Parsing and message flow performance.

The number and length of message flows have implications for performance, and it is important to keep the number of nodes to a minimum. See Message flow design and performance for more information about optimizing message flow performance. Other aspects of processing in a message flow that might affect performance are the amount, efficiency, and complexity of ESQL, access to databases, and how many message tree copies are made. For more guidance about these factors, see Code design and performance.

It is also important to consider how you split your business logic; how much work should the application do, and how much should the message flow do? Every interaction between an application and a message flow involves I/O and message parsing, and therefore adds to processing time. Design your message flows, and design or restructure you applications, to minimize these interactions.

For more information about these factors, see Optimizing message flow response times and Optimizing message flow throughput.

Messages and message models

The type, format, and size of the messages that are processed can have a significant effect on the performance of a message flow. For example, if you process persistent messages, they have to be stored for safekeeping.

You might need to process messages with a well-defined structure; if so, you can create DFDL models for your messages. If you do not plan to interrogate the structure, you can work with undefined messages, such as BLOB messages.

If you are working in XML, be aware that it can be verbose, and therefore produce large messages, but XML message content is easier to understand than other formats, such as CWF. Field size and order might be important; these factors can be included in your DFDL model.

For more information about these factors, see Optimizing message flow response times and Performance considerations for regular expressions in TDS messages.

Broker configuration

You can create and configure one or more brokers, on one or more computers, and for each broker you can create multiple integration servers, and multiple message flows. Your configuration decisions can influence message flow performance, and how efficiently messages can be processed.

For more information about these factors, see Tuning the broker, and Optimizing message flow throughput.

All these factors are examined in more detail in the Designing for Performance SupportPac (IP04).

For a description of common performance scenarios, review Troubleshooting performance problems.

For further articles about IBM Integration Bus and performance, review these sources:

- The Business Integration Zone on developerWorks®. This site has a search facility; enter "performance" and review the links that are returned.
- The developerWorks article on message flow performance.
- The developerWorks article on WebSphere Message Broker Explorer accounting and statistics.
- The developerWorks article on monitoring resource use.

Related tasks:



Message flow design and performance

Consider the performance implications arising from the design of your message flow.



Optimizing message flow throughput

Each message flow that you design must provide a complete set of processing for messages received from a particular source. This design might result in very complex message flows that include large numbers of nodes that can cause a performance overhead, and might create potential bottlenecks. You can increase the number of message flows that process your messages to provide the opportunity for parallel processing and therefore improved throughput.



Optimizing message flow response times

You can use different solutions to improve message flow response times.



Setting configuration timeout values

Change timeout values that affect configuration tasks in the broker.



Tuning the broker

You can complete several tasks that enable you to tune different aspects of the broker performance.



Resolving problems with performance

Use the advice given here to help you to resolve common problems with performance.



Analyzing message flow performance

You can configure your broker to collect statistics about the operation and behavior of your message flows. Use this information to assess the performance of your message flows.



Tuning message flow performance

You can configure your message flows to enhance their performance in terms of resource usage, message flow throughput, and response times.



Analyzing resource performance

You can collect statistics to assess the performance of resources used by integration servers.



Troubleshooting performance problems

Follow this guidance to resolve common problems with IBM Integration Bus performance.

Related reference:



Message Sets: Performance considerations when using regular expressions

Take care when specifying regular expressions: some forms of regular expression can involve a large amount of work to find the best match, which might degrade

| performance.

| **Planning for security**

| It is important to be aware of security issues when you are planning to install and
| use IBM Integration Bus.

| **About this task**

| On Linux and UNIX systems, you must complete security tasks before you install
| IBM Integration Bus. On Windows and z/OS systems, security tasks are completed
| during and after installation.

| After installation, refer to one of the following documents for further security
| considerations:

- | • For Windows, Linux and UNIX systems, see [Creating user IDs](#).
- | • For z/OS systems, see [Configuring broker user IDs on z/OS](#).

| For an introduction to various aspects of security, see [Security overview](#).

| **Related reference:**

| Installation

| Installation information for IBM Integration Bus is provided in the online
| information center.

Chapter 2. Installing and uninstalling

Install and uninstall IBM Integration Bus components and service.

Find out where you can see the latest information about installation of IBM Integration Bus components, and how to apply service. You can also view information about complementary products. This section also describes how to remove service (not available on all operating systems), and how to uninstall components.

- “Installing”
- “Uninstalling” on page 150

Related tasks:

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

Chapter 3, “Migrating,” on page 171

To migrate a broker domain to IBM Integration Bus Version 9.0, plan your migration strategy, perform pre-migration tasks, migrate your domain components, then complete post-migration tasks.

Related reference:

“Installation,” on page 619

Use the reference information in this section to understand installation requirements, installation options, and how they affect your computer.

Related information:

Installing and Migrating (EPUB)

Installing and Migrating (Kindle MOBI)

Installing

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

About this task

See “Finding the latest information” on page 1 for details about how to check that you have access to the most recent information available.

The “Installation Guide” on page 32 defines hardware and software requirements for IBM Integration Bus and its corequisite and prerequisite products. It describes the tasks that you must complete to prepare for installation, to install IBM Integration Bus, and to verify the installation. When you have completed installation, use this information center to create and configure your components and brokers.

If product fixes or updates are made available, refer to the following topics for information about how to apply these changes:

- “Applying service to the Integration Bus component” on page 130
- “Applying service to the IBM Integration Toolkit” on page 141
- “Applying service to the IBM Integration Explorer” on page 146

- “Applying service to IBM Integration Administration for WebSphere Application Server” on page 147

You do not have to apply service to the runtime components and the IBM Integration Toolkit at the same time; all Version 9.0 fix pack levels are compatible with all other fix pack levels. However, if a fix pack delivers additional nodes, you must ensure that the IBM Integration Toolkit and the brokers to which you deploy the message flows that include these nodes are at the same fix pack level, and that you have set the correct function level of those brokers by using the **-f** flag on the **mqsichangebroker** command.

To install complementary products, see “Installing complementary products” on page 114.

Related tasks:

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.



Resolving problems when installing

Work through the advice provided to help you to deal with problems that can arise when the product is installed.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Installation,” on page 619

Use the reference information in this section to understand installation requirements, installation options, and how they affect your computer.

“Support for 32-bit and 64-bit platforms” on page 625

IBM Integration Bus operates in 32-bit mode or 64-bit mode, on supported operating systems.

Related information:



WebSphere Message Broker Library web page

Installation Guide

Installation information for IBM Integration Bus is provided in the online information center.

See “Installation,” on page 619 for reference information about system requirements, including operating systems, hardware, and additional software and memory requirements.

The Installation and Installing topics provide general preparation, planning, and security information for all platforms. The topics also provide installation instructions for distributed systems, common problems and solutions, and the names and locations of the installation log files. Verification programs on Linux on x86 and Windows are also provided.

- If you want to establish a simple installation on a Windows system, and you have only a primary MQ installation, run the Launchpad; see “Installing by using the Windows Launchpad” on page 60.
- If you want to install a more complex setup with components in nondefault locations, or to install multiple copies at different levels, run the individual component installers:
 - “Installing the Integration Bus component” on page 69
 - “Installing the IBM Integration Toolkit” on page 79
 - “Installing IBM Integration Explorer” on page 84
- If you want to perform a silent installation of all components on a Linux or UNIX system, see “Installing IBM Integration Bus components and prerequisite products by using the sample script file” on page 65.
- If you are installing on z/OS, the equivalent installation instructions are provided in the Program Directory that is supplied in hardcopy with your product media.
- If you are a WebSphere Application Server user, and want to manage remote integration nodes, see “Installing IBM Integration Administration for WebSphere Application Server” on page 93.

The Installation and Installing topics do not cover configuration or customization tasks on any operating system; they describe only how you install the product components onto your computers. When you have completed installation on distributed systems, you must initialize the local environment before you can create or configure resources. This task is described in “Setting up a command environment” on page 119. Environment initialization is not required on z/OS. For information about designing and configuring your broker environment on all systems, see Chapter 5, “Configuring brokers for test and production environments,” on page 289.

The topics within the Installation and Installing sections of the online information center are also available in English and supported languages within the downloadable version of the information center from the IBM Integration Bus Library web page.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.



Resolving problems when installing

Work through the advice provided to help you to deal with problems that can arise when the product is installed.

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

“Applying service to the Integration Bus component” on page 130

Apply maintenance updates and program fixes to the Integration Bus component.

“Applying service to the IBM Integration Toolkit” on page 141

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

“Applying service to the IBM Integration Explorer” on page 146

You can apply maintenance or fixes to the IBM Integration Explorer.

“Applying service to IBM Integration Administration for WebSphere Application Server” on page 147

Apply maintenance updates and program fixes to the IBM Integration Administration for WebSphere Application Server feature.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.



Resolving problems when uninstalling

Work through the advice provided to help you to deal with problems that can arise when the product is uninstalled.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Installation,” on page 619

Use the reference information in this section to understand installation requirements, installation options, and how they affect your computer.

Preparing for installation

Use the instructions in this tutorial for an overview of the installation tasks, and to prepare for installation of IBM Integration Bus.

About this task

The tasks that you perform to complete installation are listed here; each task indicates whether it is required or optional. A summary of each task is provided, along with pointers to later chapters or sections that describe the task in more detail.

Procedure

1. Required: Make sure that you have acquired the product packages that you need for installation.

Physical and electronic packages are available for IBM Integration Bus. DVDs are supplied in the physical package, and electronic images are available from IBM Passport Advantage®.

For more information about available packages and their contents, see “Installation packages” on page 645. If you have ordered the product for electronic delivery from IBM Passport Advantage, check that you have downloaded all the images that you need for all components and all platforms.

Find out how to access, download, and extract product images in “Accessing DVDs” on page 51.

For instructions about downloading and applying service updates, see “Applying service” on page 129.

2. Required: Make sure that you have access to the product documentation that you need for installation.
 - The product readme file `readme.html` contains the latest available information.
 - The information center describes planning and preparation on all platforms.

- This information center describes installation procedures for all components on all distributed platforms.
- The *Program Directory for IBM Integration Bus for z/OS* describes installation procedures for all components on z/OS.

Find out how to get these documents in “Finding the latest information” on page 1.

3. Required: Decide which components you want to install on your computers. The following information provides a minimum level of detail about IBM Integration Bus components; read “Choosing what to install” on page 59 to find out more about the components.

- The IBM Integration Toolkit. You must install this component on at least one computer. You can install this component only on Windows 32-bit, Windows 64-bit, Linux on x86, and Linux on x86-64.

Use the IBM Integration Toolkit to create, manage, deploy, and delete message flows and associated resources in a development environment.

- The IBM Integration Explorer. You can install this component only on Windows 32-bit, Windows 64-bit, Linux on x86, and Linux on x86-64.

Use the IBM Integration Explorer to administer your brokers in a production environment.

- The Integration Bus component. You must install this component on at least one computer.

You can create multiple brokers on a single computer. Deploy your message flow resources to one or more brokers to process your application messages.

4. Required: Decide which platform you want to install your chosen components onto.

The following table shows the components that you can install on the supported platforms.

Table 2. Summary of component and platform support

Component	Platform
IBM Integration Toolkit	<ul style="list-style-type: none"> • Linux on x86 • Linux on x86-64 • Windows 32-bit • Windows 64-bit
IBM Integration Explorer	<ul style="list-style-type: none"> • Linux on x86 • Linux on x86-64 • Windows 32-bit • Windows 64-bit
Integration Bus component	All supported platforms

5. Required: Prepare each computer on which you are installing one or more components.

- a. Check that your target computers meet the initial hardware, storage, and software requirements.

The requirements vary depending on what computers you want to install IBM Integration Bus on, and what components you are installing; read the details in “Hardware requirements” on page 620 and “Operating system requirements” on page 627.

The supported hardware and software environments are updated occasionally. To view the latest requirements, see the product requirements website:

www.ibm.com/software/integration/wbimessagebroker/requirements/

- b. Complete the appropriate tasks for your computer to set up security and UNIX kernel configuration, and get ready to access the installation media. All these tasks are described in “Preparing the system” on page 43.
- c. Check that your user ID has the required authority to complete installation:
 - AIX systems: Use the user ID root to perform the preparatory tasks that are described in “Security on Linux and UNIX systems” on page 45. When these steps have been completed, a user without root authorities can complete the installation.
 - Linux and other UNIX systems: Your user ID must have root authorities to perform the preparatory tasks that are described in “Security on Linux and UNIX systems” on page 45. When these steps have been completed, a user without root authorities can complete the installation.
 - Windows systems: Your user ID must be a member of the group Administrators.
 - z/OS systems: Use a user ID that has suitable RACF® privileges to perform the installation.

This information is a summary only; more details are provided in “Setting up security” on page 44.

6. Required: Check the programs that you use to perform component installation.

The following table lists the programs that are available.

Table 3. Summary of available installation methods

Platform	Tools
Windows only	<p>The Windows Launchpad.</p> <p>This program installs prerequisite products if they are not already installed, and identifies prerequisite products that are not at the supported level. See “Installing by using the Windows Launchpad” on page 60.</p>
Linux, UNIX, and Windows systems	<p>Installation wizards on each supported platform have unique names, which are listed in “Installation wizard names” on page 660.</p> <ul style="list-style-type: none"> • To install the Integration Bus component, see “Installing the Integration Bus component” on page 69. • To install the IBM Integration Toolkit, see “Installing the IBM Integration Toolkit” on page 79. • To install the IBM Integration Explorer, see “Installing IBM Integration Explorer” on page 84. <p>You can install IBM Integration Bus components and prerequisite products in silent mode on Linux, UNIX, and Windows. For more information, see “Installing IBM Integration Bus components and prerequisite products by using the sample script file” on page 65.</p> <p>If you are installing the IBM Integration Toolkit on a Red Hat Enterprise Linux 6.0 x86_64 (64-bit) computer, you must install the 32-bit versions of operating system libraries before you run the IBM Installation Manager. For full instructions, see the following technote on the IBM Support Portal: Unable to install Installation Manager on RHEL 6.0 (64-bit).</p>
z/OS only	<p>SMP/E</p> <p>To install runtime components, see the <i>Program Directory for IBM Integration Bus for z/OS</i>.</p>

7. Required: Install additional products that are required by IBM Integration Bus. IBM Integration Bus requires other software products to work successfully. The order in which you install these products is not important. However, you must install all required products before you can configure and start IBM Integration Bus components.

The following table gives a summary of these requirements.

Table 4. Summary of prerequisite products

Component	Prerequisite products
IBM Integration Toolkit	<ul style="list-style-type: none"> • A web browser to view the information center.
IBM Integration Explorer	<ul style="list-style-type: none"> • Install WebSphere MQ Explorer. • A web browser to view the information center.
Integration Bus component	<ul style="list-style-type: none"> • WebSphere MQ to communicate with other components. • A Java™ Runtime Environment (JRE), if using z/OS.

Full details of all these requirements are provided in “Additional software requirements” on page 633:

- For web browsers, see “Browsers” on page 635.
- For details of supported versions of WebSphere MQ, see “WebSphere MQ” on page 633.
- For more information about JREs, see “JRE” on page 634.

8. Optional: Configure a minimum broker domain and verify its operation.

- To create a minimum broker domain, install the IBM Integration Toolkit and the Integration Bus component on a single computer. Because the IBM Integration Toolkit is required to complete this task, you must choose a Linux on x86, Linux on x86-64, or Windows computer.
- Use the Default Configuration wizard, which you can start from the IBM Integration Toolkit, to create the required components after installation.
- Use this broker domain to create message flow resources, verify your installation, investigate how the product works, and explore the product samples.

The verification process is described in “Verifying your IBM Integration Bus installation” on page 104.

9. Optional: Change the broker operation mode.

When you install IBM Integration Bus and create brokers, they are configured with an operation mode set either to `developer` (if you have installed the Developer Edition) or `advanced` (all other editions). You must configure your brokers to conform to the license that you have purchased. Therefore, if you have purchased the Standard Edition, Express Edition, Remote Adapter Deployment, or the license for Scale mode, you must set the operation mode of all your brokers to the correct value.

See “Checking the broker operation mode and function level” on page 111 for more details.

10. Optional: Change the broker function level.

If new message flow nodes are delivered in a fix pack, they show in the IBM Integration Toolkit, but are not enabled in the runtime broker environment. If you deploy a BAR file that includes a message flow that uses a new node, the deployment fails.

If you want to use and test the new nodes, you can enable them on an individual broker basis. See “Checking the broker operation mode and function level” on page 111 for more details.

Results

You have completed the tutorial.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

Related reference:

“Installation,” on page 619

Use the reference information in this section to understand installation requirements, installation options, and how they affect your computer.

Related information:

 [IBM Integration Bus requirements](#)

Coexistence and migration

IBM Integration Bus Version 9.0 can coexist with previous versions.

You can install IBM Integration Bus Version 9.0 on a computer on which you have installed previous versions, but each version must be installed into its own directory, referred to as *install_dir*. Different versions can coexist and can run independently, and you can migrate brokers from one version to another, if and when appropriate. For the Integration Bus component and IBM Integration Toolkit, you can install multiple instances of the same version on the same computer, each in its own separate directory. For example, the Integration Bus component can coexist with different fix pack levels, such as Version 8.0.0.0 with Version 8.0.0.1. You can install multiple instances of the same version of the Integration Bus component and IBM Integration Toolkit on the same computer, however only one version of IBM Integration Explorer is allowed.

The following sections provide further details:

- “Coexistence”
- “Migration” on page 43

Coexistence

IBM Integration Bus Version 9.0 can coexist with previous versions as follows:

- The Integration Bus component can coexist with runtime components at Version 8.0 and Version 7.0.
- The IBM Integration Toolkit can coexist with the toolkit at Version 8.0 and Version 7.0.

The following sections describe how to achieve coexistence, and the restrictions that apply.

Integration Bus component on distributed systems

When you install the Integration Bus component on distributed systems, the default action taken by the installation wizard is to complete a typical installation, which installs the Integration Bus component into a default directory. A default directory for a typical installation is provided, however you can change it if you prefer.

If you accepted the default location during installation, this directory is as follows. The default directory includes the version, release, modification, and fix of the product that you are installing in the format v.r.m.f (version.release.modification.fix):

Linux /opt/ibm/mqsi/v.r.m.f

UNIX /opt/IBM/mqsi/v.r.m.f

Windows 32-bit

C:\Program Files\IBM\MQSI\v.r.m.f

Windows 64-bit

C:\Program Files\IBM\MQSI\v.r.m.f for the 64-bit version of IBM Integration Bus

C:\Program Files (x86)\IBM\MQSI\v.r.m.f for the 32-bit version of IBM Integration Bus

These locations define the default value of *install_dir* on each platform.

Each unique version, release, modification, and fix of the product is therefore installed into a different default location.

The installation wizard differentiates at version, release, modification, and fix level; therefore it differentiates between different modification levels and fix pack levels. The current modification level is 0 (Version 9.0.0). If a later modification level is made available, it installs into a different default location, and therefore does not upgrade the Version 9.0.0 to the higher modification level (for example, Version 9.0.1).

The wizard does not install one fix pack over a previous fix pack, and prohibits you from installing a previous modification or fix pack over a more recent one.

You can install the product at the same version and release more than once; these installations can be at the same modification or fix pack level, or at different levels. To achieve concurrent installations, you must select the custom installation option and specify a unique directory for each installation (one of which can be the default directory).

You can also use the custom installation to install into a non-default directory.

If you have never completed a typical installation of the product on the computer on which you have selected a custom installation, the directory is initially set to the default directory, but you can change this default value to your chosen value.

If you attempt to install the Integration Bus component into a directory that already contains an installation of the Integration Bus component at a previous version, you are prompted to confirm that you want to continue with the installation, because it overwrites the existing installation. Cancel the installation and select a different directory to preserve your existing configuration.

If you install the same version and release more than once, the native installer support cannot manage these installations in the normal way. For more information, see “How to uninstall multiple installations of the Integration Bus component” on page 655.

You can use multiple installations at different modification or fix pack levels to test out fixes or new functions, or to stage your adoption of a new fix pack level. For more information, see “Applying service” on page 129.

During and after installation, files are also stored in the working directory, which is associated with the user ID with which you are currently logged on. The location depends on the operating system:

Linux and UNIX

`/var/mqsi`

Windows

`%PROGRAMDATA%\IBM\MQSI`

The environment variable `%PROGRAMDATA%` defines the system working directory. The default directory depends on the operating system:

- On Windows 7 and later operating systems: `C:\ProgramData\IBM\MQSI`

The actual location might be different on your computer.

If you have multiple installations on a single computer, you can review the contents of the file `install.properties`, which is stored in the root of the working directory. For each installation at Version 6.1 and above, the file is updated with the location and the level.

This example shows the contents of `install.properties` on a Windows 32-bit operating system on which a single installation has completed:

```
C:\Program Files\IBM\MQSI\9.0=9.0.0.0
```

(The backslash character `\` is interpreted as an escape character. It is inserted before each non-alphabetic and non-numeric character in the string to preserve the character. A colon, a space, and several backslash characters are escaped in this example.)

If you want to revert from your latest installation to a previous level for any reason, you must uninstall the current version and install the previous level of the product. Before you uninstall, back up any resources that you want to return to a previous state.

Because the version and release are included in the directory structure when you complete a typical installation, you can also install Version 9.0 and later releases on the computer on which you have already installed previous releases of WebSphere Message Broker. The Version 9.0 installation can coexist with the existing installation; you can operate the two configurations independently.

If you use custom installations for Version 9.0 and later releases, you can specify a unique installation directory for each release, and therefore achieve coexisting releases on a single computer.

The number of installations of Version 9.0, or later, is limited only by the availability of system resources.

Because different versions and releases can coexist, you can migrate to Version 9.0 from an earlier version in a controlled manner, and you do not have to migrate all brokers at the same time. For more information, see “Migration” on page 43.

z/OS

Integration Bus component on z/OS

On z/OS, you can install multiple copies of the Integration Bus component on the same computer if you specify a different installation location for each copy. The installations can run independently of each other. The code can be at the same or different version and release levels; Version 6.0, Version 6.1, Version 7.0, Version 8.0, and Version 9.0. The number of installations is restricted only by the availability of system resources.

For more details about locations, libraries, and file system paths, see the *Program Directory for IBM Integration Bus for z/OS*, in the IBM Publication Center publication number, GI13-3314.

Coexistence with components of the same version

Although you can install multiple versions of the Integration Bus component on the same system, you should not install both an x86 and an x86-64 version, of the same version number, on the same system. Some data files used by the Integration Bus component are specific to each version, and using files of the wrong version with a broker could cause unpredictable results. This warning applies for all versions of the product and for all platforms.

IBM Integration Toolkit on Linux on x86, Linux on x86-64, and Windows

Linux When you install the IBM Integration Toolkit, the default action taken by the installation wizard is to install Installation Manager files, shared files, and product-specific files into the following directories:

- Installation Manager installation directory:
/opt/IBM/InstallationManager
- Shared resources directory:
/opt/IBM/SDPShared/
- Package group directory:
/opt/IBM/IntegrationToolkit90

This location defines the default value of *install_dir* on this platform.

For a description of these directories, see “Installing the IBM Integration Toolkit” on page 79.

You can install multiple instances of the IBM Integration Toolkit Version 9.0 at the same modification or fix pack level, or at different levels, on a single computer. Each installation must be in a separate package group; package groups are described in more detail in “IBM Installation Manager” on page 636.

Windows

When you install the IBM Integration Toolkit, the default action taken by the installation wizard is to install Installation Manager files, shared files, and product-specific files into the following directories:

- Installation Manager installation directory:
C:\Program Files\IBM\InstallationManager for the 32-bit version of IBM Integration Bus
C:\Program Files (x86)\IBM\InstallationManager for the 64-bit version of IBM Integration Bus
- Shared resources directory:
C:\Program Files\IBM\SDPShared\ for the 32-bit version of IBM Integration Bus
C:\Program Files (x86)\IBM\SDPShared\v.r for the 64-bit version of IBM Integration Bus
- Package group directory:
C:\Program Files\IBM\IntegrationToolkit90 for the 32-bit version of IBM Integration Bus
C:\Program Files (x86)\IBM\IntegrationToolkit90 for the 64-bit version of IBM Integration Bus

This location defines the default value of *install_dir* on this platform.

For a description of these directories, see “Installing the IBM Integration Toolkit” on page 79.

You can install multiple instances of the IBM Integration Toolkit at the same modification or fix pack level, or at different levels, on a single computer. Each installation must be in a separate package group; package groups are described in more detail in “IBM Installation Manager” on page 636.

If you install the IBM Integration Toolkit on Windows and you specify your own directory location, be aware of the file system limit of 256 characters imposed by Windows file systems. This limit can cause restrictions in path specification to resources (for example, message flows), and might cause access problems if the combination of path and resource name exceeds this limit. Keep installation locations and resource names short to avoid problems associated with this restriction.

Setting the environment for an installation

Because you can have more than one installation on a single computer, you must ensure that the commands that you issue on that computer are directed to the correct version of installed code.

- On Linux and UNIX systems, you must run the profile file **mqsiprofile** to set up the correct environment before you run other IBM Integration Bus commands, such as **mqsicreatebroker**. The profile file is stored in *install_dir/bin*.

If you add the profile file to your system logon profile, it is run automatically whenever you log on.

- On Windows systems, a command console is available for each installation. So you must run commands in the correct window for a particular installation.

If you prefer, you can run the **mqsiprofile.cmd** file, which is stored in *install_dir\bin*.

If you have installed an earlier version of this product on the same computer, check that the earlier profile is not set for the current user ID. The two profiles are incompatible and might cause unpredictable results. Consider using a different user ID for each version and associate the correct profile with each user ID to avoid potential problems.

This requirement is not applicable on z/OS systems.

For more details about **mqsiprofile**, see “Setting up a command environment” on page 119.

IBM Integration Explorer

Only one copy of IBM Integration Explorer can be installed at any time on a single system, so you must ensure the latest level of IBM Integration Explorer is installed. For example, if you have IBM Integration Bus Version 9.0 and WebSphere Message Broker Version 8.0 installed on the same system, you must ensure that IBM Integration Explorer Version 9.0 is installed. This only applies to Windows, Linux on x86, and Linux on x86-64 systems. However, you can use IBM Integration Explorer to connect to brokers from previous versions.

Migration

Because you can install IBM Integration Bus Version 9.0 on the same computer as previous versions and other installations of Version 9.0, you are not required to complete any migration tasks before you install Version 9.0.

If IBM Integration Bus Version 9.0 requires a later version of WebSphere MQ than you are currently using, you must update any existing installations to use a version of WebSphere MQ that is supported by Version 9.0 before installing IBM Integration Bus Version 9.0.

All WebSphere MQ installations must be at Version 7.5.0.1 for distributed platforms, and at Version 7.1 for z/OS. For information about WebSphere MQ migration, see the WebSphere MQ information center. If you are migrating from WebSphere Message Broker Version 6.1, see “Migrating the Version 6.1 WebSphere Message Broker component” on page 225 for more information about how to manage the WebSphere MQ migration.

For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

For details about migrating to IBM Integration Bus Version 9.0, see Chapter 3, “Migrating,” on page 171.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

Chapter 3, “Migrating,” on page 171

To migrate a broker domain to IBM Integration Bus Version 9.0, plan your migration strategy, perform pre-migration tasks, migrate your domain components, then complete post-migration tasks.

Preparing the system

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

About this task

You might also want to complete other tasks, depending on your installation intentions.

Read the following sections before installing:

Procedure

1. “Setting up security” on page 44.
2. “Configuring temporary space on distributed systems” on page 50.
3. If you are installing on a distributed system, see “Accessing DVDs on the local system” on page 52.
4. If you are installing on Linux or UNIX systems, see the following topics:
 - “Checking the kernel configuration on Linux and UNIX systems” on page 58
 - Linux and UNIX systems: Configuring the syslog daemon
5. If you are installing on z/OS, see “Disk space requirements on z/OS” on page 623.

What to do next

When you have completed these tasks, follow the installation instructions for distributed systems in the appropriate chapter:

- On distributed systems, choose which IBM Integration Bus components to install. For more information, see “Choosing what to install” on page 59.
- On z/OS, refer to the *Program Directory for IBM Integration Bus for z/OS* on the IBM Integration Bus Library web page.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

Related information:

 IBM Integration Bus requirements

Setting up security

Set up the required security before you install IBM Integration Bus.

About this task

This section describes security requirements for installing the product and verifying your installation.

After installation, check the topics under the Security heading in the information center to review and implement the security requirements for additional users doing other tasks.

Security control of IBM Integration Bus components, resources, and tasks depends on the definition of users and groups of users (principals) to the security subsystem of the operating system. Check that you have the correct authority, and that the required principals are in place, before you install IBM Integration Bus.

User ID restrictions: some operating systems and other products impose restrictions on user IDs:

- On Windows systems, user IDs can be up to 12 characters long, but on Linux, UNIX, and z/OS systems, they are restricted to eight characters. Database products, for example DB2[®], might also restrict user IDs to eight characters. If you have a mixed environment, ensure that the user IDs that you use within the broker environment are limited to a maximum of eight characters.
- Ensure that the case (upper, lower, or mixed) of user IDs in your broker environment is consistent. In some environments, uppercase and lowercase user IDs are considered the same, but in other environments, user IDs of different case are considered unique. For example, on Windows the user IDs 'tester' and 'TESTER' are identical, but on Linux and UNIX systems they are recognized as different user IDs.
- Check the validity of spaces and special characters in user IDs to ensure that, if used, these characters are accepted by all relevant systems and products in your broker environment.

If your user ID does not conform to these restrictions, you might have problems with installation or verification. If so, use an alternative user ID, or create a new one, to complete installation and verification.

Set up the security appropriate to the operating systems that you are using:

- If you are installing on Linux or UNIX systems, go to “Security on Linux and UNIX systems.”
- If you are installing on Windows, go to “Security on Windows systems” on page 47.
- If you are installing on z/OS, go to “Security on z/OS systems” on page 50.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Preparing the system” on page 43

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

Related information:

 IBM Integration Bus requirements

Security on Linux and UNIX systems:

Set up the required security on Linux and UNIX systems before you install IBM Integration Bus.

About this task

Use the security facilities that are provided by your operating system to complete these tasks; for example, the Systems Management Interface Tool (SMIT) on AIX, or the System Administration Manager on HP-Itanium.

Complete the following actions:

Procedure

1. Log in to the system.

On AIX, you must log in as root. On Linux and on other UNIX computers, your user ID must have root authority. You need this level of authority to set up the security requirements for installing the Integration Bus component. The product can then be installed by a user who does not have root privileges.

If you are using a Linux on x86 or a Linux on x86-64 system and are not planning to install the Integration Bus component, continue with step 4 on page 47. Otherwise, follow your local security guidelines to acquire root authority; either log in as root, or log in as another user and become root.

The use of a user ID other than root itself has some advantages; it provides an audit trail of the user ID that installs the product and it limits the scope of root authority to tasks performed in a single session. The use of a user ID other than root might also be mandatory if you are logging in from a remote system.

2. If you plan to run the installation as a user with root authority, then complete the following steps:
 - a. Under root authority, the installation automatically creates a security group called *mqbrkrs*. You must add your root authority login ID to the group after it has been created.
 - b. If you have already installed WebSphere MQ on this system, a group called *mqm* and a user called *mqm* are defined. If you have not yet installed WebSphere MQ, you must create this group and user.
 - c. Add your root authority user login ID to the group *mqm*, along with the user ID *mqm*.

- d. On some systems, you must log off and log on again for these new group definitions (*mqbrkrs* and *mqm*) to be recognized.
3. If you plan to run the installation as a user without root authority, a user with root authority must complete the following steps before installation. If you have other versions of IBM Integration Bus already installed on the system, you must complete subtasks c, e, f, h, and i only.

- a. Create a security group called *mqbrkrs*. For example:

- For non-AIX systems


```
sudo groupadd mqbrkrs
```
- For AIX


```
sudo mkgroup mqbrkrs
```

Add your non-root installation user login ID to the group after it has been created.

- b. If WebSphere MQ has already been installed on this system, a group called *mqm* and a user called *mqm* are defined. If WebSphere MQ has not been installed, the group and user must be created.
- c. Add your non-root installation user login ID to the group *mqm*, along with the user ID *mqm*.
- d. If it does not exist, create the `/var/mqsi` directory. For example:


```
sudo mkdir /var/mqsi
```
- e. Ensure that the correct ownership and access permissions are set for the `/var/mqsi` directory. For example:


```
sudo chown mqm:mqbrkrs /var/mqsi
sudo chmod 775 /var/mqsi
```

If the directory already exists, perform the command recursively:

```
sudo chown -R mqm:mqbrkrs /var/mqsi
sudo chmod -R ug+rwX /var/mqsi
```

Note: This command recursively changes the ownership of all files and directories. Review existing files with a system administrator to ensure this action does not create a security risk.

- f. If multiple users are going to create and use multiple brokers, set the group ID of the `/var/mqsi` directory so new files and directories inherit the same group ID. Otherwise a broker created by *User1*, with a primary group of *User1*, will be accessible to *User2* (with a primary group of *User2*). For example:


```
sudo chmod g+s /var/mqsi
```

If the broker already exists, perform the command recursively:

```
find /var/mqsi -type f -exec chmod g+s {} \;
```

- g. If the `/var/mqsi/install.properties` file exists, ensure that your non-root installation user ID has write access to it. For example:


```
sudo chmod 664 /var/mqsi/install.properties
```
- h. If they do not exist, create the directories `/opt/ibm/mqsi` and `/opt/ibm/IE02` for Linux, or `/opt/IBM/mqsi` and `/opt/ibm/IE02` for UNIX. For example:
 - For Linux


```
sudo mkdir /opt/ibm/mqsi
sudo mkdir /opt/ibm/IE02
```
 - For UNIX

```
sudo mkdir /opt/IBM/mqsi
sudo mkdir /opt/ibm/IE02
```

- i. Ensure that the correct ownership is assigned to the `mqsi` directory, along with access permissions to both the `mqsi` and `IE02` directories. For example:

- For Linux

```
sudo chown mqm:mqbrkrs /opt/ibm/mqsi
sudo chmod 775 /opt/ibm/mqsi
sudo chmod 775 /opt/ibm/IE02
```

- For UNIX

```
sudo chown mqm:mqbrkrs /opt/IBM/mqsi
sudo chmod 775 /opt/IBM/mqsi
sudo chmod 775 /opt/ibm/IE02
```

- j. If there are any existing logs from a previous installation of IBM Integration Bus or `IE02` you must delete or rename them, as the installation program may not have the correct permission to replace them. The logs in question can be found in the `/var/mqsi` directory and have a file type of `.log`.

- k. If setting the `IATEMPDIR` variable, ensure that your non-root user ID has write permission to the directory you choose. For example:

```
sudo chmod 775 /tmp/IATEMP
```

Note: This action is only needed if you have insufficient space in the default file system for the installation.

4. Verification procedures are provided for Linux on x86 and Linux on x86-64. To complete verification, you do not require root authority. If you install with root authority, but do not want to complete verification with root authority, log off when you have completed installation. Log in with the same or a different user ID, but do not become root.

If you log in with another user ID, and have not already added this ID to the groups `mqbrkrs` and `mqm`, do so before you open the IBM Integration Toolkit.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Preparing the system” on page 43

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

“Setting up security” on page 44

Set up the required security before you install IBM Integration Bus.

Related information:

 [IBM Integration Bus requirements](#)

Security on Windows systems:

Set up the required security on Windows systems before you install IBM Integration Bus.

About this task

Before you install the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer, log on with a user ID that has Administrator authority.

If you are installing the Integration Bus component, the installation wizard calls the **mqsisetsecurity** command which completes the following tasks:

- Creates a new security group called *mqbrkrs*.
- Adds your current (logged on) user ID to the group *mqbrkrs*.
- Adds your current user ID to the group *mqm*, if that group exists.

The *mqm* group exists if you have already installed WebSphere MQ on this system. If you have not, call the **mqsisetsecurity** command when you have completed WebSphere MQ installation. If you use the Windows Launchpad (described in “Installing by using the Windows Launchpad” on page 60), it completes WebSphere MQ installation first.

If you prefer to create principals before you install IBM Integration Bus, use the security facilities provided by the Windows Control Panel.

If you are running Terminal Services on this computer, change user mode to ensure that actions taken during installation are completed correctly; for example, the creation of .ini files and other related files in the default system directory C:\Windows. If you do not change user mode, files might be written to other locations and, although the installation might complete successfully, the product might not work as expected.

- Before you install any product components, enter the following command to change user mode:
change user /install
- When installation is complete, enter the following command to restore the original user mode:
change user /execute

To complete verification, your user ID must have Administrator authority. If you log in with a different user ID from the ID with which you perform installation, you must add that user ID to the groups *mqbrkrs* and *mqm*. Use either the Windows security facilities or the **mqsisetsecurity** command (run this command after you have logged on with that different ID) to complete these additions.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Preparing the system” on page 43

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

“Setting up security” on page 44

Set up the required security before you install IBM Integration Bus.

“Security in a Windows domain environment”

Set up the required security configuration in a Windows domain environment.

Related information:

 IBM Integration Bus requirements

Security in a Windows domain environment:

Set up the required security configuration in a Windows domain environment.

About this task

You can use Windows domain groups to organize different levels of authorization to selective IBM Integration Bus resources across your domain. To design and implement this domain group topology, add each domain group to the relevant local security groups on the domain workstations. You can now manage authorities by adding domain user accounts to the appropriate domain groups. For information about the group membership required to administer IBM Integration Bus resources, see Security requirements for Windows systems.

Procedure

1. Design your authorization group categories, and define domain groups on the domain controller system that correspond to these authorization categories, by using Windows security. For example, suppose you have a single domain containing three distinct sets of systems, used in development, testing, and production. Within your organization, various user roles require different levels of authorization to WebSphere MQ and IBM Integration Bus resources on those systems.

Here is an example of how those authorization categories could map to domain groups:

Domain group	Description
ADM-MQprd	WebSphere MQ administrator authorities on production machines
ADM-MQuat	WebSphere MQ administrator authorities on test machines
ADM-MQdev	WebSphere MQ administrator authorities on development machines
ADM-MBprd	IBM Integration Bus administrator authorities on production machines
ADM-MBuat	IBM Integration Bus administrator authorities on test machines
ADM-MBdev	IBM Integration Bus administrator authorities on development machines

2. Define and configure domain user accounts on the domain controller, by using Windows security. Add each user account to one or more domain groups to determine the authorizations granted that account. For example:

Table 5.

Domain user account	Role	Group membership
MQadmPRD	WebSphere MQ administrator for production systems	ADM-MQprd
MQadmUAT	WebSphere MQ administrator for test systems	ADM-MQuat
MQadmDEV	WebSphere MQ administrator for development systems	ADM-MQdev
MBadmPRD	IBM Integration Bus administrator for production systems	ADM-MBprd
MBadmUAT	IBM Integration Bus administrator for test systems	ADM-MBuat
MBadmDEV	IBM Integration Bus administrator for development systems	ADM-MBdev
john.smith	WebSphere MQ and IBM Integration Bus administrator for production environments	ADM-MQprd, ADM-MBprd

3. Install and configure IBM Integration Bus on domain workstations.

- a. Install IBM Integration Bus on the workstation.
- b. Add your domain groups to local groups *mqm* or *mqbrkrs* as appropriate. In our example, if a particular workstation is to serve as a development machine, add domain group *ADM-MQdev* to local group *mqm*, and domain group *ADM-MBdev* to local group *mqbrkrs*.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Preparing the system” on page 43

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

“Setting up security” on page 44

Set up the required security before you install IBM Integration Bus.

“Security on Windows systems” on page 47

Set up the required security on Windows systems before you install IBM Integration Bus.



Resolving problems when installing

Work through the advice provided to help you to deal with problems that can arise when the product is installed.

Related information:



IBM Integration Bus requirements

Security on z/OS systems:

User ID security required for z/OS system product installation.

About this task

The user ID that you use to install the product must be no more than eight characters in length. It must also have suitable RACF privileges to perform SMP/E installation in your environment. The user ID must have a valid OMVS segment, because the product installs into the file system paths specified during the SMP/E APPLY processing.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Preparing the system” on page 43

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

“Setting up security” on page 44

Set up the required security before you install IBM Integration Bus.

Related information:



IBM Integration Bus requirements

Configuring temporary space on distributed systems

Configure temporary directory space to enable IBM Integration Bus Java components to operate correctly.

About this task

IBM Integration Bus contains Java components. Some of these components require temporary directory space on the file system in order to extract class files from node packages (PAR files) and Java packages (JAR files).

The extracted files might not be visible to interactive users on the system, however, they consume space in the temporary directory.

The Java Runtime Environment also creates files in the temporary directory to support debug facilities.

To enable correct operation, allow at least 50 MB of space per integration server in the file system temporary directory for IBM Integration Bus components. This space is required in addition to any space required for operation of user developed artifacts. Use of JavaCompute nodes, Java user-defined nodes, or additional plug-in nodes implemented in Java might require further temporary directory space.

On Linux, UNIX, and z/OS computers, the TMPDIR directory is typically /tmp; on Windows computers, it is c:\temp. If this directory is not large enough to hold the JAR files, the broker does not start.

To change the location of the temporary directory, use one of the following methods:

Procedure

- Use the environment variable TMPDIR.
- Set the system property java.io.tmpdir.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Preparing the system” on page 43

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

Related information:

 [IBM Integration Bus requirements](#)

Accessing DVDs

Accessing DVDs to install IBM Integration Bus.

About this task

When you install or upgrade IBM Integration Bus, you can access DVDs on the local system, or you can set up a shared drive and access the shared resource from multiple computers.

You can also install or upgrade from installation images that you have obtained from Passport Advantage, if you are registered with this scheme:

Procedure

1. Read the instructions that are provided with the packages.

2. Download the images that you require for the operating systems in your environment.
3. Extract the contents of the images. Specify a short path for the directory to which you are extracting; the depth of directory structure and the directory names might cause problem if restrictions on some operating systems are reached. For example, the limit of 256 characters on Windows might be exceeded.
4. Set up the local or remote access to these images in the same way that you do for a DVD. For local installations, see “Accessing DVDs on the local system”; for remote installations, see “Accessing DVDs on a remote system” on page 55.

What to do next

If you are installing on Windows, you cannot enter a Universal Naming Convention (UNC) path (\\server\drive) to access the installation program; you must map the drive, otherwise the Java process times out. If you cannot map the drive, or choose not to map the drive, copy the contents of the DVD to a local drive and install from that drive.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Preparing the system” on page 43

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

Related information:

 IBM Integration Bus requirements

Accessing DVDs on the local system:

If you want to install product components from a local DVD, or a local downloaded image, complete this task.

About this task

Always consult your operating system documentation for exact details.

AIX

1. Log in as root. You cannot complete installation successfully if you have logged in as another ID and become root.
You must log in as root to set up the security requirements for installing the Integration Bus component. The installation can then be performed by a user who does not have root privileges.
2. Complete the security setup that is described in “Security on Linux and UNIX systems” on page 45.
3. Create a DVD mount point directory:

```
mkdir /dvdbroker
```


where */dvdbroker* is the mount point.
4. Insert the DVD into the drive of the computer on to which you want to install product components.
5. Use SMIT to mount the DVD, or use the following command:

```
mount -r -v cdrfs /dev/dvd0 /dvdbroker
```

where */dev/dvd0* is the DVD device and */dvdbroker* is the mount point.

You are now ready to install the product that is supplied on this DVD.

HP-Itanium

The HP-Itanium DVD has the format ISO 9660, with Rockridge extensions enabled. If volume management software is in use, the DVD mounts automatically when you insert it into the DVD drive. Alternatively, you can mount the DVD as described in the following procedure.

If the DVD is mounted incorrectly, some of the files cannot be read and the installation fails with a corrupted directory. You must mount the DVD with Rockridge extensions enabled.

1. Log in and ensure that your user ID has root authority.

You need root level authority to set up the security requirements for installing the Integration Bus component. The installation can then be performed by a user who does not have root privileges.

2. Complete the security setup that is described in "Security on Linux and UNIX systems" on page 45.
3. Create a DVD mount point directory and grant read-only access to all users:

```
mkdir /dvdbroker  
chmod 775 /dvdbroker
```

where */dvdbroker* is the mount point.

4. Insert the DVD into the drive of the computer on to which you want to install product components.
5. Mount the DVD by using the following command:

```
mount -F cdrfs /dev/dsk/device /dvdbroker
```

where */device* is the DVD device, for example */c0t0d0* and */dvdbroker* is the mount point.

You are now ready to install the product that is supplied on this DVD.

Linux

1. Log in and ensure that your user ID has root authority.

You need root level authority to set up the security requirements for installing the Integration Bus component. The installation can then be performed by a user who does not have root privileges.

2. Complete the security setup that is described in "Security on Linux and UNIX systems" on page 45.
3. Create a DVD mount point directory:

```
mkdir /dvdbroker
```

where */dvdbroker* is the mount point.

4. Insert the DVD into the drive of the computer on to which you want to install product components.
5. Run the following command:

```
mount -o ro -t iso9660 /dev/dvdrom /dvdbroker
```

where */dev/dvdrom* is the name of your DVD device (for example, */dev/hdc*) and */dvdbroker* is the mount point.

You are now ready to install the product that is supplied on this DVD.

Solaris

1. Log in and ensure that your user ID has root authority.
You need root level authority to set up the security requirements for installing the Integration Bus component. The installation can then be performed by a user who does not have root privileges.
2. Complete the security setup that is described in “Security on Linux and UNIX systems” on page 45.
3. Insert the DVD into the drive of the computer on to which you want to install product components.
4. Enter the following command to check whether the Volume Manager is running on your system:

```
/usr/bin/ps -ef | /bin/grep vold
```

If the Volume Manager is running, the DVD is mounted on `/dvdrom/vol_label` automatically, where `vol_label` is the volume label of the current DVD; for example, `wmb6_sol` for Runtime Disk 1.

5. If the Volume Manager is not started, run the following commands to mount the DVD:

```
mkdir -p /dvdbroker  
mount -F hsfs -o ro /dev/dsk/dvdrom /dvdbroker
```

where `/dev/dsk/dvdrom` is the DVD location (for example, `c0t0d0`) and `/dvdbroker` is the mount point directory.

Check where your DVD is located by using the command `iostat -En`. Alternatively, use the **volcheck** command to mount a DVD device automatically.

You are now ready to install the product that is supplied on this DVD.

Windows

1. Log on with a user ID that has Administrator authority.
2. Complete the security setup that is described in “Security on Windows systems” on page 47.
3. Insert the DVD into the drive of the computer on to which you want to install product components. The Launchpad opens.

You are now ready to install the product that is supplied on this DVD.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Preparing the system” on page 43

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

“Accessing DVDs” on page 51

Accessing DVDs to install IBM Integration Bus.

Related information:

 IBM Integration Bus requirements

Accessing DVDs on a remote system:

If you want to install product components from a remote (server) DVD, complete this task.

About this task

Always consult your operating system documentation for exact details of this task.

If you want to perform more than one installation of one or more components, you might find that a remote server setup provides some performance benefits, particularly for the IBM Integration Toolkit which is the largest component. You might also find this method more convenient if you want to run installations by using the silent interface.

To enable a remote installation, you must complete tasks on both the server (the computer on which the DVD is mounted) and each target system (on which you want to install the product). For details of the commands that are used in these examples, refer to the operating system documentation.

To set up a server, see “Setting up the server.”

To set up a target system, see “Setting up the target system” on page 57.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Preparing the system” on page 43

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

“Accessing DVDs” on page 51

Accessing DVDs to install IBM Integration Bus.

Related information:

 [IBM Integration Bus requirements](#)

Setting up the server:

You can either share the DVD drive on the server, or copy the installation images onto a disk and share the directory on that disk.

About this task

You can share a DVD drive on Linux or UNIX with any other supported Linux or UNIX system, but not with Windows. You can share a Windows DVD drive only with other Windows systems.

1. If you want to share a copy of the installation image, create the copy:
 - a. Create a directory on the server to store the installation images:

Linux and UNIX

Enter the following command:

```
mkdir /instbroker
```

where *instbroker* is the directory into which you copy the product files.

Windows

Enter the following command:

```
md m:\instbroker
```

where *m* is the drive on which you want to store the installation images and *instbroker* is the directory on that drive.

- b. Insert and mount the appropriate DVD in the drive as described in "Accessing DVDs on the local system" on page 52.

If you have inserted a runtime or toolkit DVD on Windows and autorun is enabled, the Launchpad is started. When the initial window opens, click **Cancel** to close it.

- c. Copy the complete contents of the DVD to the new directory.

Linux and UNIX

Enter the following command:

```
cp -rf /dvdrom/. /instbroker
```

Windows

Enter the following command:

```
xcopy f:\*.* m:\instbroker /e
```

where *f* is the DVD drive.

2. Grant users access to the drive that contains the product code. These instructions are the same for a disk drive on which you have copied the DVD contents, and for the DVD drive itself.

AIX Either type **smit** and click **Communications Applications and Services > NFS > Network File System (NFS) > Add a Directory to Exports List**, or enter the fast path command **smitty mknfsexp**. Complete the fields as appropriate and press Enter.

HP-UX and Linux

Use the **exportfs** command. The following example gives all users read-only access using NFS:

```
exportfs -i -o ro /instbroker
exportfs -a
```

where */instbroker* represents the DVD drive or the directory that contains the DVD copy.

Solaris

Use the **share** and **exportfs** commands. The following example gives all users read-only access using NFS:

```
share -F nfs -o ro -d "Broker LAN server" /instbroker
exportfs -a
```

where "Broker LAN server" is an optional description and */instbroker* represents the server DVD drive or directory containing the DVD copy.

Windows

Open Windows Explorer and right-click the drive that you want to share. Click **Sharing** and follow the instructions on the Properties dialog box.

Related tasks:

"Installing" on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration

Bus Installation Guide.

“Preparing the system” on page 43

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

“Accessing DVDs” on page 51

Accessing DVDs to install IBM Integration Bus.

Related information:

 IBM Integration Bus requirements

Setting up the target system:

Set up a target system to access DVDs from a server.

About this task

1. On Linux and UNIX systems, create a new directory on which to mount the shared directory. Enter the following command:

```
mkdir /remotebroker
```

where *remotebroker* is the name of the new directory.

2. Access the remote directory:

Linux and UNIX

Enter the following command:

```
mount server name:instbroker /remotebroker
```

where *server name* is the name of the server on which you created the DVD copy.

Windows

Connect to the appropriate drive and folder by using the **net use** command at a command prompt on the target system, for example:

```
net use x: \\server_name\instbroker
```

where *x*: is the required mapped drive on the target system.

If your shared installation directory name contains spaces (for example, Broker Image), enclose it in double quotation marks.

If your server is protected, you might need to specify a user ID and password on this command (see the Windows online help for more information about the **net use** command). Alternatively, use Windows Explorer or an alternative method to map the shared resource to a drive letter.

You cannot enter a UNC path (\\server\drive) to access the installation program; you must map the drive, as shown, otherwise the Java process times out. If you cannot map the drive, or choose not to map the drive, copy the contents of the DVD onto a local drive and install from that drive. In addition, you cannot enter a UNC path when the installation wizard requests a path as input; the wizard cannot interpret a UNC path.

3. Change to the remote image directory. You are now ready to run the Launchpad (on Windows only) or the installation wizard to install the product from the remote directory to your local system.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Preparing the system” on page 43

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

“Accessing DVDs” on page 51

Accessing DVDs to install IBM Integration Bus.

Related information:

 [IBM Integration Bus requirements](#)

Checking the kernel configuration on Linux and UNIX systems

Check the kernel configuration parameters on Linux and UNIX systems for prerequisite and corequisite products.

About this task

IBM Integration Bus has no specific requirements for kernel configuration parameters; however, other products might require particular settings. If you do not tune your kernel parameters to suit the products that you have installed, you might see unexpected results or a deterioration in performance.

Follow these steps to configure your kernel parameters:

Procedure

1. Check the documented values for the following products:

- WebSphere MQ
- DB2 (if it is installed)
- Other software that you have installed to work with IBM Integration Bus, including other databases.

You can access the relevant information for IBM products through IBM Knowledge Center.

2. Take the highest value for each parameter and compare it to the corresponding value in your kernel configuration.
3. If the current value is lower than the highest documented value, update the current setting by using the appropriate tooling that is supplied by the operating system provider. If the current value is higher, leave it unchanged.
4. On Solaris, increase the maximum number of concurrent open file descriptors on your system to at least 256.
5. If you have changed any kernel values, you might need to restart your system for these changes to take effect. Check the documentation for your operating system for further information about these parameters.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Preparing the system” on page 43

On some operating systems, you must complete several tasks before you install IBM Integration Bus.

Related information:

 [IBM Integration Bus requirements](#)

Choosing what to install

Choose which components and products to install.

Before you begin

On some operating systems, you must complete several tasks before you install IBM Integration Bus. For more information, see “Preparing the system” on page 43.

About this task

IBM Integration Bus consists of three components; the Integration Bus component, the IBM Integration Explorer, and the IBM Integration Toolkit. You can also install other products, such as the IBM Integration ODBC Database Extender:

- The Integration Bus component
- The IBM Integration Explorer
- The IBM Integration Toolkit
- The IBM Integration ODBC Database Extender

You can install the Integration Bus component on all supported operating systems. You can install the IBM Integration Explorer and the IBM Integration Toolkit on Windows 32-bit, Windows 64-bit, Linux on x86, and Linux on x86-64.

You can install IBM Integration Bus components and prerequisite products in silent mode on Linux, UNIX, and Windows. For more information, see “Installing IBM Integration Bus components and prerequisite products by using the sample script file” on page 65.

The Integration Bus component

The Integration Bus component is a set of execution processes that provides message processing facilities that interact with various application clients that use both point-to-point and publish/subscribe communications. You can create a broker only on the computer on which you have installed the Integration Bus component. You can use the IBM Integration Explorer, the IBM Integration Toolkit, or the command line to create local brokers. The message flows that you create are hosted by a broker. A broker can host many message flows, in one or more integration servers, and can support many clients.

You define how messages are received, processed, and delivered to receiving applications or subscribers:

- You can customize some message processing nodes in a message flow with mappings, ESQL, Java, PHP, and XSL style sheets.
- You can create message models to define message structures that are determined by C and COBOL data structures, industry standards such as SWIFT or EDIFACT, and XML DTD or schema.
- You can develop user-defined extensions (nodes and parsers) to support message processing options that are not provided by the supplied nodes and parsers.
- You can debug message flows and step through processing to check paths and results.
- You can use message flow aggregation to manage multiple requests and responses that are generated by a single input message.

You can install more than one IBM Integration Bus component on any system. For more information about how different installations can coexist, see “Coexistence and migration” on page 38.

For installation of the Integration Bus component, you can choose between a typical installation and a custom installation. These installation options are explained in “Coexistence and migration” on page 38.

IBM Integration Explorer

The IBM Integration Explorer is a stand-alone administration environment, that is based on the Eclipse platform, that communicates with one or more brokers. Administrators use the IBM Integration Explorer to manage the resources that are associated with these brokers. Install the IBM Integration Explorer on computers on which you intend to perform only administrative tasks.

IBM Integration Toolkit

The IBM Integration Toolkit is an integrated development environment and graphical user interface that is based on the Eclipse platform and the Rational® framework.

Application developers work in separate instances of the IBM Integration Toolkit to develop message flows, message sets, and user-defined nodes and parsers. You can access a shared repository (for example, CVS) to store resources and make them accessible in a secure manner to multiple users.

What to do next

Choose the interface that you want to use to install IBM Integration Bus. For more information, see “Installation and uninstallation interfaces” on page 652

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

Related information:

 [IBM Integration Bus requirements](#)

Installing by using the Windows Launchpad

Use the Windows Launchpad to install the IBM Integration Bus components and the prerequisite products.

Before you begin

Choose which IBM Integration Bus components to install. For more information, see “Choosing what to install” on page 59.

About this task

On Windows only, use the Launchpad for additional help with installing:

- The IBM Integration Toolkit.
- The IBM Integration Explorer.
- The Integration Bus component.
- Prerequisite products for the Integration Bus component.

If you use the Launchpad, you can install everything that you need, and do not have to follow the procedures described in other chapters for installing the Integration Bus component, the IBM Integration Toolkit, and the IBM Integration Explorer.

The Launchpad works with both physical media (the DVD) and with electronic images that you have downloaded from IBM Passport Advantage; however, the Launchpad depends on a file structure identical to that on the DVD, therefore you must not make any changes during or after download.

Note: Installing a 32-bit Broker image on a 64-bit Windows operating system - On some Windows platforms, the 32-bit installable image as well as the 64-bit installable image of the Integration Bus component is supported on the 64-bit hardware/Windows operating system. This combination is not true for all 64-bit versions of the Windows operating system. Please see the IBM Integration Bus Requirements Web site for support details. In particular, if you require to install the 32-bit Windows Integration Bus component on a 64-bit Windows operating system that supports this combination, you can use the Launchpad to do this, but be aware the Launchpad asks you to navigate to the program setup64.exe. In this specific instance, you can safely navigate to the setup32.exe program on the media and continue. Alternatively, from outside the Launchpad you can navigate to, and run, the setup32.exe installation program .

Multiple installations on a single computer

About this task

You can use the Launchpad to install only one instance of each component on a single computer. If you have selected and installed the broker or toolkit components, you cannot use the Launchpad to install these components again in a different location.

To install additional instances, run the appropriate installation wizard directly. For more information about completing these tasks, see “Installing the Integration Bus component” on page 69, “Installing the IBM Integration Toolkit” on page 79, and “Installing IBM Integration Explorer” on page 84.

The Launchpad also manages only one installation of WebSphere MQ on a single computer. Refer to the relevant documentation for these products if you want to install multiple instances.

Installation summary

About this task

The following steps summarize the actions that you must take to complete the installation.

Procedure

1. Check the readme.html file for any updates to these installation instructions.
2. Check that you have enough memory and disk space; refer to “Memory and disk space requirements” on page 621.
3. Decide whether you want to install from a server, or install locally on each system. These choices are described in “Accessing DVDs” on page 51.
4. Start the Windows Launchpad to install IBM Integration Bus and its prerequisite product; WebSphere MQ. For more information about using the Launchpad, see “Starting the Windows Launchpad” on page 62.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.



Resolving problems when installing

Work through the advice provided to help you to deal with problems that can arise when the product is installed.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related information:



IBM Integration Bus requirements

Starting the Windows Launchpad

Use the instructions to start the Windows Launchpad.

About this task

The Launchpad is available on every DVD and downloaded image from which product components or prerequisite products can be installed.

If you are using physical product media, the Launchpad starts automatically if autorun is enabled. If autorun is not enabled, or if you are installing from a downloaded image, navigate to root directory of the DVD or image, and double-click the file `mqsil\launchpad.exe` or type `mqsil\launchpad` in a command window and press **Enter**.

The **Installation** window is displayed. On the **Installation** window, you can install the set of products that are required for a default configuration of IBM Integration Bus, including WebSphere MQ, if it is not already installed. For more information, see “Installing a default configuration by using the Installation window” on page 63.

Access further information from the left pane:

- Click **Readme** to view the readme file `readme.html` in a new web browser window.

The Launchpad might have to search for an installation wizard for some of your selections. If you are installing from DVD, all the required products are available, but if you are installing from a downloaded image, the program might be on another downloaded image, or might not be in the expected location. If necessary, the Launchpad prompts you to take the appropriate action to find the file. The following table shows, for each supplied product, the program names and their locations on the downloaded images.

Table 6. Installation wizard names and locations used by the Windows Launchpad

Product	Installation wizard name	Directory	Downloaded image
WebSphere MQ Version 7.5	IBM WebSphere MQ.msi	\WebSphere_MQ\MSI	Runtime Disk 1
Integration Bus component	setup32.exe setup64.exe	\ (root directory)	Runtime Disk 1
IBM Integration Explorer	install.exe	\IBExplorer	Runtime Disk 1

Table 6. Installation wizard names and locations used by the Windows Launchpad (continued)

Product	Installation wizard name	Directory	Downloaded image
IBM Integration Toolkit	install.exe	\IM_Win32 ¹	Toolkit Disk 1

Notes:

1. The Launchpad starts Installation Manager, which installs itself (if required), and starts the IBM Integration Toolkit installation.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

“Installing by using the Windows Launchpad” on page 60

Use the Windows Launchpad to install the IBM Integration Bus components and the prerequisite products.

Related information:

 IBM Integration Bus requirements

Installing a default configuration by using the Installation window:

Use the Windows Launchpad Installation window to install the set of products that are required for a default configuration of IBM Integration Bus.

Before you begin

Read “Installing by using the Windows Launchpad” on page 60 for background information.

For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

About this task

1. Start the Windows Launchpad. See “Starting the Windows Launchpad” on page 62 for more information.

When the Launchpad starts, the Installation window opens and displays the following minimum set of products, along with version numbers, that are required for a default configuration of IBM Integration Bus:

- WebSphere MQ
- IBM Integration Bus
- IBM Integration Toolkit
- IBM Integration Explorer

2. Check the initial installation status that is shown for each listed product:

- **Required** indicates that the product is not installed but is required for a minimum configuration, or you have cleared the associated check box to indicate that you do not intend to install the product, or the Launchpad cannot install it (and no check box is shown).
- **Pending** indicates that the product is not installed but is required for a minimum configuration, or the check box is selected to show that this product is to be installed.

- **Installed** indicates that the product is already installed at a level that is supported by IBM Integration Bus. The installed version is shown and no check box is displayed.
 - **Partial Installation** indicates that the product is installed, but not all components that are required to ensure that the successful operation of a minimum configuration broker domain are present on the system. The associated check box is selected to show that additional components are installed.
3. Click the plus sign to the left of each listed product in turn. The Launchpad displays more information about the product, which you can use to decide if you want it installed. The additional information also provides an estimate of the time that is taken to complete each product installation, if appropriate.
 4. WebSphere MQ Explorer requires the WebSphere Eclipse Platform to be installed; when you select WebSphere MQ, the Eclipse Platform is automatically installed.
 5. If you do not want to install a listed product, clear the check box that is associated with the product. Its status is changed to **Required**, because you cannot configure and verify your installation without all the listed products. If the product you have cleared is a prerequisite for another product, that other product is also cleared. You can complete installation of any remaining products, and install the cleared products at a later time.
 6. Click **Launch Installation for IBM Integration Bus**.

If you have cleared one or more of the required products, you are asked to confirm your choices.

The Launchpad installs the products that you have selected in the order shown. You cannot change the Installation window after you have started the installation process. When the Launchpad starts each installation, it updates status from **Pending** to **In Progress**.

If you have the incorrect WebSphere MQ level, for example version 7.0.1, you might get the following warning:

One or more of the selected installs will upgrade an existing version of the installed software. Please be aware that the old version may be removed with no further prompting.
Click OK to continue.

The Launchpad uninstalls the incorrect WebSphere MQ level, and then installs the correct level of WebSphere MQ.

- If you have selected WebSphere MQ, the Launchpad starts the installation wizard silent interface for WebSphere MQ; default values are used for all options. A progress bar is displayed so that you can check on progress.
- If you have selected IBM Integration Bus (the Integration Bus component), the Launchpad starts the installation wizard graphical interface. You must supply the input that is required by the installation wizard.

The installation wizard guides you through a series of windows where you can make choices about where to install the component.

You must also read and accept the Software License Agreement that is displayed.

The license agreement covers your use of WebSphere MQ. Be aware it is only licensed for use with IBM Integration Bus and must not be used for other purposes.

- If you have selected the IBM Integration Toolkit, the Launchpad starts the installation wizard graphical interface.

The process is controlled by Installation Manager, which installs itself if it is not already installed on this computer. For further information about Installation Manager, see “Additional software requirements” on page 633.

The installation wizard guides you through a series of windows where you can make choices about where to install the component, which package group to install it in, and which language support you want to install. You must supply all input that is required by the installation wizard. You must also accept the Software License Agreement that is displayed.

If you want to launch the IBM Integration Toolkit when its installation wizard is complete, select **IBM Integration Toolkit** in the list of installed products that is displayed on the completion window. When you click **Finish**, the wizard ends and returns control to the Launchpad, and the IBM Integration Toolkit is started.

- If you have selected IBM Integration Explorer, the Launchpad starts the installation wizard graphical interface. You must supply the input that is required by the installation wizard.

The installation wizard guides you through a series of windows where you can make choices about where to install the component.

You must also read and accept the Software License Agreement that is displayed.

The status of each product changes to **Installed** when the Launchpad completes its installation.

When the Launchpad has installed all your selected products, it returns control to the Installation window.

7. Click **Refresh** to check the status of each product listed.
8. Click **Exit Launchpad** to end the program.

What to do next

Check the Installation has installed all the WebSphere MQ components that you require. IBM Integration Bus requires only the server, WebSphere MQ Explorer, and the Java Messaging component, and only these components are installed. If you want additional components, use the WebSphere MQ installer.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

“Installing by using the Windows Launchpad” on page 60

Use the Windows Launchpad to install the IBM Integration Bus components and the prerequisite products.

Related information:

 [IBM Integration Bus requirements](#)

Installing IBM Integration Bus components and prerequisite products by using the sample script file

Use the Linux and UNIX `installAllSilent.sh` or Windows `installAllSilent.bat` sample script file to install the IBM Integration Bus components and prerequisite products in silent mode.

Before you begin

Choose which IBM Integration Bus components to install. For more information, see “Choosing what to install” on page 59.

For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

About this task

Two sample script files are supplied. For Linux and UNIX the `installAllSilent.sh` script file is located in `/sample-scripts`, for Windows the `installAllSilent.bat` script file is located in `\sample-scripts`. Use the relevant script file to install the following products:

- Integration Bus component
- IBM Integration Toolkit (Linux on x86, Linux on x86-64, and Windows only)
- IBM Integration Explorer (Linux on x86, Linux on x86-64, and Windows only)
- WebSphere MQ
- IBM Integration ODBC Database Extender

If you use a script file, you can install everything that you need, and do not have to follow the procedures that are described in other chapters for installing the Integration Bus component, the IBM Integration Toolkit, and the IBM Integration Explorer.

Note: Your user ID must have root or administrator authority in order to use the script file.

Multiple installations on a single computer

About this task

You can use a script file to install only one instance of each component on a single computer. If you have selected and installed the broker or toolkit components, you cannot use the script file to install these components again in a different location.

To install additional instances, run the appropriate installation wizard directly. For more information about completing these tasks, see “Installing the Integration Bus component” on page 69, “Installing the IBM Integration Toolkit” on page 79, and “Installing IBM Integration Explorer” on page 84.

The script file also manages only one installation of WebSphere MQ on a single computer. Refer to the relevant documentation for these products if you want to install multiple instances.

Installation summary

About this task

The following steps summarize the actions that you must take to complete the installation.

Procedure

1. Check the `readme.html` file for any updates to these installation instructions.
2. Check that you have enough memory and disk space; refer to “Memory and disk space requirements” on page 621.

3. Before installing IBM Integration Toolkit on Linux, check that the default permissions are set correctly by running the following command under a user ID with root authority:

```
umask
```

A value of 0022 should be returned, indicating the permissions are correctly set. If any other value is returned, set the correct permissions by running the following command:

```
umask 0022
```

4. Decide whether you want to install from a server, or install locally on each system. These choices are described in “Accessing DVDs” on page 51.
5. Start the installation of the IBM Integration Bus components and prerequisite products in silent mode by calling the script file.

Calling the script file

About this task

The `installAllSilent.sh` and `installAllSilent.bat` script files work with both physical media (the DVD) and with electronic images that you have downloaded from IBM Passport Advantage. If you run the script without specifying any additional options, the script depends on a file structure that is identical to the file structure on the DVD. For more information about the structure of the DVD, see “The Integration Bus component and IBM Integration Explorer package” on page 649.

If you have downloaded the installation images into a file structure identical to the file structure on the DVD, use the following command to run the script file in silent mode. Where `-ACCEPT_ALL_LICENSES` indicates that the license has been accepted. By using the `-ACCEPT_ALL_LICENSES` parameter, you are accepting the terms of the license agreement:

Linux and UNIX

From a command prompt, change to the `/sample-scripts` directory and run one of the following commands:

```
installAllSilent.sh -ACCEPT_ALL_LICENSES
```

You can customize the script depending on your requirements. For example, you might want to refine the components you require to be installed by editing and updating the script file. If the installation images are located elsewhere, run the script with the following options:

```
installAllSilent.sh -ACCEPT_ALL_LICENSES -WMQ_INSTALL_IMAGE  
/Directory_of_WMQ_install_image
```

Where `Directory_of_WMQ_install_image` is the location of the WebSphere MQ installation image.

```
installAllSilent.sh -ACCEPT_ALL_LICENSES -IB_INSTALL_IMAGE  
/Directory_of_IB_install_image
```

Where `Directory_of_IB_install_image` is the location of the Integration Bus component installation image.

Linux on x86 and Linux on x86-64 only

From a command prompt, change to the `/sample-scripts` directory and run one of the following commands:

```
installAllSilent.sh -ACCEPT_ALL_LICENSES -IX_INSTALL_IMAGE  
/Directory_of_IX_install_image
```

Where *Directory_of_IX_install_image* is the location of the IBM Integration Explorer installation image.

```
installAllSilent.sh -ACCEPT_ALL_LICENSES -ITK_INSTALL_IMAGE  
/Directory_of_ITK_install_image
```

Where *Directory_of_ITK_install_image* is the location of the IBM Integration Toolkit installation image.

Windows 32-bit and Windows 64-bit

From a command prompt, change to the \sample-scripts directory and run one of the following commands:

```
installAllSilent.bat -ACCEPT_ALL_LICENSES
```

You can customize the script depending on your requirements. For example, you might want to refine the components you require to be installed by editing and updating the script file. If the installation images are located elsewhere, run the script with the following options:

```
installAllSilent.bat -ACCEPT_ALL_LICENSES -WMQ_INSTALL_IMAGE  
\Directory_of_WMQ_install_image
```

Where *Directory_of_WMQ_install_image* is the location of the WebSphere MQ installation image.

```
installAllSilent.bat -ACCEPT_ALL_LICENSES -IB_INSTALL_IMAGE  
\Directory_of_IB_install_image
```

Where *Directory_of_IB_install_image* is the location of the Integration Bus component installation image.

```
installAllSilent.bat -ACCEPT_ALL_LICENSES -IX_INSTALL_IMAGE  
\Directory_of_IX_install_image
```

Where *Directory_of_IX_install_image* is the location of the IBM Integration Explorer installation image.

```
installAllSilent.bat -ACCEPT_ALL_LICENSES -ITK_INSTALL_IMAGE  
\Directory_of_ITK_install_image
```

Where *Directory_of_ITK_install_image* is the location of the IBM Integration Toolkit installation image.

On Linux and UNIX, the IBM Integration ODBC Database Extender SupportPac automatically installs in silent mode when the installation of the Integration Bus component completes. For more information, see “Installing the IBM Integration ODBC Database Extender” on page 76.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.



Resolving problems when installing

Work through the advice provided to help you to deal with problems that can arise when the product is installed.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related information:



IBM Integration Bus requirements

Installing the Integration Bus component

Use the installation wizard to install the Integration Bus component.

Before you begin

Read the topic on “Preparing for installation” on page 34 and perform the tasks that are described in “Preparing the system” on page 43.

About this task

Complete the following tasks to install the Integration Bus component on all supported operating systems.

Note: In all cases, the installation log files are only available after you have exited the installer.

If you are installing the Integration Bus component on Windows, you can use the LaunchPad to complete this task. For more information, see “Installing by using the Windows Launchpad” on page 60.

Also, you can install IBM Integration Bus components and prerequisite products in silent mode on Linux, UNIX, and Windows. For more information, see “Installing IBM Integration Bus components and prerequisite products by using the sample script file” on page 65.

If you do not want to use the LaunchPad or the sample script file, complete the tasks in this topic instead.

The following list identifies the choices that you have for installing the Integration Bus component, and the actions that you must take to complete your chosen task:

Procedure

1. Check the `readme.html` for any updates to these installation instructions. The `readme` file location is shown in “System requirements” on page 619.
2. Check that you have enough memory and disk space; see “Memory and disk space requirements” on page 621.
3. If you do not already have WebSphere MQ installed, install it before you install the Integration Bus component.

Although you can install WebSphere MQ after you have installed the Integration Bus component, the installation wizard checks that you have the supported level of WebSphere MQ, or later, installed. If this check fails when you are using the graphical or console interface, the installation wizard displays a warning that lists potential problems. If you decide to continue, you must complete the installation of WebSphere MQ before you create or start any brokers.

If you start the installer by using the silent interface and WebSphere MQ is not installed, or WebSphere MQ is installed but it is not at the prerequisite level, the Integration Bus component installs without displaying the prerequisite warning. You must therefore ensure that you have the prerequisite installation of WebSphere MQ installed before you create or start any brokers.

4. Decide if you want to install from a remote server, or to install locally on each system. These choices are described in “Accessing DVDs” on page 51 for DVDs and for installation images that you can download from Passport

Advantage. The instructions here do not differentiate between DVDs and downloaded images because the behavior is the same.

5. Decide whether to use the graphical installation, a console installation, or a silent installation. For more information about these interfaces, see “How to install and uninstall the Integration Bus component” on page 653.
 - To use the graphical installation, continue with these instructions.
 - To use the installation wizard with a screen reader, see “Installing the Integration Bus component by using console mode with a screen reader” on page 71.
 - To use a console installation, see “Installing the Integration Bus component in console mode” on page 72.
 - To use a silent installation, see “Installing the Integration Bus component in silent mode” on page 73.
6. Determine the installation wizard name for your operating system. For more information, see “Installation wizard names” on page 660.
7. Start the installation wizard graphical interface locally or remotely.
 - a. To install locally, load the product DVD, then complete the following steps:

Linux and UNIX

Open a command prompt and navigate to the root directory of the DVD. Type the installation wizard name with no options, and press **Enter**.

Windows

Take one of the following actions:

- If autorun is enabled, the Launchpad is started immediately. To use the Launchpad, see “Installing by using the Windows Launchpad” on page 60. To cancel the Launchpad, click **Exit Launchpad**.
 - In Windows Explorer, navigate to the root directory of the DVD. Locate the installation wizard and double-click the wizard to start it.
 - Open a command prompt and navigate to the root directory of the DVD. Type the installation wizard name with no options, and press **Enter**.
- b. To install from a remote server, access the remote DVD drive or network drive on which the product media is available. Find the installation wizard on the DVD or network drive and start it as previously described.
8. When the wizard starts, navigate through the windows and provide input when requested. You must also read and accept the Software License Agreement.

If the directory that you specify for installation already contains a previous version of WebSphere Message Broker, for example; Version 8.0.0.2, the installation wizard prevents you from installing Version 9.0 in this location. You must specify a different location. You can then migrate components to Version 9.0 from the previous version, when appropriate.

Linux and UNIX only

The IBM Integration ODBC Database Extender SupportPac automatically installs in silent mode when the installation of the Integration Bus component completes. For more information, see “Installing the IBM Integration ODBC Database Extender” on page 76.

9. When the summary window displays, check your choices and click **Next** to complete the installation. A progress bar displays so that you can check on progress.
10. If you experience problems during installation, see *Resolving problems when installing*.

What to do next

When the installation has completed, see the following topics:

- “Verifying your IBM Integration Bus installation” on page 104
This topic describes how to verify your installation on Linux on x86, Linux on x86-64, or Windows by using either the IBM Integration Toolkit or the IBM Integration Explorer.
- “Checking the broker operation mode and function level” on page 111
- “Start and main menu updates after installation” on page 665

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.



Resolving problems when installing

Work through the advice provided to help you to deal with problems that can arise when the product is installed.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.



Resolving problems when uninstalling

Work through the advice provided to help you to deal with problems that can arise when the product is uninstalled.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related information:



IBM Integration Bus requirements

Installing the Integration Bus component by using console mode with a screen reader

You can install the Integration Bus component by using console mode with screen reader software for accessibility reasons.

About this task

Complete the following steps to use console mode with screen reader software:

Procedure

1. Install a screen reader program, such as JAWS.
2. Install the Integration Bus component by using console mode. For more information, see “Installing the Integration Bus component in console mode” on page 72.

Related tasks:

“Installing the Integration Bus component in console mode” on page 72

Install the Integration Bus component by using the installation wizard in console

mode.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Installing the Integration Bus component in console mode

Install the Integration Bus component by using the installation wizard in console mode.

Before you begin

On Linux and UNIX systems, ensure that you set up the required security before you install the Integration Bus component. See “Security on Linux and UNIX systems” on page 45 for more information.

About this task

The following list identifies the choices that you have for installing the Integration Bus component in console mode, and the actions that you must take to complete your chosen task:

Procedure

1. Determine the installation wizard name for your operating system. See “Installation wizard names” on page 660.
2. Locate the installation wizard in the root directory of the local or remote DVD, or the network drive.
3. If you are installing on HP-Itanium, `/usr/sbin/` must be included in the PATH statement.
4. Enter the following command at a command prompt for default invocation:
`installer -i console` (where *installer* is the installation wizard name).
If you start the installation from a directory other than the one in which the wizard exists, include the absolute or relative path with the command name.
5. When the wizard starts, navigate through the windows and provide input when requested. You must also read and accept the Software License Agreement.

The installation process specifies a default installation location that, if accepted, is version specific; for example: `C:\Program Files\IBM\MQSI\v.r.m.f`. The default directory includes the version, release, modification, and fix of the product, in the format v.r.m.f (version.release.modification.fix). You do have the option of specifying a different directory location, but the installation process does not proceed if the location that you specify already contains a previous version of the Integration Bus component. If the location that you specify already contains a previous version of the Integration Bus component, you must specify a different location. You can then migrate components to Version 9.0 from the previous version, when appropriate.

6. When the summary window is displayed, check your choices and follow on-screen instructions to complete installation.

Linux and UNIX only

The IBM Integration ODBC Database Extender SupportPac automatically installs in silent mode when the installation of the

Integration Bus component completes. For more information, see “Installing the IBM Integration ODBC Database Extender” on page 76.

7. If you experience problems during installation, see Resolving problems when installing.

What to do next

When you have completed installation, see the following topics:

- “Verifying your IBM Integration Bus installation” on page 104
This topic describes how to verify your installation on Linux on x86, Linux on x86-64, or Windows by using either the IBM Integration Toolkit or the IBM Integration Explorer.
- “Checking the broker operation mode and function level” on page 111
- “Start and main menu updates after installation” on page 665

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Installing the Integration Bus component” on page 69

Use the installation wizard to install the Integration Bus component.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.



Resolving problems when installing

Work through the advice provided to help you to deal with problems that can arise when the product is installed.

Related information:



IBM Integration Bus requirements

Installing the Integration Bus component in silent mode

Install the Integration Bus component by using the installation wizard in silent mode.

About this task

Decide whether to install only the Integration Bus component in silent mode or IBM Integration Bus components and prerequisite products.

- To install the Integration Bus component in silent mode, continue with these instructions.
- To install IBM Integration Bus components and prerequisite products in silent mode, see “Installing IBM Integration Bus components and prerequisite products by using the sample script file” on page 65 (Linux, UNIX, and Windows only).

The following list identifies the choices that you have for installing the Integration Bus component in silent mode, and the actions that you must take to complete your chosen task.

Procedure

1. Determine the installation wizard name for your operating system. See “Installation wizard names” on page 660.
2. Locate the installation wizard in the root directory of the local or remote DVD, or the network drive.
3. If you are installing on HP-Itanium, `/usr/sbin/` must be included in the PATH statement.
4. Decide if you want to specify the silent installation options by using a response file or by using the command line, then start the installation.

- **Using a response file:**

When specifying a silent installation, you can use a response file to define the required responses to the options provided by the installer.

- You can record a response file of your own by using the following command:

```
installer -r response_file_directory
```

Where *installer* is the installation wizard name. Where *response_file_directory* is the directory path in which the response file is saved.

- Alternatively, you can use the following example response files. These examples are also available on the runtime installation media under the folder `sample-scripts`

In these examples, `LICENSE_ACCEPTED=TRUE` is specified, which means that the license has been accepted. The `v.r.m.f` under `USER_INSTALL_DIR=` specifies the full version release of the Integration Bus component, for example `9.0.0.0`. By specifying `INSTALLER_UI=silent`, the installer will run in silent mode and proceed to use the values specified in the response file.

The flag `External_License_Path` can be used to point to the directory containing the license files if they are not located in the same directory of the installer, see “Installing service on Linux and UNIX” on page 136 and “Installing service on Windows” on page 133 for information about when this flag is applicable. Finally, the flag `INSTALL_dotNET_FRAMEWORK` in the Windows response file, controls whether the Microsoft .NET framework is installed.

By using the example scripts, you are accepting the terms of the license agreement.

Linux

```
# IBM Integration Bus Sample Response File
# This file contains sample values for installing IBM Integration Bus.

# Has the license been accepted?
#-----
LICENSE_ACCEPTED=TRUE

# Run the installer in silent mode
#-----
INSTALLER_UI=silent

# Choose Install Folder
#-----
USER_INSTALL_DIR=/opt/ibm/mqsi/v.r.m.f

# License location
#-----
External_License_Path=
```

UNIX

```
# IBM Integration Bus Sample Response File
# This file contains sample values for installing IBM Integration Bus.

# Has the license been accepted?
#-----
LICENSE_ACCEPTED=TRUE

# Run the installer in silent mode
#-----
INSTALLER_UI=silent

# Choose Install Folder
#-----
USER_INSTALL_DIR=/opt/IBM/mqsi/v.r.m.f

# License location
#-----
External_License_Path=
```

Windows

```
# IBM Integration Bus Sample Response File
# This file contains sample values for installing IBM Integration Bus.

# Has the license been accepted?
#-----
LICENSE_ACCEPTED=TRUE

# Run the installer in silent mode
#-----
INSTALLER_UI=silent

# Install the MS .NET Framework
#-----
INSTALL_dotNET_FRAMEWORK=TRUE

# Choose Install Folder
#-----
USER_INSTALL_DIR=C:\Program Files\IBM\MQSI\v.r.m.f

# License location
#-----
External_License_Path=
```

When the response file is complete, start the installation by running the following command from within the directory in which you downloaded the Integration Bus component. If you start the installation from a directory other than the one in which the installation wizard exists, include the absolute or relative path with the command name.

```
installer -f response_file_name
```

Where *installer* is the installation wizard name. Where *response_file_name* is the name of your response file.

For more information about how to use response files, how to create response files, and how to edit response files to define your requirements, see “Using response files with the Integration Bus component” on page 656.

- **Using the command line:**

When specifying a silent installation, you can use the command line to define the required responses to the options provided by the installer. Start the installation by running the following command from within the directory in which you downloaded the Integration Bus component.

In this example, `LICENSE_ACCEPTED=TRUE` is specified, which means that the license has been accepted. By using this command, you are accepting the terms of the license agreement.

```
installer -i silent -DLICENSE_ACCEPTED=TRUE -DUSER_INSTALL_DIR=  
installation_directory
```

Where *installer* is the installation wizard name. Where *installation_directory* is your chosen installation directory or the default installation directory for your platform. For more information about the default installation directories for distributed operating systems, see “Coexistence and migration” on page 38.

5. The installation wizard completes without any user interaction.

Linux and UNIX only

The IBM Integration ODBC Database Extender SupportPac automatically installs in silent mode when the installation of the Integration Bus component completes. For more information, see “Installing the IBM Integration ODBC Database Extender.”

If you experience problems during installation, see Resolving problems when installing.

What to do next

When you have completed installation, see the following topics:

- “Verifying your IBM Integration Bus installation” on page 104
This topic describes how to verify your installation on Linux on x86, Linux on x86-64, or Windows by using either the IBM Integration Toolkit or the IBM Integration Explorer.
- “Checking the broker operation mode and function level” on page 111
- “Start and main menu updates after installation” on page 665

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Installing the Integration Bus component” on page 69

Use the installation wizard to install the Integration Bus component.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.



Resolving problems when installing

Work through the advice provided to help you to deal with problems that can arise when the product is installed.

Related information:



IBM Integration Bus requirements

Installing the IBM Integration ODBC Database Extender

IBM Integration ODBC Database Extender encapsulates the unixODBC driver manager, which is an implementation of the Open DataBase Connectivity interface for UNIX systems. This topic describes how you install IBM Integration ODBC Database Extender.

Before you begin

Before you start:

Read the information about the unixODBC Project and the IBM solidDB® product family.

Install IBM Integration Bus Version 9.0 or later.

About this task

IBM Integration ODBC Database Extender is required when using IBM Integration Bus to interface with an ODBC data source that is not supported through the DataDirect ODBC drivers.

Note: Applies to Linux and UNIX only - If you install the Integration Bus component, IBM Integration ODBC Database Extender is automatically installed in silent mode when the installation of the Integration Bus component completes. In this case, there is no need to follow the installation steps provided in this topic.

Procedure

1. To install the IBM Integration ODBC Database Extender, download the package for your platform to the directory of your choice, or select the option to install the IBM Integration ODBC Database Extender from the product disk.

To download the package externally, select IBM Integration Bus on the WebSphere MQ SupportPacs web page and locate the package file.

The package file is called `install-ie02.bin` on all platforms.

This package must be run with the same privileges (typically root) that you used when installing IBM Integration Bus. The default IBM Integration ODBC Database Extender installation directory is `/opt/ibm/IE02/2.0.1`.

2. Run the installer in one of the following modes:

- Graphical
- Console
- Silent

- a. **Graphical.** From within the directory where you downloaded the package file, run the following command:

```
./install-ie02.bin
```

The installer launches within a separate window and guides you through the installation process. However, if the installer cannot launch in Graphical mode, it launches in console mode.

- b. **Console.** From within the directory where you downloaded the package file, run the following command:

```
./install-ie02.bin -i console
```

The installer launches, and guides you through the installation process from within the same console as the command was run.

- c. **Silent.** When specifying a silent installation, a response file is required to define the required responses to the options provided by the installer. See following example file:

```
# This file was built by the Replay feature of InstallAnywhere.  
# It contains variables that were set by Panels, Consoles or Custom  
# Code.
```

```
#Has the license been accepted
#-----
LICENSE_ACCEPTED=TRUE

#Choose Install Folder
#-----
USER_INSTALL_DIR=/opt/ibm/IE02/2.0.1
```

Note the above example specifies that the license has been accepted. By using this example script you are accepting the terms of the license agreement.

The response file must be named `installer.properties` and must be located in the same directory as the IBM Integration ODBC Database Extender installer.

Run the following command from within the directory where you downloaded the IBM Integration ODBC Database Extender:

```
./install-ie02.bin -i silent
```

After the installation has completed, review the installation log that is located under the installation path specified during the installation. The file name of the log is:

```
WebSphere_Message_Broker_ODBC_Database_Extender_InstallLog.log.
```

If you are installing over a previous installation (within the same installation directory), the previous installation is silently uninstalled before the new version is installed.

What to do next

Next, Configure the IBM Integration Bus Database Extender.

Related tasks:

“Configuring the IBM Integration ODBC Database Extender”

IBM Integration ODBC Database Extender encapsulates the unixODBC driver manager, which is an implementation of the Open DataBase Connectivity interface for UNIX systems, and this topic describes how you configure it.

“Connecting to a database from Linux and UNIX systems by using the IBM Integration ODBC Database Extender” on page 375

IBM Integration ODBC Database Extender encapsulates the unixODBC driver manager. You must set up and configure the broker to use it.

Related reference:



Sample IBM Integration Bus ODBC configuration files

A copy of the sample ODBC definition files that are supplied with IBM Integration Bus.

Configuring the IBM Integration ODBC Database Extender:

IBM Integration ODBC Database Extender encapsulates the unixODBC driver manager, which is an implementation of the Open DataBase Connectivity interface for UNIX systems, and this topic describes how you configure it.

Before you begin

Before you start:

Install the IBM Integration ODBC Database Extender.

About this task

To allow IBM Integration Bus to take advantage of the additional database support, it needs to know where you have installed the SupportPac. The location of the SupportPac needs to be set by the value of the environment variable `IE02_PATH`.

This environment variable is automatically set within the IBM Integration Bus profile during the installation of the SupportPac, by the creation of a script. The script name contains the pattern `IE02` and is placed within the directory `/var/mqsi/common/profiles`. See the following example file:

```
#!/usr/bin/sh
# This file was created as part of the IBM Integration
Bus ODBC
# Database Extender SupportPac install
export IE02_PATH=/opt/ibm/IE02/2.0.1
```

If you alter the location of your `$MQSI_WORKPATH` and, therefore, alter the location where IBM Integration Bus dynamically runs additional scripts while loading its profile, you must either copy the existing file, or create a new file containing the required contents within your new `${MQSI_WORKPATH}/common/profiles` directory.

You must not put the `lib` directory of the IBM Integration ODBC Database Extender into your library path (`LD_LIBRARY_PATH` or equivalent). If you do so, the IBM Integration ODBC Database Extender is not correctly loaded by IBM Integration Bus and unpredictable results can occur.

What to do next

Configure the database that you are going to use; see “Connecting to a database from Linux and UNIX systems by using the IBM Integration ODBC Database Extender” on page 375

Related tasks:

“Connecting to a database from Linux and UNIX systems by using the IBM Integration ODBC Database Extender” on page 375

IBM Integration ODBC Database Extender encapsulates the `unixODBC` driver manager. You must set up and configure the broker to use it.

Related reference:



Sample IBM Integration Bus ODBC configuration files

A copy of the sample ODBC definition files that are supplied with IBM Integration Bus.

Installing the IBM Integration Toolkit

Use the installation wizard graphical interface to install the IBM Integration Toolkit on Windows and Linux on x86.

About this task

If you are installing the IBM Integration Toolkit on Windows, you can use the LaunchPad. For more information about using the Windows Launchpad, see “Installing by using the Windows Launchpad” on page 60.

Also, you can install IBM Integration Bus components and prerequisite products in silent mode on Linux, UNIX, and Windows. For more information, see “Installing IBM Integration Bus components and prerequisite products by using the sample script file” on page 65.

If you are installing the IBM Integration Toolkit on a Red Hat Enterprise Linux 6.0 x86_64 (64-bit) computer, you must install the 32-bit versions of operating system libraries before you run the IBM Installation Manager. For full instructions, see the following technote on the IBM Support Portal: Unable to install Installation Manager on RHEL 6.0 (64-bit)

If you do not want to use the Windows LaunchPad or the sample script file, complete the following steps.

Procedure

1. Check the product readme file, `readme.html`, for updates to these installation instructions. The product readme file includes information pertinent to all components and platforms, and is maintained in US English on the product readmes web page. Translated readme files are available on the documentation FTP site.
2. Check that your computer has enough memory and disk space for your installation. Requirements for memory and disk space depend on the installation operating system, and on the IBM Integration Bus components that you are installing. For more information, see “Memory and disk space requirements” on page 621.
3. Before installing IBM Integration Toolkit on Linux, check that the default permissions are set correctly by running the following command under the user ID that will perform the installation:

```
umask
```

A value of `0022` should be returned, indicating the permissions are correctly set. If any other value is returned, set the correct permissions by running the following command:

```
umask 0022
```

4. Decide whether you want to install the IBM Integration Toolkit from a server or if you want to install the IBM Integration Toolkit locally. These choices are described in “Accessing DVDs” on page 51 for DVDs and for installation images that you can download from Passport Advantage.
5. Decide whether you want to use the installation wizard graphical interface to install the IBM Integration Toolkit or if you want to complete a silent installation. For more information about these interfaces, see “How to install and uninstall the IBM Integration Toolkit” on page 657. If you decide to use the installation wizard graphical interface, continue with these instructions. Alternatively, to use a silent installation, see “Installing the IBM Integration Toolkit in silent mode” on page 83.

In either scenario, the installation of the IBM Integration Toolkit is controlled by the IBM Installation Manager. If the Installation Manager is not already installed, it is installed before the IBM Integration Toolkit is installed.

6. Start the installation wizard graphical interface by navigating to the `Integration_Toolkit` directory and, depending on user ID authority and platform, run the following batch file:
 - If your user ID has root or administrator authority:
 - Linux on x86: `installToolkit.sh`
 - Windows: `installToolkit.bat`

- If your user ID does not have root or administrator authority:
 - Linux on x86: `installToolkit.sh -nonAdmin`
 - Windows: `installToolkit.bat -nonAdmin`

The installation wizard checks your system locale setting. If the locale setting listed in “Multicultural support” on page 663 is supported, the wizard continues in this locale. If the current setting is not supported, the wizard continues in US English. This language is used for installation only, and does not affect other processes on your computer.

The Installation Manager starts and the Install Packages window opens. The installation wizard is preconfigured to install both the Installation Manager and the IBM Integration Toolkit, therefore the IBM Integration Toolkit packages are already selected in this window. If the Installation Manager has not been installed on this computer, its packages are also selected and cannot be cleared.

7. Click **Next** to continue. The Software License Agreement window opens.
8. Read the license agreement, select **I accept the terms in the license agreements**, and click **Next**.

If you do not accept the license, the installation wizard ends. If you have not installed the Installation Manager on this computer, or if you have installed the Installation Manager but have not yet installed any product that is managed by the Installation Manager, the Shared Directory window opens. Continue with step 9. Otherwise, because the shared resources directory has already been defined, the Package group directory window opens, so you can continue with step 10.

9. If the IBM Integration Toolkit is not the first product that is installed by the Installation Manager, you might not be prompted to specify the shared resources directory. If you are prompted, specify the location of the shared resources directory that is used by all the products that are managed by the Installation Manager. The following directories show the default locations:
 - Linux on x86: `/opt/IBM/SDPShared/`.
 - Windows: `C:\Program Files\IBM\SDPShared\` for 32-bit editions, or `C:\Program Files (x86)\IBM\SDPShared\` for 64-bit editions.

To specify a different location, type over the default location, or click **Browse**.

The shared resources directory must not contain another installation of IBM Integration Toolkit, or other files or products.

If the Installation Manager is not yet installed, you must also specify its installation directory. The following directories show the default locations:

- Linux on x86: `/opt/IBM/InstallationManager/`.
- Windows: `C:\Program Files\IBM\InstallationManager\` for 32-bit editions, or `C:\Program Files (x86)\IBM\InstallationManager\` for 64-bit editions.

To specify a different location, type over the default location, or click **Browse**.

Click **Next**. The Package group directory window opens. You can create a new package group for the IBM Integration Toolkit, which requires you to specify an installation directory, or you can choose an existing package group to share the resources.

For more information about installing the IBM Integration Toolkit into a package group, and for further reference information about Package groups, see “IBM Installation Manager” on page 636.

10. If you want to use the IBM Integration Toolkit in a locale other than US English, select additional support from the list presented. English is always selected and installed; you cannot clear this selection. If you select one or

more alternative locales, documentation and properties files for all supported languages are installed. Click **Next** to continue. The Summary window opens.

11. Check your choices and click **Back** if you want to make further changes to your responses on any of the previous windows. This window displays guidance information for the space that is required for the packages that you are about to install and indicates that your disk has sufficient space.
Click **Next** to start installation. The Install Progress window opens.
12. The features that you are installing, their associated directories, and the locales that you have selected display for information. A progress bar displays, which you can use to check the status of the installation. When the installation has finished, the Completion window opens.
13. The wizard displays an indication of success or failure, and lists the products and options that have been installed. Click **View Log File** to check the results of the installation.

On Windows, you can indicate that you want the IBM Integration Toolkit to start when you click **Finish** to close the wizard. This option is not available on Linux on x86 because you might want to complete verification while logged on as a different user ID that does not have root authority.

14. If you experience problems during installation, see Resolving problems when installing.
15. If you installed IBM Integration Toolkit without root or administrator authority, switch to a user with root or administrator authority and run the following script file from the IBM Integration Toolkit home directory to complete the installation:
 - Linux on x86: <IIT90_Home>/wmbt/postconfig/postInstall_admin.sh
 - Windows: <IIT90_Home>\wmbt\postconfig\postInstall_admin.bat

What to do next

When you have completed the installation, see “Verifying your IBM Integration Bus installation” on page 104, “Checking the broker operation mode and function level” on page 111, and “Start and main menu updates after installation” on page 665.

If you do not install optional locales at this time, you can install them later in the following way:

- On Linux on x86, navigate to the /eclipse directory within the Installation Manager installation directory, and start the Installation Manager program IBMIM. (You cannot use the main menu entries unless you are already logged on as root; the menu item does not provide an option to become root, and root authority is required for all installation tasks.)
- On Windows, click **Start > Programs > IBM Installation Manager > IBM Installation Manager** to start the Installation Manager, and click **Modify Packages** to change your installation.

If you prefer to use the command line, navigate to the \eclipse directory within the Installation Manager installation directory, and start the Installation Manager program IBMIM.exe.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.



Resolving problems when installing

Work through the advice provided to help you to deal with problems that can arise when the product is installed.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

“Applying service to the IBM Integration Toolkit” on page 141

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

Related information:



IBM Integration Bus requirements

Installing the IBM Integration Toolkit in silent mode

Install the IBM Integration Toolkit by using the installation wizard in silent mode.

About this task

To perform a silent installation, complete the following steps:

Procedure

1. Determine the installation wizard name for your operating system. See “Installation wizard names” on page 660.
2. Locate the installation wizard in the `Integration_Toolkit` directory of the local or remote DVD, or the network drive.
3. If your user ID has root or administrator authority, enter the following command at a command prompt for an installation with all default settings:

Linux on x86

```
./installToolkit-silent.sh
```

Uses the `Integration_Toolkit/IM_Linux/mbtoolkit-silent.xml` response file that contains all of the Linux default settings for installation.

Windows

```
installToolkit-silent.bat
```

Uses the `Integration_Toolkit/IM_Win32/mbtoolkit-silent.xml` response file that contains all of the Windows default settings for installation.

If you want to specify non-default settings, include a different response file on the invocation. To create a different response file, complete the first installation by using the graphical interface, specifying the `-record` option. The installation wizard records a response file that includes all your chosen selections. For more information about how to record and use response files, see “Using response files with the IBM Integration Toolkit” on page 659.

4. If your user ID does not have root or administrator authority, you must first modify the relevant `mbtoolkit-silent.xml` response file to set the `installLocation` and `eclipseLocation` value to the path that is accessible to your user ID. Enter the following command at a command prompt to launch the silent installation:

Linux on x86

```
./installToolkit-silent.sh -nonAdmin
```

Uses the Integration_Toolkit/IM_Linux/mbtoolkit-silent.xml response file that contains all of the Linux default settings for installation.

Windows

```
./installToolkit-silent.bat -nonAdmin
```

Uses the Integration_Toolkit/IM_Win32/mbtoolkit-silent.xml response file that contains all of the Windows default settings for installation.

After installation has completed, switch to a user with root or administrator authority and run the following script file from the IBM Integration Toolkit home directory to complete the installation:

Linux on x86

```
<IIT90_Home>/wmbt/postconfig/postInstall_admin.sh
```

Windows

```
<IIT90_Home>\wmbt\postconfig\postInstall_admin.bat
```

5. The installation wizard completes without any user interaction. Check the log for success or failure of the installation process.
6. If you experience problems during installation, refer to Resolving problems when installing.

What to do next

When you have completed installation, see “Verifying your IBM Integration Bus installation” on page 104, “Checking the broker operation mode and function level” on page 111, and “Start and main menu updates after installation” on page 665.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Installing the IBM Integration Toolkit” on page 79

Use the installation wizard graphical interface to install the IBM Integration Toolkit on Windows and Linux on x86.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.



Resolving problems when installing

Work through the advice provided to help you to deal with problems that can arise when the product is installed.

Related information:



IBM Integration Bus requirements

Installing IBM Integration Explorer

To use IBM Integration Explorer only, without installing the complete IBM Integration Toolkit, use the IBM Integration Explorer installation wizard to install the IBM Integration Explorer.

Before you begin

Before you start:

- Alternatively, to install the complete IBM Integration Toolkit, see “Installing the IBM Integration Toolkit” on page 79.
- You can install IBM Integration Bus components and prerequisite products in silent mode on Linux, UNIX, and Windows. For more information, see “Installing IBM Integration Bus components and prerequisite products by using the sample script file” on page 65.
- The IBM Integration Explorer installation searches the Windows registry for a version of WebSphere MQ Explorer. If registry access is disabled by a local security policy such as "Prevent access to registry editing tools", the installation cannot find WebSphere MQ Explorer. Before you install IBM Integration Explorer, remove this restriction temporarily and reinstate it after the installation.

About this task

One version only of the IBM Integration Explorer can be active at any one time because one version only of the WebSphere MQ Explorer can be installed on any one system.

If you already have IBM Integration Explorer installed, and you decide to install a later version of IBM Integration Explorer, you must upgrade the existing version. For more information, see “Migrating the WebSphere Message Broker Explorer” on page 250.

If you add another installation of WebSphere MQ after you install IBM Integration Explorer, you must install IBM Integration Explorer again to enable it in all WebSphere MQ installations.

To install IBM Integration Explorer on Windows or Linux on x86 and Linux on x86-64, complete the following tasks.

Procedure

1. Check the readme.html for any updates to these installation instructions. The readme file location is shown in “System requirements” on page 619.
2. Check that you have enough memory and disk space; for more information, see “Memory and disk space requirements” on page 621.
3. If you do not already have WebSphere MQ installed, install it before you install the IBM Integration Explorer.
4. Decide whether you want to install from a server, or to install locally on each system. These choices are described in “Accessing DVDs” on page 51 for DVDs and for images that you downloaded from Passport Advantage (if you are registered with the scheme). The instructions here do not differentiate between DVDs and downloaded images; their behavior is the same.
5. Decide whether to use the graphical installation, a console installation, or a silent installation.
 - To use the graphical installation, continue with these instructions.
 - To use the installation wizard with a screen reader, see “Installing the IBM Integration Explorer by using console mode with a screen reader” on page 87.

- **Windows** To use a console installation on Windows, see “Installing IBM Integration Explorer in console mode on Windows” on page 88.
 - **Linux** To use a console installation on Linux, see “Installing IBM Integration Explorer in console mode on Linux” on page 89.
 - To use a silent installation, see “Installing IBM Integration Explorer in silent mode” on page 91.
6. Open the installation wizard by using the executable file or binary file. On Windows, this file is `install.exe`. On Linux on x86 and Linux on x86-64, this file is `install.bin`. These files are in the `\IBExplorer` directory on the DVD.
 7. When you start the wizard, continue with the installation process by working through each of the following panels. The installation wizard itself contains help information about selected panels.
 8. Select the language to use for the installation process, then click **OK**.
 9. On the Introduction panel, click **Next**.
 10. Read the software license agreement, select the option to accept the terms of the license, then click **Next**.
 11. Enter (or browse for) a product installation directory for IBM Integration Explorer, or accept the default location.
 On Windows, the default installation directory for IBM Integration Explorer is `C:\Program Files\IBM\IBExplorer`. On Linux on x86 and Linux on x86-64, the default installation directory is `/opt/IBM/IBExplorer`.
 If the product installation directory exists because you previously installed IBM Integration Explorer, click **Refresh** to refresh the existing installation or click **Select** to select a new product directory.
 12. Click **Next**.
 13. Read the summary panel. If necessary, click **Previous** to go back and modify earlier panels.
 14. Click **Install** and wait while the files are installed.
 15. On the Install complete panel, click **Done**.

Results

After installing IBM Integration Explorer you must complete the following step:

- To initialize the new installation of IBM Integration Explorer, run the following command from a command line in which the **mqsiprofile** command has not run:

```
strmqcfg -i
```

To run this command, you must be the administrator or root user.

You can now use the IBM Integration Explorer. To start the IBM Integration Explorer, complete one of the following steps:

- On Windows:
 Click **Start > All Programs > IBM Integration Bus 9.0.0.0 > IBM Integration Explorer**.
- On Linux on x86 and Linux on x86-64:
 Open a command shell in which the **mqsiprofile** command has not run and enter the `strmqcfg` command, or run `/usr/bin/strmqcfg`.

What to do next

After you complete installation, see “Verifying your IBM Integration Bus installation” on page 104.

Related tasks:

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

“Installing the IBM Integration Explorer by using console mode with a screen reader”

You can install the IBM Integration Explorer by using console mode with screen reader software for accessibility reasons.

“Installing IBM Integration Explorer in console mode on Windows” on page 88

You can install IBM Integration Explorer using the installation wizard in console mode on Windows and Linux.

“Installing IBM Integration Explorer in console mode on Linux” on page 89

You can install IBM Integration Explorer using the installation wizard in console mode on Windows and Linux.

“Installing IBM Integration Explorer in silent mode” on page 91

You can install IBM Integration Explorer by using the installation wizard in silent mode.



Resolving problems when installing

Work through the advice provided to help you to deal with problems that can arise when the product is installed.

“Verifying your IBM Integration Bus installation” on page 104

You can verify your installation of IBM Integration Bus by using the IBM Integration Toolkit or the IBM Integration Explorer.

“Applying service to the IBM Integration Explorer” on page 146

You can apply maintenance or fixes to the IBM Integration Explorer.

“Migrating the WebSphere Message Broker Explorer” on page 250

Migrate from a previous version of the WebSphere Message Broker Explorer to the IBM Integration Explorer Version 9.0.

“Installing the IBM Integration Toolkit” on page 79

Use the installation wizard graphical interface to install the IBM Integration Toolkit on Windows and Linux on x86.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Installing the IBM Integration Explorer by using console mode with a screen reader

You can install the IBM Integration Explorer by using console mode with screen reader software for accessibility reasons.

About this task

Complete the following steps to use console mode with screen reader software:

Procedure

1. Install a screen reader program, such as JAWS.
2. Install the IBM Integration Explorer by using console mode. For more information, see “Installing IBM Integration Explorer in console mode on Windows” or “Installing IBM Integration Explorer in console mode on Linux” on page 89.

Related tasks:

“Installing IBM Integration Explorer in console mode on Windows”

You can install IBM Integration Explorer using the installation wizard in console mode on Windows and Linux.

“Installing IBM Integration Explorer in console mode on Linux” on page 89

You can install IBM Integration Explorer using the installation wizard in console mode on Windows and Linux.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Installing IBM Integration Explorer in console mode on Windows

You can install IBM Integration Explorer using the installation wizard in console mode on Windows and Linux.

About this task

Use the following steps to install the IBM Integration Explorer using the console mode on Windows. If you encounter any problems during the installation process, you can view the install log `IBExplorer_install.log`.

Procedure

1. Type the following command on a command line to run a console install:
`<dvd_rom>/IBExplorer/install.exe -i console`
2. Select the language that you want to use for the installation process by typing the number next to the language and press **Enter**. The installer uses the term “locale” rather than “language”. Alternatively, press **Enter** to accept the default language.
3. Read the console install instructions, and press **Enter** to continue.
4. Read the software license agreement and type 1 to accept the terms of the license. Press **Enter**.
5. Type the name of the product installation directory for the IBM Integration Explorer and press **Enter**. Alternatively, press **Enter** to accept the default location. The default installation directory for the IBM Integration Explorer on Windows is `C:\Program Files\IBM\IBExplorer`.
6. Read the Pre-Installation Summary information and press **Enter** to install the IBM Integration Explorer. Wait while the files are installed.
7. On the Installation Complete panel, press **Enter** to exit the console installer.

Results

You can now use the IBM Integration Explorer. To use the IBM Integration Explorer you must start the WebSphere MQ Explorer. To start the WebSphere MQ Explorer, click **Start > All Programs > IBM WebSphere MQ > WebSphere MQ Explorer**, or open a command line in which the **mqsiprofile** command has not been run and enter the **strmqcfg** command.

What to do next

When you have completed installation, see “Verifying your IBM Integration Bus installation” on page 104.

Related tasks:

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

“Installing IBM Integration Explorer in console mode on Linux”

You can install IBM Integration Explorer using the installation wizard in console mode on Windows and Linux.

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

“Installing IBM Integration Explorer” on page 84

To use IBM Integration Explorer only, without installing the complete IBM Integration Toolkit, use the IBM Integration Explorer installation wizard to install the IBM Integration Explorer.

“Installing the IBM Integration Explorer by using console mode with a screen reader” on page 87

You can install the IBM Integration Explorer by using console mode with screen reader software for accessibility reasons.

“Installing IBM Integration Explorer in silent mode” on page 91

You can install IBM Integration Explorer by using the installation wizard in silent mode.

“Verifying your IBM Integration Bus installation” on page 104

You can verify your installation of IBM Integration Bus by using the IBM Integration Toolkit or the IBM Integration Explorer.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Installing IBM Integration Explorer in console mode on Linux

You can install IBM Integration Explorer using the installation wizard in console mode on Windows and Linux.

About this task

Use the following steps to install the IBM Integration Explorer using the console mode on Linux. If you encounter any problems during the installation process, you can view the install log `IBExplorer_install.log`.

Procedure

1. Type the following command on a command line to run a console install:
`<cd_rom>/IBExplorer/install.bin -i console`
2. Select the language that you want to use for the installation process by typing the number next to the language and press **Enter**. The installer uses the term "locale" rather than "language". Alternatively, press **Enter** to accept the default language.
3. Read the software license agreement and type 1 to accept the terms of the license. Press **Enter**.
4. Type the name of the product installation directory for the IBM Integration Explorer and press **Enter**. Alternatively, press **Enter** to accept the default location. The default installation directory for the IBM Integration Explorer on Linux systems is `/opt/IBM/IBExplorer`. If the product installation directory already exists because you have previously installed the IBM Integration Explorer, you can either refresh the existing installation or select a new product directory.
5. Confirm that the install location is correct, and press **Enter** to continue.
6. Read the Pre-Installation Summary information and press **Enter** to install the IBM Integration Explorer. Wait while the files are installed.
7. On the Installation Complete panel, press **Enter** to exit the console installer.

Results

You can now use the IBM Integration Explorer. To use the IBM Integration Explorer you must start the WebSphere MQ Explorer. To start the WebSphere MQ Explorer, open a command shell in which the `mqsiprofile` command has not been run and enter the `strmqcfg` command, or run `/usr/bin/strmqcfg`.

What to do next

When you have completed installation, see "Verifying your IBM Integration Bus installation" on page 104.

Related tasks:

"Setting up a command environment" on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

"Installing IBM Integration Explorer in console mode on Windows" on page 88

You can install IBM Integration Explorer using the installation wizard in console mode on Windows and Linux.

Chapter 5, "Configuring brokers for test and production environments," on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

"Installing IBM Integration Explorer" on page 84

To use IBM Integration Explorer only, without installing the complete IBM Integration Toolkit, use the IBM Integration Explorer installation wizard to install the IBM Integration Explorer.

"Installing the IBM Integration Explorer by using console mode with a screen reader" on page 87

You can install the IBM Integration Explorer by using console mode with screen reader software for accessibility reasons.

“Installing IBM Integration Explorer in silent mode”

You can install IBM Integration Explorer by using the installation wizard in silent mode.

“Verifying your IBM Integration Bus installation” on page 104

You can verify your installation of IBM Integration Bus by using the IBM Integration Toolkit or the IBM Integration Explorer.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Installing IBM Integration Explorer in silent mode

You can install IBM Integration Explorer by using the installation wizard in silent mode.

About this task

To install silently, you specify the path to a response file as an argument to the installation command or you can enter the parameters on the command line. You can create a response file during an installation by using the graphical installation wizard, or you can use a sample response file that is supplied in the `samples-scripts` directory in the root directory of the installation media.

Installing IBM Integration Explorer in silent mode by entering parameters on the command line:

Procedure

To run the installation wizard in silent mode by entering parameters on the command line, enter the following commands into a command prompt, where *InstallDirectory* is the path to where you want to install the product:

- On Windows:

```
install.exe -i silent -DLICENSE_ACCEPTED=true -DUSER_INSTALL_DIR=InstallDirectory
```

- On Linux:

```
install.bin -i silent -DLICENSE_ACCEPTED=true -DUSER_INSTALL_DIR=InstallDirectory -DTEMPDIR=TempDirectory
```

where *TempDirectory* is a directory to which you have write and execute access.

The **-DTEMPDIR** parameter is required only when you do not have write and execute access to the `/tmp` directory.

Installing IBM Integration Explorer in silent mode by using a response file:

Procedure

Use the following steps to create the response file and run the installation wizard in silent mode:

1. Create a response file by using the graphical installation wizard or by manually editing a response file:

- Create a response file during an installation by using the graphical installation wizard. You must provide a full path for the response file, or the response file is not created by the installation wizard.

Enter the following commands into a command prompt:

- On Windows:

```
install.exe -r <filepath>
```

- On Linux:
`install.bin -r <filepath>`

This command starts the GUI installation as normal, except that it also records all your answers and saves them in the file you specified, for example, `c:\temp\ibx-response.properties`. You can then use this response file with the silent installer for subsequent installations of IBM Integration Explorer.

- Alternatively, use the supplied template in the `samples-scripts` directory in the root directory of the local or remote DVD, or the network drive. Or use the following template to create a response file manually:

Note: The `ibx-response.properties` file that is supplied on the Installation media has `LICENSE_ACCEPTED=FALSE` so you must override this setting on the command line, or copy and update the file to indicate your acceptance of the license.

```
# Thu Jul 09 16:44:28 BST 2009
# Replay feature output
# -----
# This file was built by the Replay feature of InstallAnywhere.
# It contains variables that were set by Panels, Consoles or Custom Code.

#Has the license been accepted
#-----
LICENSE_ACCEPTED=TRUE

#Choose Install Folder
#-----
USER_INSTALL_DIR=C:\\Program Files\\IBM\\IBExplorer

#Uncomment for Silent Install
#-----
INSTALLER_UI=silent
```

2. To run the installation wizard in silent mode with the response file, run the following command:

- On Windows:
`install.exe -f <filename>`
- On Linux:
`install.bin -f <filename>`

where `<filename>` is the path to the response file, for example:
`d:\integrationbus_runtime1\sample-scripts\ibx-response.properties`.

Results

You can now use the IBM Integration Explorer. To use the IBM Integration Explorer you must start the WebSphere MQ Explorer. To start the WebSphere MQ Explorer, complete one of the following steps:

- On Windows:
Click **Start > All Programs > IBM WebSphere MQ > WebSphere MQ Explorer**, or, on a command line in which the `mqsiprofile` command has not been run, enter the `strmqcfg` command.
- On Linux:
Open a command shell in which the `mqsiprofile` command has not been run and enter the `strmqcfg` command, or run `/usr/bin/strmqcfg`.

What to do next

When you have completed installation, see “Verifying your IBM Integration Bus installation” on page 104.

Related tasks:

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

“Installing IBM Integration Explorer” on page 84

To use IBM Integration Explorer only, without installing the complete IBM Integration Toolkit, use the IBM Integration Explorer installation wizard to install the IBM Integration Explorer.

“Installing the IBM Integration Explorer by using console mode with a screen reader” on page 87

You can install the IBM Integration Explorer by using console mode with screen reader software for accessibility reasons.

“Installing IBM Integration Explorer in console mode on Windows” on page 88

You can install IBM Integration Explorer using the installation wizard in console mode on Windows and Linux.

“Verifying your IBM Integration Bus installation” on page 104

You can verify your installation of IBM Integration Bus by using the IBM Integration Toolkit or the IBM Integration Explorer.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Installing IBM Integration Administration for WebSphere Application Server

If you want to view or manage your IBM Integration resources from the WebSphere Application Server administrative console, use IBM Installation Manager on Windows and UNIX, or a provided script on z/OS, to install the IBM Integration feature into WebSphere Application Server. Use the **manageprofiles** command to configure the WebSphere Application Server administrative console to make available the IBM Integration Administration feature.

Before you begin

If you have previously installed the IBM Integration Administration feature and you want to update it, see “Applying service to IBM Integration Administration for WebSphere Application Server” on page 147.

About this task

Complete the following steps to install and configure the WebSphere Application Server administrative console to make available the IBM Integration Administration feature:

Procedure

1. Install the IBM Integration feature onto your WebSphere Application Server system; see “Installing the IBM Integration feature: Windows, Linux, and UNIX” and “Installing the IBM Integration feature: z/OS” on page 97.
2. Configure the WebSphere Application Server administrative console to make available the IBM Integration Administration feature; see “Configuring IBM Integration Administration for WebSphere Application Server” on page 102.

What to do next

Now you can view or manage your IBM Integration resources from the WebSphere Application Server administrative console; see Administering integration nodes from WebSphere Application Server.

For information about uninstalling IBM Integration Administration for WebSphere Application Server, see “Uninstalling the IBM Integration feature: Windows, Linux, and UNIX” on page 99 and “Uninstalling the IBM Integration Administration feature: z/OS” on page 100.

For information about upgrading IBM Integration Administration for WebSphere Application Server, see “Upgrading IBM Integration Administration for WebSphere Application Server” on page 103.

Related tasks:



Administering brokers and broker resources

Administering brokers and associated integration node resources includes the tasks that you perform frequently to activate and manage those resources. Choose the method you prefer to administer your brokers and associated resources.

“Applying service to IBM Integration Administration for WebSphere Application Server” on page 147

Apply maintenance updates and program fixes to the IBM Integration Administration for WebSphere Application Server feature.

“Upgrading IBM Integration Administration for WebSphere Application Server” on page 103

Upgrade a previous version of the IBM Integration Administration feature on your WebSphere Application Server to the latest version.

Installing the IBM Integration feature: Windows, Linux, and UNIX

Learn about how to use IBM Installation Manager to install IBM Integration Administration for WebSphere Application Server.

Before you begin

If you have previously installed the IBM Integration feature and you want to update it, see “Applying service to IBM Integration Administration for WebSphere Application Server” on page 147.

For more information about the versions of WebSphere Application Server supported by this feature, see the Prerequisites under the detailed system requirements for each supported platform, see IBM Integration Bus Requirements.

Ensure that IBM Installation Manager is installed and configured. If you are working with WebSphere Application Server Version 7.0, you must complete the following steps before you install the IBM Integration feature:

1. Install IBM Installation Manager

2. Configure links to product and service repositories in IBM Installation Manager
3. Import the WebSphere Application Server installation into IBM Installation Manager

See the documentation on IBM Installation Manager, and on installing features, in the WebSphere Application Server Version 7.0 product documentation.

About this task

Install the IBM Integration administration package onto your WebSphere Application Server system by pointing IBM Installation Manager at the WebSphere Application Server repository file that is included in your IBM Integration installation.

Two repository files are included:

- `wmb.admin.repository.zip`. This is the base version of the package.
- `wmb.admin.repository_updates.zip`. This file contains updates to the package.

Note: Add one or both packages to the IBM Installation Manager repositories list, as described below. You must add the base repository at a minimum. If you want to install the latest updates, but you do not want IBM Installation Manager to retrieve them automatically from an external IBM location, then clear the **Search service repositories during installation and updates** check box and use the update package supplied with your installation.

Procedure

To install the administration package, complete the following steps:

1. Stop the WebSphere Application Server server.

Navigate to the `install_root/bin` or `install_root\bin` directory, where `install_root` is the root installation directory for WebSphere Application Server. On a command line, enter the following command:

- On Windows:

```
stopServer server1 -profileName profileName
```

- On Linux and UNIX:

```
stopServer.sh server1 -profileName profileName
```

`server1` is the name of the application server.

`profileName` identifies the profile of the server process in a multi-profile installation. This parameter is not required for a single-profile installation.

2. Open IBM Installation Manager.

For example, on Windows, click **Start > All Programs > IBM Installation Manager > IBM Installation Manager**.

3. Add one or both IBM Integration repository locations to the IBM Installation Manager list of repositories.

- a. On the IBM Installation Manager main panel, click **File > Preferences**.

- b. Click **Repositories** and then **Add repository**.

- c. Click **Browse** and then navigate to the location of the IBM Integration repositories. You can copy the repositories onto your local machine or access them through a shared drive. The files are installed as part of your IBM Integration Bus installation at the following location:

- On Windows: `install_dir\tools\wmb.admin.repository.zip`

- On Linux, UNIX, and z/OS: `install_dir/tools/wmb.admin.repository.zip`
`install_dir` is the location of your IBM Integration Bus installation.
 A file called `wmb.admin.zos.tar` is also installed at the same location as `wmb.admin.repository.zip`. To administer broker resources from WebSphere Application Server on z/OS, you must copy this file onto the z/OS, system. See “Installing the IBM Integration feature: z/OS” on page 97.
- d. Select the repository file and click **Open**.
 - e. Select the check box beside the IBM Integration repository file and clear all other repository file selections. If you do not want Installation Manager to retrieve updates automatically from an external IBM website then ensure that the **Search service repositories during installation and updates** check box is deselected.
 - f. Click **OK**.
4. Install the IBM Integration installation package
 - a. On the main IBM Installation Manager panel, click **Install**. The Install Packages panel is opened.
 - b. On the Install panel, select packages to install. Select the installation package that matches your version of WebSphere Application Server, and click **Next**.
 - c. On the Licenses panel, review and accept the product license. Click **I accept the terms in the license agreement**. Click **Next**.
 - d. If you see a message about running processes that might interfere with the current operation, you must stop the processes before the installation can continue.
 - e. On the Location panel, click **Use the existing package group**, and select **IBM WebSphere Application Server *versionName*** from the list of Package Group Names, where *versionName* is the version of your WebSphere Application Server installation. If you select any other package group, an error message is displayed to say that the product cannot coexist with the selected package group.
Click **Next**.
 - f. On the Features panel, select the features to install. There is only one feature for this installation package. Select the **IBM Integration Bus Administration feature** and click **Next**.
 - g. On the Summary panel, review the installation summary and click **Install** to start the installation.
 - h. When the installation is complete, review the installation log, and click **Finish**. You are prompted to use the **manageprofiles** command to augment your WebSphere Application Server profile; see “Configuring IBM Integration Administration for WebSphere Application Server” on page 102.

What to do next

Now that you have installed the IBM Integration Bus package, configure the administration run time by using the WebSphere Application Server **manageprofiles** command. See “Configuring IBM Integration Administration for WebSphere Application Server” on page 102.

Related tasks:

“Installing the IBM Integration feature: Windows, Linux, and UNIX” on page 94
 Learn about how to use IBM Installation Manager to install IBM Integration Administration for WebSphere Application Server.

“Finding the latest information” on page 1
Access the latest information for IBM Integration Bus.

Installing the IBM Integration feature: z/OS

Learn about how to install IBM Integration Administration for WebSphere Application Server on z/OS.

Before you begin

For information about the versions of WebSphere Application Server that are supported with this feature, visit the IBM Integration Bus Requirements website.

About this task

Install the IBM Integration administration package onto your WebSphere Application Server system by running a script and passing it the location of the WebSphere Application Server feature installation file. The script and the installation file are included in your IBM Integration installation.

Procedure

Before installing the administration package, complete the following preparation steps:

1. Identify the file system locations that are used in the installation. The instructions use the following variables to represent these locations:

WAS_HOME

This term represents the location of your server configuration, and its value is usually assigned to the `$WAS_HOME` environment variable. The path typically ends with `/AppServer`. `$WAS_HOME/bin` is the path that you specify for WebSphere Application Server commands such as `startServer.sh`.

WAS_RO_HOME

This term represents the location of the WebSphere Application Server product installation files that are referenced from the server configuration. Its value is not assigned by default to an environment variable. It is usually defined to be read-only. The path to `WAS_RO_HOME/systemApps/isclite.ear/isclite.jar` typically has no symbolic links.

2. Ensure that your user ID has write access to both `WAS_HOME` and `WAS_RO_HOME`, and has administrative authority to make `WAS_RO_HOME` writable.
3. For WebSphere Application Server V8, you must set the program control on the `bbgjutil` module. A user ID with administrative authorities is required to set this control.

For example:

```
extattr +p WAS_HOME/lib/s390x9-64/bbgjutil  
extattr +p WAS_HOME/lib/s390x-64/bbgjutil
```

4. Copy the installation file `wmb.admin.zos.tar` to a suitable location on your WebSphere Application Server system. The file is installed as part of your IBM Integration Bus installation at the following location:
 - On Windows: `wmqi_install_path\tools\wmb.admin.zos.tar`
 - On Linux, UNIX, and z/OS: `wmqi_install_path/tools/wmb.admin.zos.tar`

where `wmqi_install_path` is the IBM Integration Bus installation directory.

5. Create a directory into which to extract the contents of `wmb.admin.zos.tar`. The creating user ID is automatically given write access to the directory. For example, run the following command:


```
mkdir /tmp/unpack
```
6. Extract the files from `wmb.admin.zos.tar` into the directory that you created. For example, from directory `/tmp/unpack`, enter the following command:


```
tar -xvf wmb.admin.zos.tar
```

 You can now see three new files in this directory:
 - `installBrokerPlugin.sh`
 - `uninstallBrokerPlugin.sh`
 - `wmb.admin.tar`
7. Ensure that the files are executable. For example, run the following command:


```
chmod a+x *.sh
```
8. Create a new directory to permanently contain the IBM Integration Bus installation files. The creating user ID is automatically given write access to the directory. For example, run the following command:


```
mkdir /my/brokerplugin
```

To install the administration package, complete the following steps:

9. First, ensure that any servers created from this WebSphere Application Server installation are stopped.

Navigate to the `WAS_HOME/bin` directory. On a command line, enter the following command:

```
stopServer server1 -profileName profileName
```

`server1` is the name of the application server.

`profileName` identifies the profile of the server process in a multi-profile installation. This parameter is not required for a single-profile installation.
10. Go to the location where the installation files were unpacked. For example, run the following command:


```
cd /tmp/unpack
```
11. If the `WAS_RO_HOME` directory is read-only, set it as writable during the installation. For example, run the following command:


```
chmount -w WAS_RO_HOME
```
12. As a user with administrative authorities, run the installation script `installBrokerPlugin.sh`. This script generates a large amount of output, so direct this output to a file or to a utility such as `more` to read it easily. For example, run the following command:


```
./installBrokerPlugin.sh -u extract_location -d was_install_root -f tar_file
```

 - `extract_location` is the directory into which `wmb.admin.tar` is to be extracted. In this example `/my/brokerplugin` would be used.
 - `was_install_root` is the `WAS_HOME` directory if you are using WebSphere Application Server V7, or the `WAS_RO_HOME` directory if you are using WebSphere Application Server V8.
 - `tar_file` is the full path and file name for `wmb.admin.tar`. In this example `/tmp/unpack/wmb.admin.tar` would be used.

The script returns information about any running WebSphere Application Server processes that are preventing the installation from completing. These processes must be stopped before the script is run again.

13. If the `WAS_RO_HOME` directory was set to writable, set it back to read-only. For example, run the following command:

```
chmount -r WAS_RO_HOME
```
14. Restart any WebSphere Application Server servers that were previously stopped.

What to do next

Now that you have installed the IBM Integration feature, configure the administration run time by using the WebSphere Application Server **manageprofiles** command. See “Configuring IBM Integration Administration for WebSphere Application Server” on page 102.

Related tasks:

“Finding the latest information” on page 1
Access the latest information for IBM Integration Bus.

Uninstalling the IBM Integration feature: Windows, Linux, and UNIX

Learn about how to use IBM Installation Manager to uninstall IBM Integration Administration for WebSphere Application Server.

Before you begin

Before uninstalling the IBM Integration administration feature, you must unaugment or delete any profiles that you augmented or created for the feature.

For example, run the following command on Windows to unaugment a profile called `AppSrv01`:

```
manageprofiles -unaugment -profileName AppSrv01
```

If more than one template has been applied to the profile, you might need to take additional steps. See the documentation on unaugmenting and deleting profiles by using the command line in the WebSphere Application Server documentation.

If you want to roll back to a previously installed version of the IBM Integration administration package. See “Applying service to IBM Integration Administration for WebSphere Application Server” on page 147.

About this task

Remove the IBM Integration administration package from your WebSphere Application Server system by using IBM Installation Manager.

Procedure

To uninstall the administration package, complete the following steps:

1. Stop the WebSphere Application Server server.
Navigate to the `install_root/bin` or `install_root\bin` directory, where `install_root` is the root installation directory for WebSphere Application Server. On a command line, enter the following command:
 - On Windows:

```
stopServer server1 -profileName profileName
```

- On Linux and UNIX:

```
stopServer.sh server1 -profileName profileName
```

server1 is the name of the application server.

profileName identifies the profile of the server process in a multi-profile installation. This parameter is not required for a single-profile installation.

2. Open IBM Installation Manager.

For example, on Windows, click **Start > All Programs > IBM Installation Manager > IBM Installation Manager**

3. Uninstall the IBM Integration Bus installation package
 - a. On the main IBM Installation Manager panel, click **Uninstall**. The Uninstall Packages panel is opened.
 - b. Select the package to uninstall; for example, select **IBM Integration Administration for Version 8 of WebSphere Application Server**. Click **Next**.
 - c. Review the Prerequisites panel, and ensure that you unaugment or delete any profiles that you created for this feature. Click **Next**.
 - d. On the Summary panel, review the summary and click **Uninstall**.
 - e. When the uninstallation is complete, review the log, and click **Finish**.

Related tasks:

“Installing the IBM Integration feature: Windows, Linux, and UNIX” on page 94
Learn about how to use IBM Installation Manager to install IBM Integration Administration for WebSphere Application Server.

“Installing IBM Integration Administration for WebSphere Application Server” on page 93

If you want to view or manage your IBM Integration resources from the WebSphere Application Server administrative console, use IBM Installation Manager on Windows and UNIX, or a provided script on z/OS, to install the IBM Integration feature into WebSphere Application Server. Use the **manageprofiles** command to configure the WebSphere Application Server administrative console to make available the IBM Integration Administration feature.

Uninstalling the IBM Integration Administration feature: z/OS

Learn about how to uninstall IBM Integration Administration for WebSphere Application Server on z/OS.

Before you begin

Before uninstalling the IBM Integration Administration feature, you must unaugment or delete any profiles that you augmented or created for the feature.

For example, run the following command to unaugment a profile called *AppSrv01*:

```
manageprofiles -unaugment -profileName AppSrv01
```

If more than one template has been applied to the profile, you might need to take additional steps. See the documentation on unaugmenting and deleting profiles by using the command line in the WebSphere Application Server documentation.

About this task

Uninstall the IBM Integration administration feature from your WebSphere Application Server system by running a script that is provided with the feature installation:

Procedure

1. Stop all of the WebSphere Application Server servers that were started from the relevant WebSphere Application Server installation.

Navigate to the *was_install_root/bin* directory, where *was_install_root* is the root installation directory for WebSphere Application Server. On a command line, enter the following command:

```
stopServer server1 -profileName profileName
```

server1 is the name of the application server.

profileName identifies the profile of the server process in a multi-profile installation. This parameter is not required for a single-profile installation.

2. Navigate to the location into which you extracted the IBM Integration administration for WebSphere Application Server feature. See “Installing the IBM Integration feature: z/OS” on page 97

For example, run the following command:

```
cd /tmp/unpack
```

3. If the WebSphere Application Server installation directory is read-only, use the `chmount` command to make the directory writable. For example, as a user with administrative authorities, run the following command:

```
chmount -w was_install_root
```

where *was_install_root* is the root installation directory for WebSphere Application Server.

4. As a user with administrative authorities, run the script `uninstallBrokerPlugin.sh`.

For example:

```
./uninstallBrokerPlugin.sh -u extract_location -d was_install_root
```

- *extract_location* is the full path to the directory into which you extracted the IBM Integration administration for WebSphere Application Server feature.
- *was_install_root* is the location of your WebSphere Application Server installation

The script returns information about any running WebSphere Application Server processes that are preventing the uninstallation from completing. These processes must be stopped before the script is run again.

5. As a user with administrative authorities, use the `chmount` command to make the WebSphere Application Server installation directory read-only.
6. Restart your WebSphere Application Server server.

Related tasks:

“Installing the IBM Integration feature: z/OS” on page 97

Learn about how to install IBM Integration Administration for WebSphere Application Server on z/OS.

Configuring IBM Integration Administration for WebSphere Application Server

Learn about how to use the **manageprofiles** command to create or augment profiles in WebSphere Application Server, and about how to enable the IBM Integration web user interface, so that you can use IBM Integration administration for WebSphere Application Server.

Before you begin

- Install the IBM Integration administration feature onto your WebSphere Application Server system; see “Installing the IBM Integration feature: Windows, Linux, and UNIX” on page 94 and “Installing the IBM Integration feature: z/OS” on page 97.
- Read the documentation about profile concepts, and about managing profiles by using commands, in WebSphere Application Server; see WebSphere Application Server documentation.

If you are using the WebSphere Application Server Network Deployment product, read the documentation on profile augmentation rules and limitations in the WebSphere Application Server documentation.

About this task

Complete the following steps:

Procedure

1. Augment or create WebSphere Application Server profiles.

The IBM Integration administration feature is supported for both *dmgr* and *default* profiles. You can use the **manageprofiles** command to create or augment profiles of these types. The profile templates to use are in the *install_root/profileTemplates* or *install_root\profileTemplates* directory, where *install_root* is the WebSphere Application Server root installation directory.

- Use the *default.broker* template to create or augment a *default* profile.

For example, run the following command (after backing up your current configuration) from *install_root\bin* on Windows to augment a *default* profile called *AppSrv01*:

```
manageprofiles -augment -profileName AppSrv01  
-templatePath "C:\Program Files\IBM\WebSphere\AppServer\profileTemplates\default.broker"
```

- Use the *dmgr.broker* template to create or augment a *dmgr* profile.

For example, run the following command from *install_root\bin* on Windows to create a *dmgr* profile called *DMgr01*:

```
manageprofiles -create -profileName DMgr01  
-templatePath "C:\Program Files\IBM\WebSphere\AppServer\profileTemplates\dmgr.broker"
```

Before uninstalling the IBM Integration administration feature, you must unaugment or delete the profiles that you augmented or created for the feature.

For example, run the following command on Windows to unaugment a profile called *AppSrv01*:

```
manageprofiles -unaugment -profileName AppSrv01
```

If more than one template has been applied to the profile, you might need to take additional steps. See the documentation on unaugmenting and deleting profiles by using the command line in the WebSphere Application Server documentation.

2. For each integration node that you want to view in the WebSphere Application Server administration console, you must configure and enable the IBM Integration web user interface.

See [Configuring the web user interface server](#).

The web user interface must be enabled, because the IBM Integration Representational State Transfer (REST) API is used to query integration node resources.

3. Create the web user accounts that are required for access to the web user interface. See [Managing web user accounts](#).
4. Restart your WebSphere Application Server server.

What to do next

You can now view the IBM Integration Bus collection pages in the WebSphere Application Server administrative console. For more information about what you can do from the IBM Integration Bus collection pages, see [Administering integration nodes from WebSphere Application Server](#), and the help topics in the administrative console.

Related tasks:



Administering brokers and broker resources

Administering brokers and associated integration node resources includes the tasks that you perform frequently to activate and manage those resources. Choose the method you prefer to administer your brokers and associated resources.



Configuring the web user interface server

To enable access to broker resources through the web user interface, configure the IBM Integration Bus web user interface server.



Managing web user accounts

You can control a web user's access to broker resources by associating the web user ID with a role, which has security permissions assigned to it.

Related reference:



Parameter values for the webadmin component

Select the objects and properties associated with the webadmin component that you want to change.

Upgrading IBM Integration Administration for WebSphere Application Server

Upgrade a previous version of the IBM Integration Administration feature on your WebSphere Application Server to the latest version.

About this task

Complete the following steps to upgrade the IBM Integration Administration feature:

Procedure

1. Uninstall the version of the IBM Integration feature that is installed onto your WebSphere Application Server system. To uninstall from distributed systems, see [“Uninstalling the IBM Integration feature: Windows, Linux, and UNIX”](#) on page 99. To uninstall from z/OS, see [“Uninstalling the IBM Integration Administration feature: z/OS”](#) on page 100.

2. Install the latest version of the IBM Integration Administration feature that is supplied in this release. To install on distributed systems, see “Installing the IBM Integration feature: Windows, Linux, and UNIX” on page 94. To install on z/OS, see “Installing the IBM Integration feature: z/OS” on page 97.

What to do next

You can view the IBM Integration Bus collection pages in the WebSphere Application Server administrative console. After the upgrade, you do not have to reenter configuration information for existing connections to integration nodes. Configuration data for the integration nodes is still shown in the collection and details pages. For more information about what you can do from the IBM Integration Bus collection pages, see *Administering integration nodes from WebSphere Application Server*, and the help topics in the administrative console.

Related tasks:



Administering brokers and broker resources

Administering brokers and associated integration node resources includes the tasks that you perform frequently to activate and manage those resources. Choose the method you prefer to administer your brokers and associated resources.



Configuring the web user interface server

To enable access to broker resources through the web user interface, configure the IBM Integration Bus web user interface server.



Managing web user accounts

You can control a web user's access to broker resources by associating the web user ID with a role, which has security permissions assigned to it.

Verifying your IBM Integration Bus installation

You can verify your installation of IBM Integration Bus by using the IBM Integration Toolkit or the IBM Integration Explorer.

About this task

This section describes how to verify your installation on Linux on x86, Linux on x86-64, or Windows by using either the IBM Integration Toolkit or the IBM Integration Explorer.

- “Verifying your installation by using the IBM Integration Toolkit” on page 105
- “Verifying your installation using the IBM Integration Explorer” on page 109

After verification, you can keep all the components to carry out further development and unit test. Development and test environments are restricted to Linux on x86, Linux on x86-64, and Windows computers on which one copy of each component is installed on each computer. You can also create additional components and resources on your test computers to investigate the ways in which your business requirements can be met by this product.

If you have purchased IBM Integration Bus Remote Adapter Deployment, Standard Edition, or Express Edition, you must modify the operation mode of the broker after you have completed the procedures in this chapter (unless you are running a test environment). The broker operation mode is always set to the default enterprise when you install the Integration Bus component from the full product packages. If you intend to keep and use the default broker that you create in this chapter, you are required to modify its mode to conform to the terms of your

license. See “Checking the broker operation mode and function level” on page 111 for details of this task.

Verifying your installation by using the IBM Integration Toolkit

Use the instructions in this tutorial to verify your installation of IBM Integration Bus and learn how to run samples with the IBM Integration Toolkit.

Before you begin

You must have installed the IBM Integration Toolkit to run this tutorial.

About this task

To verify your installation by using the IBM Integration Toolkit, complete the following tasks:

- Create a Default Configuration.
- Run the Pager samples.
- (Optional) Start the Samples Preparation wizard.
- Remove the samples.

Before you can run sample programs, you must use the Default Configuration wizard to create a broker, which has a fixed name and properties that the samples depend on.

The Default Configuration wizard requires that the following conditions are met:

- You have installed the Integration Bus component and the IBM Integration Toolkit.
- None of the Default Configuration wizard resources exist (the resources are listed in the table that is included later in this section).
- This configuration is required for test and evaluation purposes only.
- Your current user ID has the following characteristics on Windows only:
 - It is a member of groups mqbrkrs and mqm.
 - It has Administrator authority.
 - It is a local ID, not a domain ID.

For more information about these security requirements, see Integration Bus server security.

You cannot complete the configuration and verification described here if the previously described conditions are not met.

Use the following instructions to complete these tasks:

Procedure

1. Start the IBM Integration Toolkit:

- On Windows:

On Windows, you cannot complete verification unless you have Administrator authority; carry out verification with the same user ID that you used to complete installation.

If you did not launch the IBM Integration Toolkit from the installation wizard, launch it from the **Start** menu, or run the script file provided. On a command line, navigate to the root directory of the package group and enter the following command:

launcher.bat

The script file runs the following command; if you prefer you can use this command yourself:

```
eclipse.exe -product com.ibm.etools.msgbroker.tooling.ide
```

- On Linux:

On Linux on x86 and Linux on x86-64, you do not need root authority to complete verification. You cannot launch the IBM Integration Toolkit from the installation wizard because you might experience problems during operation if you were to create resources such as brokers when you are logged in as root, and this option is therefore unavailable.

Log off from the user ID with which you have installed the product. Log in as the same ID (if ID is not root), or log in as another ID, but do not become root.

Launch the IBM Integration Toolkit from the main menu or run the script file provided. On a command line, navigate to the root directory of the package group and enter the following command:

```
./launcher
```

If you prefer, you can also run the application directly:

```
./eclipse -product com.ibm.etools.msgbroker.tooling.ide
```

However, you must set the `LD_LIBRARY_PATH` before running the application. For details about how to set `LD_LIBRARY_PATH`, see the launcher script.

When you first launch the IBM Integration Toolkit, you are asked to specify the location of your workspace. This directory exists on your local drive, and is where the IBM Integration Toolkit stores all the resources that you create. You can accept the default directory shown, or you can specify your own choice either by typing it in, or by clicking **Browse** to specify the location. Select **Use this as the default and do not ask again** to inhibit the display of the workspace dialog next time you launch the IBM Integration Toolkit.

The IBM Integration Toolkit opens and the Welcome page is displayed.



2. Click the **Get Started** icon to begin the configuration and verification process.

The Get Started page opens, from which you can create a default configuration that is used by a sample program to verify that your installation is successful.

3. Create the Default Configuration:
 - a. On the Get Started page, click the link to **Create the Default Configuration**.



The "Creating the Default Configuration" page opens.

- b. Click **Start the Default Configuration wizard**.

Follow the guidance provided by the wizard to navigate through its pages. The wizard creates a default broker that can be used by a sample program to verify that your installation is successful.

The wizard displays a progress bar that shows which task it is currently performing. It also reports on all the actions that it takes by writing progress information into a scrollable text window from which you can copy and paste some or all of the information displayed.

The information in the text window is also written to a log file in your workspace directory structure. The default workspace directory is shown in the example, but you can choose another location when you start the IBM Integration Toolkit.

Linux on x86 and Linux on x86-64

```
user_home_dir/IBM/IntegrationToolkit90/workspace/.metadata/
DefaultConfigurationWizard.log
```

Windows

```
user_home_dir\IBM\IntegrationToolkit90\workspace\.metadata\
DefaultConfigurationWizard.log
```

If the wizard encounters an error in processing, it informs you of what has happened and returns any error information, for example a return code from a command. If you know why the error has occurred from the error text, and can correct the situation, you can do so now. Return to the error message display and click **Yes** to continue the wizard.

If you do not understand the error, and do not know how to fix it, click **No**. If the wizard can, it rolls back all the actions that it has taken so far, so that when it completes, your system is in the same state as it was before you started the wizard. The text window shows you exactly what the wizard has, and has not, done.

Click **Open Log File** to access the log from the summary page of the wizard; this option is available whether the wizard has succeeded or failed.

The wizard creates the resources shown in the following table.

Table 7. Resources created by the Default Configuration wizard

Name	Type
IB9NODE	Broker
IB9QMGR	WebSphere MQ queue manager that hosts the broker. The queue manager has a listener at the first available port greater than or equal to 2414.

It also starts the broker so that it is ready to process a sample.

- c. On the final page, ignore the option to start the Samples Preparation wizard; you start this wizard later in these instructions.
 - d. Click **Finish** to close the wizard. When the wizard completes, it opens the Integration Development perspective and displays the resources that the wizard has created.
4. To verify your installation, click **Help > Samples and Tutorials > IBM Integration Toolkit - IBM Integration** to open the Samples and Tutorials panel. The Samples and Tutorials panel can also be opened from the Welcome page.
- a. Expand **Application Samples**, and click **Pager Samples** to open the Pager samples page. The following options are displayed:
 - **Import and deploy the Pager samples**
This option starts the Samples Preparation wizard, which helps you to import the samples into your workspace, and to deploy the samples and associated resources (for example, message flows) to the default broker.
 - **Run the Pager samples**
This option opens the help page that contains a description of each of three sample programs, and icons that you can click to start each one.
 - **Find out what the Pager samples do**

This option opens a page that describes in detail what the Pager samples do and how they work. You can examine the message flows that implement the sample function, and the messages that are handled by those flows.

- b. Click **Import and deploy the Pager samples**. The Samples Preparation wizard starts and displays its first page. The option to import and deploy to the default broker is preselected.
- c. Click **Next** and follow the guidance provided by the wizard to navigate through its pages.

The wizard displays a progress bar that shows which task it is currently performing. It also reports on all the actions that it takes by writing progress information into a scrollable text window.

You can copy and paste some or all of the information that is reported in this text window. This information is also written to the following log file:

Linux on x86 and Linux on x86-64

```
user_home_dir/IBM/IntegrationToolkit90/workspace/.metadata/  
samplePreparationWizard.log
```

Windows

```
user_home_dir\eclipse\workspace\.metadata\  
samplePreparationWizard.log
```

If the wizard encounters an error in processing, it informs you of what has happened and returns any error information, for example a return code from a command. If you know why the error has occurred from the error text, and can correct the situation, you can do so now. Return to the error message display and click **Yes** to continue the wizard.

If you do not understand the error, and do not know how to fix it, click **No**. If the wizard can, it rolls back all the actions that it has taken so far, so that when it completes, your system is in the same state as it was in before you started the wizard. The text window shows you exactly what the wizard has, and has not, done.

The wizard displays information messages to show that the Pager samples and associated resources are deployed and ready to run.

- d. Click **Next** when you have read the messages about the actions that have been completed by the wizard. The confirmation page is displayed.
 - e. Click **Finish** to close the wizard. The "Pager samples" page (from which you launched the wizard) is redisplayed.
 - f. Click **Run the Pager samples**. On the page that opens, click **How to use the applications** to read about the Text Messenger and Surf report publisher applications. When you understand what the applications do, and how to use them, click the icon that represents the application that you want to run. If you want more detailed information about the contents of these applications, and how the message flows work, click **Find out what the Pager samples do**.
 - g. When you have sent and received messages successfully, you have verified that your installation is complete. You can now close your Pager applications and the Samples Gallery.
5. (Optional) You can start the Samples Preparation wizard to create the resources and start other supplied sample programs. Click **File > New > Other > IBM Integration Bus - Default Configuration and Samples** in the IBM Integration Toolkit, and select **Prepare the Samples**. The Samples Preparation wizard opens, and lists other samples that are available.

You can view information about samples only when you use the information center that is integrated with the IBM Integration Toolkit or the online information center. You can run samples only when you use the information center that is integrated with the IBM Integration Toolkit.

6. To remove the sample or samples when you have finished with them, run the Samples Preparation wizard again and remove the samples that you have added. This action removes the samples from the broker, and removes the sample resources from your workspace.

When you have completed your verification tests, run the Default Configuration wizard to remove all the default resources. Use the same workspace and the same user ID that you used to create the resources. To start the wizard from the IBM Integration Toolkit, click **File > New > Other** and expand IBM Integration Bus - Default Configuration and Samples. Select **Create the Default Configuration** and click **Next**.

Results

You have completed the tutorial.

Related tasks:

“Creating a default configuration” on page 274

Use the Default Configuration wizard to create and test a basic broker configuration.



Dealing with problems

Learn how to resolve some of the typical problems that can occur.

Verifying your installation using the IBM Integration Explorer

Use the instructions in this tutorial to verify your installation of IBM Integration Bus and learn how to deploy broker archive files with the IBM Integration Explorer.

Before you begin

You must have installed the IBM Integration Explorer to run this tutorial. If you are using the IBM Integration Toolkit you can verify your installation using one of the samples. For a list of samples, see Samples.

About this task

To verify your installation using the IBM Integration Explorer you can complete the following tasks:

- Create a Default Configuration
- Deploy a broker archive file
- Check the results of a deployment
- Test the message flow

Use the following instructions to complete these tasks:

Procedure

1. To use the IBM Integration Explorer you must start the WebSphere MQ Explorer. To start the WebSphere MQ Explorer, complete one of the following steps:
 - On Windows:

Click **Start > All Programs > IBM Integration Bus 9.0.0.0 > IBM Integration Explorer**, or open a command line in which the **mqsiprofile** command has not been run and enter the **strmqcfg** command.

- On Linux:

Enter the **strmqcfg** command on a command line in which the **mqsiprofile** command has not been run, or run **/usr/bin/strmqcfg**.

2. Create the Default Configuration:

- Start the Create the Default Configuration wizard using the following link:
Start the Default Configuration wizard
You can use this link only when you use the information center that is integrated with the IBM Integration Toolkit or the IBM Integration Explorer.
- Alternatively, you can use the following instructions to start the Create the Default Configuration wizard:
 - a. In the **WebSphere MQ Explorer - Navigator** view, click the **Integration Nodes** folder.
 - b. In the **WebSphere MQ Explorer - Content** view, click the **Create the Default Configuration** button. The Create the Default Configuration wizard is displayed.
 - a. You can click **Cancel** at any time to cancel the creation of the default configuration.
 - b. The Default Configuration Summary page lists the resources created. Click **Next** to continue.
 - c. Click **Finish**.

The default integration node is created and started. The integration node is also added to the **WebSphere MQ Explorer - Navigator** view.

3. Deploy the broker archive file to the integration node:

- a. Expand the **Broker Archive Files** folder.
- b. Expand the **Getting Started** folder.
- c. Right-click the **pager.bar** file, and click **Deploy File**.
- d. Select the **IB9NODE** integration node, and click **Finish**. The broker archive file is deployed to the integration node.

4. Check the results of the deployment:

- a. In the **WebSphere MQ Explorer - Navigator** view, expand the **Integration Nodes** folder.
- b. Click **IB9NODE** to select the integration node.
- c. View messages from the integration node in the Administration Log view. If the deployment was successful, message **BIP2881I** is displayed in the Administration Log view. You can also see the **TextMessenger** message flow and the **PagerMessageSets** message set deployed to the Default integration server under the **IB9NODE** integration node.

5. Test the message flow:

- a. In the **WebSphere MQ Explorer - Navigator** view, expand **IB9QMGR** in the **Queue Managers** folder.
- b. Right-click the **Queues** folder, and click **New > Local Queue**.
- c. Enter **TEXTMESSENGER** as the name for the queue, and click **Finish**. This queue is the input queue for the Pager message flow.
- d. Click **OK**.
- e. Right-click the **Queues** folder, and click **New > Local Queue**.

- f. Enter `TEXTMESSENGER_FAIL` as the name for the queue, and click **Finish**. This queue is the failure queue for the Pager message flow.
- g. Click **OK**.
- h. Right-click the **Queues** folder, and click **New > Local Queue**.
- i. Enter `PAGER` as the name for the queue, and click **Finish**. This queue is the output queue for the Pager message flow.
- j. Click **OK**.
- k. In the **WebSphere MQ Explorer - Content** view right-click the `TEXTMESSENGER` queue, and click **Put Test Message**. The Put test message dialog is displayed.
- l. Enter the following message in the message data field, and click **Put message** to put the test message to the input queue for the Pager message flow:


```
<Pager><text>This is my message to the pager.</text></Pager>
```

The test message is passed through the message flow, transformed, and the output message is put to the Pager queue.

- m. Click **Close** to close the Put test message dialog.
- n. Right-click the Pager queue, and click **Browse Messages**. The Message browser dialog opens and displays any messages on the queue. If the test is successful, the queue shows a message like the following:


```
<?xml version="1.0"?><!DOCTYPE Pager>
<!--MRM Generated XML Output on :Sat Jun 20 10:38:56 2009-->
<Pager><Text> Powered by IBM.</Text></Pager>
```

6. Click **Close** to close the Message browser dialog.

Results

You have completed the tutorial.

Related concepts:

IBM Integration Explorer

The IBM Integration Explorer is a graphical user interface based on the Eclipse platform for administering your integration nodes (brokers).

Related tasks:

“Creating a default configuration” on page 274

Use the Default Configuration wizard to create and test a basic broker configuration.

Administering brokers and broker resources

Administering brokers and associated integration node resources includes the tasks that you perform frequently to activate and manage those resources. Choose the method you prefer to administer your brokers and associated resources.

Dealing with problems

Learn how to resolve some of the typical problems that can occur.

Checking the broker operation mode and function level

You must ensure that your production brokers conform to the terms of your license. You can also change the function level to enable the use of nodes that are supplied in the latest fix pack.

About this task

The following tasks are described in this topic:

- “Configuring your brokers to conform to your license”
- “Changing the function levels of your brokers” on page 113

For a full description of the broker operation mode, see “Operation modes” on page 5.

Configuring your brokers to conform to your license Procedure

• Upgrading from IBM Integration Bus for Developers (Developer Edition)

If you installed the Developer Edition, and you have now purchased a fully supported version of the product such as the Advanced Edition, you can keep the components and all the associated resources that you have already created and configured. Before you install the purchased product, uninstall the Developer Edition. You can retain your existing brokers and workspace, and any Developer mode brokers remain in Developer mode by default. You can leave these brokers in Developer mode, or you can use the `mqs imode` command to change the operation mode to reflect the license that you have purchased.

To change the operation mode, complete the following steps.

- If you intend to keep the brokers that you created with the Developer Edition for further development and unit test by your developers, change the operation mode of each broker to advanced, as described in “Changing the operation mode of your broker” on page 356.

Check the license agreement file to ensure that your configuration conforms to any restrictions for development and unit test. Development and unit test conditions are described in “License requirements” on page 641.

- If you intend to use the brokers that you created with Developer Edition for production purposes, change the operation mode to conform to the license that you have purchased:
 - If you have purchased Remote Adapter Deployment, change the operation mode to adapter.
 - If you have purchased Standard Edition, change the operation mode to standard or scale.
 - If you have purchased Express Edition, change the operation mode to express.
 - If you have purchased a license for Scale mode, change the operation mode to scale.
 - If you have purchased the full (unrestricted) license, change the operation mode to advanced.

If you reinstall IBM Integration Bus with the new physical or electronic packages for your purchased product, the default operation mode of all new brokers that you create has the value advanced. If you have purchased a full license, you can leave the operation mode as advanced; in all other cases you must change the mode of your brokers to conform to the license that you have purchased (standard, adapter, scale, or express).

• Remote Adapter Deployment, Standard Edition, Express Edition, or Scale mode

If you have purchased IBM Integration Bus Remote Adapter Deployment, IBM Integration Bus Standard Edition, IBM Integration Bus Express Edition, or a

license for IBM Integration Bus Scale mode, and installed components from the full runtime package, read the following information.

- All brokers that you have created (for example, by completing the verification procedures) have an operation mode set to advanced, which is the default setting for this installation. You can keep these brokers for further development and unit test, subject to any restrictions that apply for unit test environments, as indicated in your license. Development and unit test conditions are described in “License requirements” on page 641.
 - If you intend to use these brokers for production purposes, follow the instructions in “Changing the operation mode of your broker” on page 356, to conform to the license that you have purchased.
 - If you have purchased Remote Adapter Deployment, change the operation mode to adapter.
 - If you have purchased Standard Edition, change the operation mode to standard or scale.
 - If you have purchased Express Edition, change the operation mode to express.
 - If you have purchased a license for Scale mode, change the operation mode to scale.
 - When you create new brokers, you must set the operation mode to conform to the license that you have purchased, because by default the operation mode is set to the value advanced. You can use the **-o** parameter on the **mqsicreatebroker** command to override this default value by specifying an alternative value of adapter, standard, express, or scale. Alternatively, you can change the mode by using the **mqsimode** command for the created broker. On z/OS you can change the operation mode only by using the **mqsimode** command.
- **Full (unrestricted) license for IBM Integration Bus**

If you have purchased a full (unrestricted) license and installed components from the full runtime package, all brokers that you create have an operation mode set to the default value of advanced, which is the correct setting for your license.

Whenever you create a new broker, on existing installations or on different installations or computers, the operation mode is set to advanced, and you do not need to change this value.

You can continue to work with all the components and associated resources that you have already created. For example, after you have completed verification, you might keep the resources that you have created for further development and unit test (subject to any restrictions that apply for unit test environments, as indicated in your license). Development and unit test conditions are described in “License requirements” on page 641.

Changing the function levels of your brokers

About this task

When a fix pack is delivered, it might include new nodes that you can add to your message flows to provide specific functions.

For Version 9.0.0.0, the function level of new brokers is set to a default value that is equivalent to the value 9.0.0.0. At this level, nodes that might be added by later fix packs are not supported by the broker.

Nodes that are added in later fix packs are available in the IBM Integration Toolkit, and you can include these nodes in a message flow. You can deploy the message flow to a broker only if you have set the function level of that broker to the value that represents the fix pack in which the nodes are first delivered.

Because you can control the function level of each broker, you can try out new nodes on test brokers without affecting the operation of your production brokers. When you are satisfied that the nodes provide the function that you require, and work as you expect, you can set the function level of other brokers in your domain, when appropriate.

To change the function level of a broker, use the **mqsichangebroker** command, specifying the **-f** flag with the appropriate value.

For more information about nodes added in fix packs, and the use of the **mqsichangebroker** command, see **mqsichangebroker** command.

Related concepts:

“Operation modes” on page 5

The operation mode that you can use for your broker is determined by the license that you purchase.



IBM Integration features

Some features of IBM Integration Bus have specific requirements, such as a specific mode of operation or the availability of an additional product, such as a database. Additional terms and conditions also apply to the use of some IBM Integration Bus features, including the requirement for other product licenses.

Related tasks:

“Changing the operation mode of your broker” on page 356

Change the operation mode in which your broker is working by using the **mqsimode** command.

Related reference:

“License requirements” on page 641

Use the reference information in this section to understand license requirements.



mqsimode command

Use the **mqsimode** command to configure and retrieve operation mode information.

Installing complementary products

IBM Integration Bus works with several other products to provide complementary services.

About this task

If you want to use these optional services in your IBM Integration Bus environment, refer to the following installation information:

- IBM Tivoli License Manager
- Eclipse plug-ins from another vendor
- “Publishing in a Citrix XenApp environment” on page 116

Related tasks:

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Software requirements” on page 625

View the operating system, database, and other software requirements.

“Hardware requirements” on page 620

View the processor and related hardware requirements on all platforms.

“Installation and uninstallation interfaces” on page 652

You can use different interfaces for installation and uninstallation of the Integration Bus component and the IBM Integration Toolkit.

Installing IBM License Metric Tool

IBM License Metric Tool (ILMT) enables you to monitor the use of IBM (and other) software products.

About this task

Use ITLM to perform the following software auditing functions:

- Monitor the licenses used by different machines.
- Help keep unnecessary licenses to a minimum.
- Guard against software license compliance problems.

Ensure that you choose the correct ILMT license for the IBM Integration Bus edition that you have purchased. See “Operation modes” on page 5.

If you are using one or more of the WebSphere Adapters with IBM Integration Bus, you must activate ILMT to include those adapters, for example in monitoring activities. If you require this support, follow the instructions in Activating IBM License Metric Tool for WebSphere Adapters.

To find out more about using ILMT to monitor usage of IBM Integration Bus and other IBM products, or to purchase ILMT, see the IBM License Metric Tool website.

For information about installing this product, see the IBM License Metric Tool product documentation.

Related tasks:

“Installing complementary products” on page 114

IBM Integration Bus works with several other products to provide complementary services.



Activating IBM License Metric Tool for WebSphere Adapters

If you want to use IBM License Metric Tool (ILMT), you must activate it for each of the WebSphere Adapters.



Configuring environments without the broker component installed

Install and run Java independent programs that use the IBM DFDL API on systems that do not have the IBM Integration Bus runtime component installed.

Related reference:



System management interfaces

The brokers provide a service for independent system management agents.

“Software requirements” on page 625

View the operating system, database, and other software requirements.

“Hardware requirements” on page 620

View the processor and related hardware requirements on all platforms.

Installing plug-ins using Eclipse Update Manager

Use Eclipse Update Manager to install and update plug-ins provided by partners and other software vendors.

About this task

Do not use the Eclipse Update Manager to update or install plug-ins that are managed by IBM Installation Manager. Always use Installation Manager to work with products that are installed with Installation Manager. For other products, use Eclipse Update Manager, but follow the restrictions and guidance provided:

- Do not delete the `platform.xml` file to prompt Eclipse to regenerate the file. The regenerated contents will be incomplete and therefore invalid.
- Always save a backup copy of your `platform.xml` file before you make any changes to your configuration, for example by using Eclipse Update Manager to install or update plug-ins that are provided by another software vendor.
- When you use Eclipse Update Manager (graphical interface or command line), always specify the package group directory as the destination; never specify the shared resource directory. The shared resource directory is managed by Installation Manager and is reserved for its use.
- Do not use Eclipse Update Manager to search for updates for features that you have installed using Installation Manager.
- Do not select **Clean up all configuration errors** in the Eclipse Update Manager interface; this option might lose information that is maintained by Installation Manager.

Related tasks:

“Installing complementary products” on page 114

IBM Integration Bus works with several other products to provide complementary services.

Related reference:



System management interfaces

The brokers provide a service for independent system management agents.

“Software requirements” on page 625

View the operating system, database, and other software requirements.

“Hardware requirements” on page 620

View the processor and related hardware requirements on all platforms.

Publishing in a Citrix XenApp environment

Supply application location and user details to Citrix to publish an IBM Integration Bus command console or IBM Integration Toolkit.

About this task

Read the Citrix documentation for general information about publishing applications. Publishing does not involve any tasks within IBM Integration Bus; you complete the following tasks in the Citrix Presentation Server console.

“Publishing an IBM Integration Bus command console”

“Publishing an IBM Integration Toolkit” on page 117

“Publishing applications using the CMP” on page 117

Publishing an IBM Integration Bus command console:

About this task

The following steps show how to set the Application Location and User information to publish an IBM Integration Bus command console.

Procedure

1. Set the **Application Type** to Application.
2. Set the **Command Line** to the following value: This example assumes that you have installed a 32-bit operating system.

```
C:\WINDOWS\system32\cmd.exe /k title IBM Integration  
Bus 9.0  
    &&"install_dir\MQSI\9.0\bin\mqsiprofile.cmd"
```

On Windows, you can install the Integration Bus component in multiple locations. Each one can have a different level of service (fix pack) applied and can support a certain set of brokers. Each installation has its own command console executable. To publish command consoles corresponding to different runtime environments, modify the path to `mqsiprofile.cmd` appropriately.

3. Set the **Working Directory** to `install_dir\MQSI\9.0`.
4. Define the users who are authorized to use the IBM Integration Bus command console. This must be a subset of users or groups who are authorized to use the command console locally, on the server.

Attention: Users who have access to the IBM Integration Bus command console through Citrix can use the console to run other commands that are unrelated to the broker.

Publishing an IBM Integration Toolkit:

About this task

To publish an IBM Integration Toolkit, set the Application Location and User information as follows:

Procedure

1. Set the **Application Type** to Application.
2. Set the **Command Line** to the package group directory for IBM Integration Toolkit that your system uses, for example:

```
E:\IntegrationToolkit90\eclipse.exe
```

3. Set the **Working Directory** to the appropriate workspace, for example:

```
E:\workspace\IntegrationToolkit90\workspace1
```

When the IBM Integration Toolkit is run from Citrix, a window inviting the user to select a workspace is displayed. This is the same window that you see when you run the IBM Integration Toolkit locally. At this stage you can specify a different workspace, for which you have the necessary Windows read and write file permissions on the server.

4. Define the users who are authorized to use the IBM Integration Bus command console. This must be a subset of users or groups who are authorized to use the command console locally, on the server.

Publishing applications using the CMP:

About this task

You can publish a Java application that uses the CMP programming interface. The steps are similar to publishing an IBM Integration Toolkit or command console.

Example

For example, to publish the CMP Exerciser sample, publish this file:

```
install_dir\sample\ConfigManagerProxy\StartConfigManagerProxyExerciser.bat
```

Related concepts:

“Users and Citrix”

Review the categories of user that might want to use with Citrix, and how to configure for concurrent multi-users.

Related tasks:



Developing applications that use the IBM Integration API

Develop Java applications that use the IBM Integration API (also known as the CMP) to communicate with, deploy to, and manage brokers and their associated resources.

Related reference:

“System requirements for Citrix XenApp” on page 640

This topic gives information about licensing issues, and the software and hardware that you need to use IBM Integration Bus in a Citrix XenApp environment.

Users and Citrix:

Review the categories of user that might want to use with Citrix, and how to configure for concurrent multi-users.

Categories of user

The following categories of user exist:

Application developers

These users do not need to run mqsi commands, but must be placed in an access control list (ACL) before they can use a broker. Put these users only in the user list for publishing IBM Integration Toolkit.

Administrators

If you use Citrix to publish the IBM Integration Toolkit and the IBM Integration Bus command console, put all administrators in the users lists for both applications. For information about the permissions that administrators need, see Security requirements for administrative tasks.

You can create Windows groups that correspond to the application developer and administrator categories, put the groups in the appropriate user lists, then allocate users to the relevant Windows group. Alternatively, you can give appropriate permissions directly to the user.

Make all users members of the Remote Desktop Users group, so that they can use Terminal Server.

Attention: Users who have access to the IBM Integration Bus command console through Citrix can use the console to run other commands that are unrelated to the broker.

For information about the rights of groups such as mqm, mqbrkrs, and Administrators, see Security requirements for Windows systems.

Concurrent multi-users

Allowing multiple concurrent users to access IBM Integration Bus by Citrix on a Windows server requires the same planning as allowing multiple concurrent users to access the IBM Integration Bus on a Linux or UNIX server. For example:

- All application developers should have their own integration server, so that they can deploy message flows and start and stop integration servers without conflicting with each other during development and testing. However, an execution group might be administered by more than one administrator. In these cases, administrators should coordinate their activities so that they do not conflict with each other. For example, when one administrator is starting an integration server, no one else should be trying to stop it.
- All users should have their own workspace. The Windows system administrator should give each user the appropriate read and write permissions for the directory that corresponds to their workspace.
- The IBM Integration Toolkit user can create the default configuration from the Samples; the product samples are deployed to the default configuration. Because you can create only one default configuration, only one user should use it at one time.

You can view information about samples only when you use the information center that is integrated with the IBM Integration Toolkit or the online information center. You can run samples only when you use the information center that is integrated with the IBM Integration Toolkit.

Related tasks:

“Publishing in a Citrix XenApp environment” on page 116

Supply application location and user details to Citrix to publish an IBM Integration Bus command console or IBM Integration Toolkit.



Creating user IDs

When you plan the administration of your broker configuration, you might have to define one or more user IDs for the tasks associated with particular roles.

Related reference:

“System requirements for Citrix XenApp” on page 640

This topic gives information about licensing issues, and the software and hardware that you need to use IBM Integration Bus in a Citrix XenApp environment.



Security requirements for Windows systems

Security requirements depend on the administrative task that you want to perform.

Setting up a command environment

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

About this task

Also complete this task if you have migrated to IBM Integration Bus Version 9.0 from an earlier version. A profile is provided to help you set up the environment.

If appropriate, you can extend the initialization performed by this profile; for example, for databases, or for other products that you want to use within the broker.

Ensure that you use this environment each time you run an administrative command, or start a broker.

For information about setting up your command and runtime environment on a Windows system, see “Command environment: Windows systems.”

For information on setting up your command and runtime environment on Linux and UNIX systems, see “Command environment: Linux and UNIX systems” on page 124.

For information about setting up your command and runtime environment for brokers on Linux and UNIX systems, see “Broker-specific command environment: Linux and UNIX systems” on page 127.

For information about setting up your command and runtime environment for integration servers on Linux and UNIX systems, see “Integration server-specific command environment: Linux and UNIX systems” on page 128.

For information about setting up your command and runtime environment for brokers on Windows systems, see “Broker-specific command environment: Windows systems” on page 122.

For information about setting up your command and runtime environment for integration servers on Windows systems, see “Integration server-specific command environment: Windows systems” on page 123.

Related tasks:

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188
Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206
Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 6.1 WebSphere Message Broker component” on page 225
Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

“Enabling ODBC connections to the databases” on page 369

Set up the resources and environment that the broker requires for Open Database Connectivity (ODBC) connections to databases on distributed systems.

“Command environment: Windows systems”

Set up the Windows environment to run IBM Integration Bus commands.

“Command environment: Linux and UNIX systems” on page 124

Set up the Linux or UNIX environment to run IBM Integration Bus commands.

Related reference:



Runtime commands

The topics in this section describe the IBM Integration Bus runtime commands.

Command environment: Windows systems

Set up the Windows environment to run IBM Integration Bus commands.

About this task

On Windows systems, the components run as services, therefore they do not inherit the environment that is set for the command prompt from which you start them. The components run the **mqsiprofile** command when they start, which completes the environment initialization.

The **mqsiprofile** command is located in the directory *install_dir\bin*.

You must not change the location of the **mqsiprofile** command, or make user modifications to the command, because it might be replaced if you install service, or an update, to the product.

If you need to run your own settings, add one or more command files (with the file extension *.cmd*) to the directory *work_path\Common\profiles*, where *work_path* identifies the machine-wide IBM Integration Bus working directory.

To verify the machine-wide IBM Integration Bus working directory, enter the following command in a command console:

```
echo %MQSI_WORKPATH%
```

When **mqsiprofile** runs, it automatically calls any additional user-written scripts in this directory.

A typical reason for adding scripts is to configure access to databases from deployed message flows. For more information, see “Running database setup scripts.”

When you have configured any additional setup, you can use one of the following methods to initialize the runtime environment for components and commands:

Procedure

1. If you have only one installation of the runtime product, open a command console by clicking **Start > All Programs > IBM Integration Bus 9.0 > Command Console**. The **mqsiprofile** command initializes the environment and invokes any additional scripts in the *common\profiles* directory.
2. If you have one or more installations of the runtime product, open a command prompt window. Locate and run the *mqsiprofile.cmd* script in the directory in which you installed the appropriate product. The **mqsiprofile** command initializes the environment and invokes any additional scripts in the *common\profiles* directory.

What to do next

Check whether the following conditions apply to your environment:

- If you have a previous version of the product on this system, ensure that you run the correct profile before using Version 9.0. The **mqsiprofile** command places the Version 9.0 commands and libraries at the front of your search path, and invokes any user profiles that you have supplied which can override any combination of *PATH*, *CLASSPATH*, or *library PATH*.
- If you use the same user ID, and you run multiple profiles (from multiple different installations or versions), you might get unexpected results. Log off and log on again before you run the specific profile that you require.

Running database setup scripts:

About this task

A broker might require access to databases from deployed message flows.

When you install a database product on Windows, the relevant settings are typically made to the system environment. However, some database managers provide a profile to perform this setup, or provide details of actions that you must take in their documentation. Always check the database product documentation for environment setup details; the information provided here is for general guidance only.

If a profile is provided for the database that you are using, complete the following steps:

Procedure

1. If you can update the profile to provide permanent values for the details that are required (for example, the database server name or the installation directory):
 - a. Complete the changes to the database profile.
 - b. Copy the profile file to the directory `work_path\Common\profiles`.
2. If you cannot update the profile permanently, but need to make changes each time, you must run it independently of the `mqsiprofile` command before you start the component.

What to do next

When your environment has been set up, see Working with databases for information about setting up your databases for use with the broker.

Related tasks:



Working with databases

Create and configure databases to use with your message flows.

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.



Changing the location of the work path

The work path directory is the location where a component stores internal data, such as installation logs, component details, and trace output. The shared-classes directory is also located in the work path directory and is used for deployed Java code. If the work path directory does not have enough capacity, redirect the directory to another file system that has enough capacity.

“Command environment: Linux and UNIX systems” on page 124

Set up the Linux or UNIX environment to run IBM Integration Bus commands.

Related reference:

“Additional software requirements” on page 633

IBM Integration Bus requires additional software products to run successfully.



Runtime commands

The topics in this section describe the IBM Integration Bus runtime commands.

Broker-specific command environment: Windows systems

Extend or change the Windows environment that is used when you run a specific broker.

About this task

If you want to run your own additional environment settings for a broker, add one or more command files (with the file extension `.cmd`) to the `work_path\config\broker_name\profiles` directory, where `work_path` is the machine-wide IBM Integration Bus working directory, and `broker_name` is the name of your broker.

To verify the machine-wide IBM Integration Bus working directory, enter the following command in a command console:

```
echo %MQSI_WORKPATH%
```

Ensure that the broker name, and any integration servers that you create, contain only characters that are valid on your file system. You might also have to create the required directory structure.

All files that are placed into the `work_path\config\broker_name\profiles` directory, and that have a `.cmd` extension, are processed when the broker starts.

Scripts in the `work_path\config\broker_name\profiles` directory are run after the **mqsiprofile** script, and any scripts in the `work_path\Common\profiles` directory have run.

To diagnose any problems, a log file is written that lists the profile scripts that have been read, and the environment that is used by this broker. The log file location is `work_path\Common\log\broker_name.profilelog`.

Integration server profile scripts are not removed automatically when an integration server is deleted. Therefore, if you have created any integration server profile files for this broker, you must delete these profile files manually if they are no longer required.

Broker profile scripts are not removed automatically when a broker is deleted. If you no longer need the profile files that you have created, delete them manually.

Related tasks:

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

“Command environment: Windows systems” on page 120

Set up the Windows environment to run IBM Integration Bus commands.

“Integration server-specific command environment: Windows systems”

Extend or change the Windows environment that is used when you run a specific IBM Integration Bus.

Integration server-specific command environment: Windows systems

Extend or change the Windows environment that is used when you run a specific IBM Integration Bus.

About this task

To enable function that becomes available in IBM Integration Bus fix packs, use the `-f` parameter on the **mqsichangebroker** command. For more information, see **mqsichangebroker** command.

To run your own additional environment settings for a broker, add one or more command files (with the file extension `.cmd`) to the `work_path\config\broker_name\server_name\profiles` directory, where `work_path` is the machine-wide IBM Integration Bus working directory, `broker_name` is the name of your broker, and `server_name` is the name of your integration server.

To verify the machine-wide IBM Integration Bus working directory, enter the following command in a command console:

```
echo %MQSI_WORKPATH%
```

Ensure that the broker name, and any integration servers that you create, contain only characters that are valid on your file system. You might also have to create the required directory structure.

All files that are placed into the `work_path\config\broker_name\server_name\profiles` directory that have a `.cmd` extension, are processed when the integration server (`server_name`) on the broker (`broker_name`) starts.

Scripts in the `work_path\config\broker_name\server_name\profiles` directory are run after the following scripts have run:

1. The `mqsiprofile` script.
2. Any scripts in the `work_path\Common\profiles` directory.
3. Any scripts in the `work_path\config\broker_name\profiles` directory.

To diagnose any problems, a log file is written that lists the profile scripts that have been read, and the environment that is used by this integration server. The log file location is `work_path\Common\log\broker_name.server_name.profilelog`.

Integration server profile scripts are not removed automatically when an integration server is deleted. If you no longer need the profile files that you have created, delete them manually.

Related tasks:

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

“Broker-specific command environment: Windows systems” on page 122

Extend or change the Windows environment that is used when you run a specific broker.

“Command environment: Windows systems” on page 120

Set up the Windows environment to run IBM Integration Bus commands.

Command environment: Linux and UNIX systems

Set up the Linux or UNIX environment to run IBM Integration Bus commands.

About this task

When you start a runtime component on Linux and UNIX systems, it inherits the environment from where you issue the `mqsistart` command.

You must therefore initialize the environment before you start a component; the command `mqsiprofile` located in the directory `install_dir/bin`, performs this initialization. If you are starting a broker, you might also need to initialize the environment for any databases that are accessed by the broker.

You must not change the location of the **mqsiprofile** command, or make user modifications to the command, because it might be replaced if you install service, or an update, to the product.

If you want to run your own additional environment settings, add a script (with the file extension `.sh`) to the directory `work_path/common/profiles`, where `work_path` identifies the machine-wide IBM Integration Bus working directory.

To verify the machine-wide IBM Integration Bus working directory, enter the following command in a command console:

```
echo %MQSI_WORKPATH%
```

When you run **mqsiprofile** again, the command automatically calls the additional user-written scripts in this location.

You must log out and log back in, to pick up the new files in the `work_path/common/profiles` directory, before you run **mqsiprofile** again.

The newly added script is not picked up if you run the **mqsistart** command from an existing initialized command shell.

A typical reason for adding scripts is to configure access to databases from deployed message flows. For more information, see “Running database setup scripts” on page 126.

When you have configured any additional setup, you can initialize the runtime environment for components and commands:

Procedure

1. Issue the **mqsiprofile** command:

```
. install_dir/bin/mqsiprofile
```

You must include the period and space preceding the location for this invocation to work correctly. Add this command to your login profile if you want it to be run at the start of every session.

This command accesses additional scripts that you have copied to the `common/profiles` directory, therefore the environment is initialized for runtime components and other resources such as databases.

If you use the `zsh` shell, issuing the **mqsiprofile** might cause the terminal session to exit. To resolve this issue, run the **unsetopt function_argzero** command before you run the **mqsiprofile** command.

2. Start the components that you want to run on this system with the **mqsistart** command.

What to do next

Check whether the following conditions apply to your environment:

- If you have a previous version of the product on the system, ensure that you run the correct profile before using IBM Integration Bus Version 9.0. The **mqsiprofile** command places the Version 9.0 commands and libraries at the front of your search path, and calls any user profiles that you have supplied that can override any combination of `PATH`, `CLASSPATH`, or library `PATH`.
- If you use the same user ID, and you run multiple profiles (from multiple different installations or versions), you might get unexpected results. Log off and log on again before you run the specific profile that you require.

- ODBC settings on Linux and UNIX systems are found in a text file defined by the ODBCINI environment variable. Set ODBCINI to point to a copy of the sample file `install_dir/ODBC/unixodbc/odbc.ini`.

You can check that your ODBC environment is configured correctly by running the **mqsicvp** command. This command also validates the connection to all data sources that are listed in the `odbc.ini` file that have been associated with a broker by using the **mqsisetdbparms** command. For more information, see **mqsicvp** command.

Running database setup scripts:

About this task

A broker might require access to databases from deployed message flows.

When you install a database product on Linux and UNIX systems, some database managers provide a profile to perform the environment setup that the database requires, or provide details of actions that you must take in their documentation. Always check the database product documentation for environment setup details; the information provided here is for general guidance only and might not be complete.

Procedure

1. If you can update the profile to provide permanent values for the details that are required (for example, the database server name or the installation directory):
 - a. Complete the changes to the profile.
 - b. Copy the profile file to the directory `work_path/Common/profiles`.
2. If you cannot update the profile permanently, but have to change it each time, you must run it independently of the **mqsiprofile** command:
 - a. Run the appropriate profiles to initialize the environment for the database:

- For DB2, issue the following command for the relevant DB2 instance:


```
. db2_instance_directory/sql/lib/db2profile
```
- For Oracle, export ORACLE_HOME and ORACLE_SID, then issue the following command:

```
. ${ORACLE_HOME}/bin/oraenv
```

- For Sybase, issue the following command, specifying your installation directory:

```
. Sybase_installation_directory/SYBASE.sh
```

- For Informix[®], check the documentation for the client on your broker system for details of the actions that you must take. For example, you might have to specify the following environment variables:

```
export INFORMIXDIR=/installation_directory_of_informix_client_software
export PATH=${INFORMIXDIR}/bin:${PATH}
export INFORMIXSERVER=server_name
export INFORMIXSQLHOSTS=${INFORMIXDIR}/etc/sqlhosts
export TERMCAP=${INFORMIXDIR}/etc/termcap
export TERM=vt100
export LIBPATH=${INFORMIXDIR}/lib:${INFORMIXDIR}/lib/esql:
    ${INFORMIXDIR}/lib/cli:$LIBPATH
```

where `server_name` is defined in the file `sqlhosts` (the required value is typically the machine name), and the location of the file `sqlhosts` is set up as part of the installation process.

To configure your system to run this setup at the start of every session, add these statements to the login profile of the user that is going to run the broker.

What to do next

When your environment has been set up, see [Working with databases](#) for information about setting up your databases for use with the broker.

Related tasks:

Working with databases

Create and configure databases to use with your message flows.

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

Changing the location of the work path

The work path directory is the location where a component stores internal data, such as installation logs, component details, and trace output. The shared-classes directory is also located in the work path directory and is used for deployed Java code. If the work path directory does not have enough capacity, redirect the directory to another file system that has enough capacity.

“Command environment: Windows systems” on page 120

Set up the Windows environment to run IBM Integration Bus commands.

Related reference:

“Additional software requirements” on page 633

IBM Integration Bus requires additional software products to run successfully.

Runtime commands

The topics in this section describe the IBM Integration Bus runtime commands.

Broker-specific command environment: Linux and UNIX systems

Extend or change the Linux or UNIX environment that is used when you run a specific IBM Integration Bus.

About this task

If you want to run your own additional environment settings for a broker, add one or more scripts (with the file extension `.sh`) to the `work_path/config/broker_name/profiles` directory, where `work_path` is the machine-wide IBM Integration Bus working directory, and `broker_name` is the name of your broker.

To verify the machine-wide IBM Integration Bus working directory, enter the following command in a command console:

```
echo %MQSI_WORKPATH%
```

Ensure that the broker name, and any integration servers that you create, contain only characters that are valid on your file system. You might also have to create the required directory structure.

All files that are placed into the `work_path/config/broker_name/profiles` directory that have a `.sh` extension, are processed when the broker starts.

Scripts in the *work_path/config/broker_name/profiles* directory are run after the **mqsiprofile** script, and any scripts in the *work_path/common/profiles* directory have run.

To diagnose any problems, a log file is written that lists the profile scripts that have been read, and the environment that is used by this broker. The log file location is *work_path/common/log/broker_name.profilelog*.

Integration server profile scripts are not removed automatically when an integration server is deleted. Therefore, if you have created any integration server profile files for this broker, you must delete these profile files manually if they are no longer required.

Broker profile scripts are not removed automatically when a broker is deleted. If you no longer need the profile files that you have created, delete them manually.

Related tasks:

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

“Command environment: Linux and UNIX systems” on page 124

Set up the Linux or UNIX environment to run IBM Integration Bus commands.

“Integration server-specific command environment: Linux and UNIX systems”

Extend or change the Linux or UNIX environment that is used when you run a specific integration server.

Integration server-specific command environment: Linux and UNIX systems

Extend or change the Linux or UNIX environment that is used when you run a specific integration server.

About this task

To enable function that becomes available in IBM Integration Bus fix packs, use the **-f** parameter on the **mqsichangebroker** command. For more information, see **mqsichangebroker** command.

If you want to run your own additional environment settings, add one or more scripts (with the file extension *.sh*) to the *work_path/config/broker_name/server_name/profiles* directory, where *work_path* is the machine-wide IBM Integration Bus working directory, *broker_name* is the name of your broker, and *server_name* is the name of your integration server.

To verify the machine-wide IBM Integration Bus working directory, enter the following command in a command console:

```
echo %MQSI_WORKPATH%
```

Ensure that the broker name, and any integration servers that you create, contain only characters that are valid on your file system. You might also have to create the required directory structure.

All files that are placed into the *work_path/config/broker_name/server_name/profiles* directory that have a *.sh* extension, are processed when the integration server (*server_name*) on the broker (*broker_name*) starts.

Scripts in the *work_path/config/broker_name/server_name/profiles* directory are run after the following scripts have run

1. The **mqsiprofile** script.
2. Any scripts in the *work_path/common/profiles* directory.
3. Any scripts in the *work_path/config/broker_name/profiles* directory.

To diagnose any problems, a log file is written that lists the profile scripts that have been read, and the environment that is used by this integration server. The log file location is *work_path/common/log/broker_name.server_name.profilelog*.

Integration server profile scripts are not removed automatically when an integration server is deleted. If you no longer need the profile files that you have created, deleted them manually.

Related tasks:

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

“Command environment: Linux and UNIX systems” on page 124

Set up the Linux or UNIX environment to run IBM Integration Bus commands.

“Broker-specific command environment: Linux and UNIX systems” on page 127

Extend or change the Linux or UNIX environment that is used when you run a specific IBM Integration Bus.

Applying service

Apply service to the IBM Integration Toolkit, the IBM Integration Explorer, and the Integration Bus component.

Service updates and other fixes are delivered occasionally in the form of Program Temporary Fixes (PTFs) or fix packs. You can download these fixes from the web and update your existing installations. Check regularly for updates and recommended fixes on the IBM Integration Bus support web page.

For information on obtaining the function level that your enterprise is using, see the **mqsichangebroker** command.

For information about installation directories for fix packs, and installing more than one fix pack at the same or different levels on a single computer, see “Coexistence and migration” on page 38.

- “Applying service to the Integration Bus component” on page 130
- “Applying service to the IBM Integration Toolkit” on page 141
- “Applying service to the IBM Integration Explorer” on page 146
- “Applying service to IBM Integration Administration for WebSphere Application Server” on page 147

Related tasks:

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

“Uninstalling service for the Integration Bus component” on page 158

On some platforms, you can remove fixes that you have applied.

“Uninstalling service from the IBM Integration Toolkit” on page 163

Remove maintenance or fixes from the IBM Integration Toolkit.

“Uninstalling service from the IBM Integration Explorer” on page 168
You must reinstall the IBM Integration Explorer to revert to a previous service level.

Related reference:



mqsichangebroker command

Use the **mqsichangebroker** command to change one or more of the configuration parameters of the broker.

Applying service to the Integration Bus component

Apply maintenance updates and program fixes to the Integration Bus component.

Before you begin

Before you apply service, check that you have backed up all associated resources. You might also choose to back up installation and work path directories. For details of this task, see Backing up resources.

About this task

You can download maintenance updates for all components of IBM Integration Bus from a website, in the form of a Program Temporary Fix (PTF), also known as a fix pack. Fix packs are cumulative; therefore, if multiple fix packs are available, you do not have to install a previous fix pack before you install the latest that is available. However, you must first purchase the GA (general availability) code to ensure that you comply with your license agreement. You can find the latest information about current fixes by following the link **Recommended fixes** in the Download section of the IBM Integration Bus support web page.

In some circumstances, fix packs are also provided as a media refresh; DVDs are re-created, and electronic images on IBM Passport Advantage are replaced. Check with your IBM representative if you want a media refresh.

Fix packs for IBM Integration Bus are installed by using the same technology as the GA release. Therefore, product files are consistently tracked and updated appropriately. For information about the default directories for a typical installation, see “Coexistence and migration” on page 38.

If you have applied one or more interim fixes to your existing installation, contact your IBM Service representative for instructions and possible updated interim fixes for the fix pack level that you are installing.

Because you can install more than one copy of the Integration Bus component on a single computer, you can choose how to apply service:

1. Apply service to an existing installation.

When you apply the fix pack, the new level of the product overwrites the existing level. The installer prohibits you from installing a previous modification or fix pack over a more recent one. You must stop all brokers before you apply service. All the brokers and resources that you have defined are retained. When the fix pack is successfully installed, restart the brokers.

Note: By default, the fix pack installer offers a new location for the installation with a directory name based on the fix pack level. To apply the fix pack to an existing installation, you must select the directory name of your existing installation when you are prompted for the target location.

2. Install the fix pack level of the product at a new location.

Fix packs are supplied as complete installations. You can install the product in a new location on your computer by using the fix pack packages. You can install only within the terms of your license:

- You must have purchased a GA level of the product on this computer.
- You must conform to the license restrictions for the number of installations on a single computer; your license might permit you to install more than one copy of the product, but allow you to run only one installation at a time.

This option provides several advantages:

- You do not have to stop brokers to complete the installation.
- You can choose which broker runs at which service level by running the appropriate **mqsiprofile** before you start the component.
- You can back out a broker to the previous service level by using the **mqsiprofile** from the older code level, and restarting the broker.

If you already have more than one installation on the computer, read the additional information in “Applying service on computers with multiple installations” on page 132.

3. Install the fix pack level of the product without having a previous installation
Fix packs are supplied as complete installations, so it is possible to install a fix pack on a computer that does not have a previous installation of the product. You can only within the terms of your license and specifically, you must have purchased a GA level of the product on the target computer.

Service is not affected by the operation mode in which your broker is working.

What to do next

If you want to remove service that you have applied, see “Uninstalling service for the Integration Bus component” on page 158.

Downloading a fix pack:

About this task

Download the latest fix pack from the IBM Integration Bus support web page:

Procedure

1. Click **Downloads** to open the page of available download packages.
2. Click **Recommended fixes**, then click the appropriate product and version. Available fixes are displayed.
3. Click the fix that you want to download. Details about the fix, and instructions for download and installation, are displayed.
4. Click **Problems Fixed** to find out what PMRs, APARs, and defects have been fixed in the fix pack.
5. Select your chosen operating system and click **FC** in the **Download package** section to start the download process.
6. If you are not already signed in, sign in with your IBM ID and password and click **Submit**. If you do not have an ID, you can register on this page.
7. Select the fixes that you want to download.
8. Select the download method to be used to download fixes.
9. Accept the download terms and conditions.
10. Click **Download now**.

11. When the download completes, follow the instructions to apply the maintenance updates for this fix.
 - a. If you are applying service to an existing installation, the installation wizard uses this current installation directory as the default directory for typical installations in graphical, console, and silent modes.

This location might therefore be different from the default directory that is set for an initial typical installation on a computer on which the product has not been installed. The default installation directories are described in “Coexistence and migration” on page 38, “Uninstalling the Integration Bus component by using the silent interface” on page 156, or “Uninstalling the Integration Bus component by using the graphical interface” on page 153. If you have previously installed components into non-default locations, check carefully that you are installing the fix pack into the required location.

 - Windows
 - Linux and UNIX
 - z/OS
 - b. If you are applying service by installing an additional copy of the product, follow the instructions that are provided in the “Installation Guide” on page 32, specifying a new installation location.

Applying service on computers with multiple installations:

About this task

If you have installed the Integration Bus component more than once on a single computer, you can choose to apply service to one or more of these installations in any order. You might therefore choose to install a service level on one instance initially, and complete some tests, before you apply the fixes to additional installations.

You must ensure that you comply with the terms of your license at all circumstances.

Distributed systems

You specify which installation that you want a broker to work with by running the **mqsiprofile** command that is associated with that installation. Any resources that you have defined are associated with the installation for which you have run **mqsiprofile**. When the **mqsiprofile** command has completed, restart the broker.

If you decide to return to a previous level, stop the broker, run the **mqsiprofile** command that is associated with the installation at the previous service level, and restart the broker.

For more information about working with multiple installations, see “Command environment: Windows systems” on page 120 and “Command environment: Linux and UNIX systems” on page 124.

z/OS systems

You can create component profiles to work with one of multiple installations. Details are provided in “Installing service on z/OS” on page 139.

For more information about components, see “Overview of broker creation on z/OS” on page 314.

If you have a single installation on a computer, and you want to apply service, you must stop all brokers, apply service, and restart the brokers. The service is applied to the product code, and you can start the brokers immediately to run against the new service level.

Related concepts:

“Operation modes” on page 5

The operation mode that you can use for your broker is determined by the license that you purchase.

Related tasks:

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

“Uninstalling the Integration Bus component” on page 151

You can uninstall the Integration Bus component on distributed systems in a number of ways.

“Uninstalling service for the Integration Bus component” on page 158

On some platforms, you can remove fixes that you have applied.

Related reference:

“Installation Guide” on page 32

Installation information for IBM Integration Bus is provided in the online information center.

“Installation and uninstallation interfaces” on page 652

You can use different interfaces for installation and uninstallation of the Integration Bus component and the IBM Integration Toolkit.

“Installation and uninstallation authorization” on page 661

Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

Installing service on Windows:

Apply maintenance updates and program fixes to the Integration Bus component.

Before you begin

Before you start:

- Download the required update package for the IBM Integration Bus component, as described in “Applying service to the Integration Bus component” on page 130.
- Check that your user ID has the correct authority to complete this task; see “Installation and uninstallation authorization” on page 661 for details.

About this task

Fix packs are supplied as self-extracting executable programs. Complete the following steps to install the updates that you have downloaded:

Note: A fix pack can be installed on a system that does not have a previous version of IIB. In this situation the license files that are provided with GA installation media must be made available. These are available on the DVD media or the download media from Passport Advantage in a directory named *license*. When performing a graphical or console installation, the installer will prompt the user to supply the location of the *license* directory. When performing a silent installation, the location of the *license* directory should be supplied using the flag

-DExternal_License_Path=<path to licenses>, or using the property External_License_Path=<path to licenses> in the response file.

Procedure

1. Check that your user ID has the correct authority to install service for the Integration Bus component. The requirements are defined in “Installation and uninstallation authorization” on page 661.
2. Ensure that all IBM Integration Bus function is stopped:
 - a. Stop all brokers on this computer by using the **mqsistop** command, or by stopping the Windows services for these components from **Start > Settings > Control Panel > Administrative Tools > Services**.
 - b. Close all instances of the IBM Integration Explorer.
 - c. Close all instances of the IBM Integration Toolkit.
 - d. Check that all files associated with IBM Integration Bus are closed.
3. Close all Windows programs before applying maintenance to the IBM Integration Bus component to ensure that data is not lost.
4. Run the fix pack self-extracting program from either Windows Explorer or the command line.

The program is referred to as *FixPackLauncher* and its name is in the form *V.R.M-prod-platform-fixpack.exe*. Where *V.R.M-prod* represents the version, release, and modification level of the product, *-platform* identifies the platform, and *-fixpack* identifies the fix pack. For example, the file name *9.0.0-WS-MB-WINIA32-FP0001.exe* identifies the file for Fix Pack 1 on Version 9.0.0 of IBM Integration Bus on Windows.

- If you start this program from Windows Explorer, or you start it from the command line with no options, the *FixPackLauncher* runs with default options. It extracts the updates from the fix pack file, and starts the graphical interface of the IBM Integration Bus installation wizard.
- If you run the wizard from the command line with options, you can choose the interface that you want to use, and how the IBM Integration Bus installation wizard runs.

For further details about how you can install, and other options that you can specify, see “Installation and uninstallation interfaces” on page 652 and “How to install and uninstall the Integration Bus component” on page 653.

If you start this wizard from a directory other than the one in which the file is saved, you must include the fully qualified path of the file and its name.

Graphical interface

Use the following command format:

```
FixPackLauncher -i swing
```

gui specifies that you want to use the graphical interface of the installation program. The graphical interface is used by default if you specify only *FixPackLauncher*.

For example, you can use either of the following commands:

```
9.0.0-WS-MB-WINIA32-FP0001.exe
```

```
9.0.0-WS-MB-WINIA32-FP0001.exe -i swing
```

Console interface

Use the following command format:

```
FixPackLauncher -i console
```

console specifies that you want to use the console interface of the installation program. You must specify this option if you want the installation to use this interface.

For example:

```
9.0.0-WS-MB-WINIA32-FP0001.exe -i console
```

Silent interface

Use the following command format. In this example, LICENSE_ACCEPTED=TRUE is specified, which means that the license has been accepted. By using this command you are accepting the terms of the license agreement:

```
FixPackLauncher -i silent -DLICENSE_ACCEPTED=TRUE -DUSER_INSTALL_DIR=installation_directory
```

silent specifies that you want to use the silent interface of the installation program. You must specify this option if you want the installation to use this interface. Where *installation_directory* is your chosen installation directory or the default installation directory for your platform. For more information about the default installation directories for distributed operating systems, see “Coexistence and migration” on page 38.

If you do not specify a response file, the default options are used from the supplied response file. If you want to tailor the options to suit your installation requirements, create your own response file and specify its location on the command.

For example, to run both the extractor program and the IBM Integration Bus installation program silently with a response file, enter the following command:

```
FixPackLauncher -i silent -f response_file_name
```

5. If you use the graphical or console interface of the IBM Integration Bus installation program, follow the prompts given and provide any input that is required to complete the installation.
6. If you run the fix pack installation program or the IBM Integration Bus installation program silently, check the installation log to ensure that the process was successful. File IBM_Integration_Bus_InstallLog.log is stored in your installation directory.

If you accepted the default location during installation, this directory is as follows. The default directory includes the version, release, modification, and fix of the product that you are installing in the format v.r.m.f (version.release.modification.fix):

Linux /opt/ibm/mqsi/v.r.m.f

UNIX /opt/IBM/mqsi/v.r.m.f

Windows 32-bit

C:\Program Files\IBM\MQSI\v.r.m.f

Windows 64-bit

C:\Program Files\IBM\MQSI\v.r.m.f for the 64-bit version of IBM Integration Bus

C:\Program Files (x86)\IBM\MQSI\v.r.m.f for the 32-bit version of IBM Integration Bus

These locations define the default value of *install_dir* on each platform.

7. When installation has successfully completed, review the release notes that are supplied in the directory readmes, and complete any manual post-installation tasks that are required.
8. When you have completed any required ODBC changes, restart the components by using the **mqsistart** command.

Related tasks:

“Applying service to the IBM Integration Toolkit” on page 141

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

“Uninstalling service for the Integration Bus component” on page 158

On some platforms, you can remove fixes that you have applied.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“How to install and uninstall the Integration Bus component” on page 653

You can install and uninstall the Integration Bus component by using one of three interfaces.

“Installation and uninstallation interfaces” on page 652

You can use different interfaces for installation and uninstallation of the Integration Bus component and the IBM Integration Toolkit.

“Installation and uninstallation authorization” on page 661

Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

“Using response files with the Integration Bus component” on page 656

You can use a response file to define the behavior of an installation wizard that is running the silent interface.



mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.



mqsistop command

Use the **mqsistop** command to stop the specified component.

Installing service on Linux and UNIX:

Apply maintenance updates and program fixes to the Integration Bus component.

Before you begin

Before you start:

- Download the required update package for the Integration Bus component, as described in “Applying service to the Integration Bus component” on page 130.
- Check that your user ID has the correct authority to complete this task; see “Installation and uninstallation authorization” on page 661 for details.

About this task

To install the updates that you have downloaded:

Note: A fix pack can be installed on a system that does not have a previous version of IIB. In this situation the license files that are provided with GA installation media must be made available. These are available on the DVD media or the download media from Passport Advantage in a directory named *license*.

When performing a graphical or console installation, the installer will prompt the user to supply the location of the *license* directory. When performing a silent installation, the location of the *license* directory should be supplied using the flag `-DExternal_License_Path=<path to licenses>`, or using the property `External_License_Path=<path to licenses>` in the response file.

Procedure

1. Check that your user ID has the correct authority to uninstall the Integration Bus component. The requirements are defined in “Installation and uninstallation authorization” on page 661.
2. Log in to the system.
 - On AIX, you must either log in as root, or use the same user ID that you initially used to complete the original installation as a user without root authorities.
 - On Linux and on other UNIX computers, your user ID must either have root authority to complete the installation, or you must use the same user ID that you initially used to complete the original installation as a user without root authorities.
3. Stop all brokers that are running on this computer by using the `mqsistop` command. If you are applying service on a computer that is running Linux on x86, close all instances of the IBM Integration Toolkit and the IBM Integration Explorer.

Check that all files associated with IBM Integration Bus are closed.

4. Change to the directory where you downloaded the fix pack file. The file name is in the form `V.R.M-prod-platform-fixpack.tar.z`, where *V.R.M-prod* represents the version, release, and modification level of the product, *-platform* identifies the platform, and *-fixpack* identifies the fix pack. For example, the file name `9.0.0-WS-MB-AIXPPC64-FP0001.tar.z` identifies the file for Fix Pack 1 on Version 9.0.0 of IBM Integration Bus on AIX.

Note: When using AIX, if you selected HTTP for the download instead of the Download Director, the file name extension is `.tar.tgz` instead of `.tar.z`.

5. Replace the `.z` file with a `.tar` file, by entering the following command:
uncompress -fv V.R.M-prod-platform-fixpack.tar.z
6. Untar the image and extract all the directories, subdirectories, and files required to apply the update, by entering the following command:
tar -xvf V.R.M-prod-platform-fixpack.tar
7. Decide which interface you want to use for the installation; `-i swing` (the default option if none is specified), `-i console`, or `-i silent`. For more information about the interfaces that you can use to run the installation, and other options that you can specify, see “Installation and uninstallation interfaces” on page 652 and “How to install and uninstall the Integration Bus component” on page 653.
8. Depending on the choice you have selected, run one of the following commands. Where *installer* is the appropriate installation program for your platform, as listed in the following table.
 - To use the installation wizard graphical interface, type the *installer* name with no options at a command prompt, and press Enter. Include the absolute or relative path names if you start installation from a directory other than the one in which the wizard exists. When you start the

installation wizard, it guides you through a series of panels where you can make choices about where to install the components, and which components you want to install.

- To use the installation wizard console interface, enter *installer -i console*. Include the absolute or relative path names if you start installation from a directory other than the one in which the wizard exists. When you start the installation wizard, it prompts you to enter input for all options, including where to install the components, and which components you want to install.
- To use the installation wizard silent interface, enter *installer -i silent*. Include the absolute or relative path names if you start installation from a directory other than the one in which the wizard exists. If you want to specify non-default settings, include a response file on the command. For more details of how to specify a response file, and how to create and edit a response file to define your requirements, see “Using response files with the Integration Bus component” on page 656.

Platform	Installation program
AIX	setupaix
HP-Itanium	setuphpia64
Linux on POWER®	setuplinuxppc
Linux on x86	setuplinuxia32
Linux on x86-64	setuplinuxx64
Linux on System z®	setuplinux390
Solaris on SPARC	setupsolaris
Solaris on x86-64	setupsolarisx64

The installer extracts the update files and runs according to the arguments supplied.

9. Follow the instructions presented by the installation program and provide all input that is required to complete the installation.
When installation has successfully completed, review the release notes that are supplied in the directory `readmes`, and complete all manual post-installation tasks that are required.
10. If you have defined ODBC connections for brokers on this computer, you must update the `odbc.ini` files before you restart your brokers.
11. When you have completed any required ODBC changes, restart the brokers by using the **mqsistart** command.

Related tasks:

- “Applying service to the IBM Integration Toolkit” on page 141
Apply maintenance updates and program fixes to the IBM Integration Toolkit.
- “Uninstalling service for the Integration Bus component” on page 158
On some platforms, you can remove fixes that you have applied.
- “Finding the latest information” on page 1
Access the latest information for IBM Integration Bus.

Related reference:

- “Installation and uninstallation interfaces” on page 652
You can use different interfaces for installation and uninstallation of the Integration Bus component and the IBM Integration Toolkit.
- “Installation and uninstallation authorization” on page 661
Check the authorization requirements for the user ID that you use to install or

uninstall the Integration Bus component and the IBM Integration Toolkit.
“Using response files with the Integration Bus component” on page 656
You can use a response file to define the behavior of an installation wizard that is running the silent interface.

mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.

mqsistop command

Use the **mqsistop** command to stop the specified component.

Installing service on z/OS:

Apply maintenance updates to the broker.

Before you begin

Before you start:

- Download the required update package for IBM Integration Bus, as described in “Applying service to the Integration Bus component” on page 130.
- Check that your user ID has the correct authority to complete this task; see “Installation and uninstallation authorization” on page 661 for details.

About this task

Use this three stage process to install service on z/OS:

1. Store the updates on your target system.
2. Install and test the fix pack. If testing is not satisfactory, you can remove this fix pack at this stage.
3. After testing, confirm the fix pack as your current base level of installation.

Follow these instructions to complete the process:

Procedure

1. Receive the updates. Use SMP/E RECEIVE to transfer the contents of the fix pack from the tape or the downloaded package to your system at the specified location.
2. Apply the updates. Use SMP/E APPLY to install the fix pack which completes the link edit steps and generates a runtime environment in your chosen location.

The installation process completes updates to USS and to four product data sets (SBIPSAMP, SBIPINST, SBIPPROC, and SBIPAUTH).

If your SMP/E target system is not on the same file system as your production system, copy the data sets (including the data set that represents your USS mount point) to the system where they are to be used, overwriting the current content. When you complete this action, all runtime components use the updated data sets when they next restart.

If you do not want all your brokers to use the updated installation, copy the data sets to a different location and start selected brokers against that copy. Further details are provided in “Multiple concurrent installations at different levels” on page 140.

You can specify the same destination for receiving the fix pack contents (the SMP/E target system) and for installing (the production system). If you specify a different location, you must copy the product data sets to the appropriate location.

When you have installed the fix pack, test it to ensure that it is working correctly in your environment. If you find problems at this stage, you can remove the fix pack by using SMP/E RESTORE. The command restores your installation to the level that you last accepted (SMP/E ACCEPT).

3. Confirm the update. Use SMP/E ACCEPT to commit the changes and to establish this fix pack as the current level. When you have completed this step, you cannot restore your system to a previous service level. If you want to use an earlier service level, you must reinstall that level.

Multiple concurrent installations at different levels:

About this task

If you want to install a later fix pack level, and retain your existing installation so that you can use both service levels, install the second level in a different location. You can then set up additional customization so that one or more of your existing brokers can use the second service level. By retaining the product files in the first location, you can switch a broker back to the first level, if required.

The following steps are a summary of the tasks that you must complete to run a broker at one service level concurrently with a broker at another service level. These steps are identical to those you take for setting up an initial installation, except where indicated.

For a full description of installation and customization for brokers, see “Creating a broker on z/OS” on page 313.

Procedure

1. To install a second service or fix pack level:
 - a. Install the second level of the broker by using the instructions provided with that level.
 - b. Copy all the template data sets and the main files from the HFS file system to a unique location.
2. To prepare a broker for use with the second installation:
 - a. Create and customize a new broker component data set. Complete the steps described in “Customizing the broker component data set” on page 323.
 - b. Copy the template JCL from the second location to the new dataset.
 - c. Customize the JCL to work with the second location. For more details about this step, see “Customizing the broker JCL” on page 324.
3. To change a broker to work with the second installation:
 - a. Submit the BIPGEN job to update the environment file ENVFILE for the broker. This task is described in “Creating the environment file” on page 325.
 - b. Copy the customized started task from the new data set into the procedures library, for example USER.PROCLIB.
 - c. Start the broker; the broker is now associated with the second location.
4. To roll back to the previous version:
 - a. Open the original broker component data set, that is associated with the first location.

- b. Resubmit the BIPGEN job from that data set.
- c. Copy the original PROC into the procedures library USER.PROCLIB.
- d. Stop and restart the broker.

Related tasks:

“Installing service on Windows” on page 133

Apply maintenance updates and program fixes to the Integration Bus component.

“Installing service on Linux and UNIX” on page 136

Apply maintenance updates and program fixes to the Integration Bus component.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

“Applying service to the IBM Integration Toolkit”

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

Related reference:



Customization tasks and roles on z/OS

System programmers do most of the customization of IBM Integration Bus.

“Installation and uninstallation authorization” on page 661

Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

Applying service to the IBM Integration Toolkit

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

Before you begin

Before you apply a fix pack or other service pack to the IBM Integration Toolkit, you might want to preserve the resources that you have created or imported into Version 9.0. For details of this task, see [Backing up the IBM Integration Explorer and IBM Integration Toolkit workspace](#).

Check that your user ID has the correct authority to complete this task; see [“Installation and uninstallation authorization” on page 661](#) for details.

About this task

The IBM Integration Toolkit uses IBM Installation Manager to install service updates. Occasionally, an update is also required to Installation Manager; if an update is required, it is downloaded automatically and you are prompted to apply service to Installation Manager before you can apply service to the IBM Integration Toolkit.

Occasionally, updates to the information center are also made available. These updates are integrated with component updates when they are made available, and are therefore automatically included when you apply any updates using these instructions.

You can install multiple versions of the IBM Integration Toolkit at the same modification or fix pack level, or at different levels, on a single computer. Each installation must be in a separate package group; package groups are described in more detail in [“IBM Installation Manager” on page 636](#).

For information about current fixes, follow the link **Recommended fixes** in the [Download](#) section of the IBM Integration Bus support web page. Click the link for Version 9.0, and select the updates you want to apply. Service release notes, the list

of problems fixed in a fix pack or other maintenance release, and the instructions you must follow to complete the service upgrade are available. The instructions provide full details of how to apply each service fix, and include information about fixes to Installation Manager, if required. Always use those instructions in preference to the information in this topic, because they are maintained regularly and provide the latest information.

If you want Installation Manager to retrieve updates from a different location, see “Changing the update repositories for Installation Manager” on page 143. If you use FTP or HTTP proxies to allow applications to access the Internet, see “Setting FTP and HTTP proxies for Installation Manager” on page 145.

Use Installation Manager for all service updates, including those updates to Installation Manager itself. In most cases, the following instructions are sufficient; occasionally, further steps are necessary, therefore you must always check with the instructions provided with each fix on the support website.

Procedure

1. Start Installation Manager by using one of the following two methods:
 - From the Linux main menu or the Windows Start menu:
 - **Linux** On Linux: You can use this option only if you are logged in as root. If you have logged in as another user, you cannot use the main menu links, even if you become root.
 - On Red Hat: Click **System** > **IBM Installation Manager**.
 - On SUSE: Click **System Tools** > **IBM Installation Manager**.
 - **Windows** On Windows: Click **Start** > **Programs** > **IBM Installation Manager** > **IBM Installation Manager**.
 - From the command line:

Navigate to the `eclipse` subdirectory in the directory in which Installation Manager is installed and run the following command:

 - **Linux** `IBMIM`
 - **Windows** `IBMIM.exe`

Installation Manager opens and displays its initial panel.

2. Click **Update Packages**, then select the package group in which you have installed the IBM Integration Toolkit.
3. Click **Next**. Installation Manager searches the selected update site or sites, and displays a list of available updates for the products and features that you have installed.

Initially **Show recommended** (below the list) is selected, and the latest updates only are shown. To include earlier updates as well, select **Show all**.

For information about how you can change the locations that are searched, see “Changing the update repositories for Installation Manager” on page 143.

4. Select the updates that you want to apply and click **Update**. The updates are applied.
5. To return to the Installation Manager initial panel, click **OK**.
6. To close Installation Manager, click **File** > **Exit**. Verify that the changes have worked; see “Verifying your installation by using the IBM Integration Toolkit” on page 105.

Results

If you want to remove service that you have applied, see “Uninstalling service from the IBM Integration Toolkit” on page 163.

Related tasks:

“Changing the update repositories for Installation Manager”

If you have a large number of IBM Integration Toolkit users, or your users do not have regular access to the Internet, you might find it more convenient to download updates to a local site or server, and modify Installation Manager to access the updates at that alternative site.

“Setting FTP and HTTP proxies for Installation Manager” on page 145

If your environment has a firewall, or uses proxies to control access to the Internet, you can change the Installation Manager settings to enhance your access security.

“Applying service to the Integration Bus component” on page 130

Apply maintenance updates and program fixes to the Integration Bus component.



Backing up the IBM Integration Explorer and IBM Integration Toolkit workspace

The IBM Integration Explorer and IBM Integration Toolkit workspaces contain your personal settings and data, such as message flow and message set resources. You can have multiple workspaces in different locations, and you can also have references to projects that are in other locations, therefore consider all these locations when you back up your resources.

“Uninstalling service from the IBM Integration Toolkit” on page 163

Remove maintenance or fixes from the IBM Integration Toolkit.

“Uninstalling the IBM Integration Toolkit” on page 159

Choose the method that you want to use to uninstall the IBM Integration Toolkit and follow the instructions to remove the component.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

“Verifying your installation by using the IBM Integration Toolkit” on page 105

Use the instructions in this tutorial to verify your installation of IBM Integration Bus and learn how to run samples with the IBM Integration Toolkit.

Related reference:

“Installation and uninstallation interfaces” on page 652

You can use different interfaces for installation and uninstallation of the Integration Bus component and the IBM Integration Toolkit.

“Installation and uninstallation authorization” on page 661

Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

Changing the update repositories for Installation Manager:

If you have a large number of IBM Integration Toolkit users, or your users do not have regular access to the Internet, you might find it more convenient to download updates to a local site or server, and modify Installation Manager to access the updates at that alternative site.

About this task

For example, if you want to limit Internet access to a restricted set of computers, or if you want to reduce download times, or both, you can set up local repositories that your users can access to apply updates.

To set up alternative repositories:

Procedure

1. Start Installation Manager from the Start menu (on Windows) or main menu (on Linux on x86), or from the command line. Installation Manager opens and displays its initial panel.
2. Click **File > Preferences**. The Preferences dialog is displayed.
3. Click **Repositories** in the left pane. The Repositories page is displayed.
4. Click **Add Repository**. If you know the name and location of the update repositories that Installation Manager is to search, enter the URL or fully qualified location of the update file, including the repository name; for example, `repository.config`.

To search for repositories, which can be on a local or a remote computer, click **Browse** and navigate to the correct location. The field **File of type** lists the values `repository.config`, `diskTag.inf`, `.jar`, and `.zip`, and repositories with these names and types are searched for by default. The identity of an update depends on what type of update it is (for example, disk images are named `diskTag.inf`),

If you are not sure what updates you might want, leave the contents of field **File of type** unchanged; all available updates are shown. Click the required repository to select it, then click **Open**. The repository is added to the displayed list of repositories. If the repository is not connected, a red cross is displayed in the **Connection** column. You might need to check your network connections, or try again at a later time.

When it is added to the list, the repository is initially shown as selected, to indicate that its contents will be included when you run Installation Manager to update packages.

Repeat this step to add additional repositories if appropriate from the same or another location. You can have one or more locations selected at any time; Installation Manager searches all selected repositories listed on the Repositories page.

5. If you want Installation Manager to search only the repositories that you have added, clear **Search service repositories during installation and update**. The service repositories are those locations on the product support Web sites that host updates. If you clear this option, the default service repositories are not searched. The service repositories location is fixed and predefined; you cannot change it.
6. Click **OK** on the Browse panel if you opened it, and click **OK** on the Repositories page.
7. Click **Apply** to confirm the update, then click **OK** to close the Preferences dialog box.

What to do next

To remove a repository from the selected list, clear its selection. If you want to delete a repository in the list, select the repository and click **Remove Repository**. To clear selection of all listed repositories, click **Clear Credentials**.

Related tasks:

“Applying service to the IBM Integration Toolkit” on page 141

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

“Applying service to the Integration Bus component” on page 130

Apply maintenance updates and program fixes to the Integration Bus component.



Backing up the IBM Integration Explorer and IBM Integration Toolkit workspace

The IBM Integration Explorer and IBM Integration Toolkit workspaces contain your personal settings and data, such as message flow and message set resources. You can have multiple workspaces in different locations, and you can also have references to projects that are in other locations, therefore consider all these locations when you back up your resources.

“Uninstalling service from the IBM Integration Toolkit” on page 163
Remove maintenance or fixes from the IBM Integration Toolkit.

“Uninstalling the IBM Integration Toolkit” on page 159
Choose the method that you want to use to uninstall the IBM Integration Toolkit and follow the instructions to remove the component.

Related reference:

“Installation and uninstallation interfaces” on page 652
You can use different interfaces for installation and uninstallation of the Integration Bus component and the IBM Integration Toolkit.

Setting FTP and HTTP proxies for Installation Manager:

If your environment has a firewall, or uses proxies to control access to the Internet, you can change the Installation Manager settings to enhance your access security.

About this task

You do not have to complete this task if you have changed the Installation Manager repository location to a local site.

Complete the following steps:

Procedure

1. Start Installation Manager from the Start menu (on Windows) or main menu (on Linux on x86), or from the command line. Installation Manager opens and displays its initial panel.
2. Click **File > Preferences**. The Preferences dialog is displayed.
3. Expand the **Internet** tree in the left pane, and select **FTP Proxy** or **HTTP Proxy**.
4. Select **Enable proxy server** and enter the details in the entry fields that are required for the protocol that you selected.
5. Click **Apply** to confirm the updates, then click **OK** to close the Preferences dialog box.

Related tasks:

“Applying service to the IBM Integration Toolkit” on page 141
Apply maintenance updates and program fixes to the IBM Integration Toolkit.
“Applying service to the Integration Bus component” on page 130
Apply maintenance updates and program fixes to the Integration Bus component.



Backing up the IBM Integration Explorer and IBM Integration Toolkit workspace

The IBM Integration Explorer and IBM Integration Toolkit workspaces contain your personal settings and data, such as message flow and message set resources. You can have multiple workspaces in different locations, and you can also have references to projects that are in other locations, therefore consider all these locations when you back up your resources.

“Uninstalling service from the IBM Integration Toolkit” on page 163
Remove maintenance or fixes from the IBM Integration Toolkit.

“Uninstalling the IBM Integration Toolkit” on page 159

Choose the method that you want to use to uninstall the IBM Integration Toolkit and follow the instructions to remove the component.

Related reference:

“Installation and uninstallation interfaces” on page 652

You can use different interfaces for installation and uninstallation of the Integration Bus component and the IBM Integration Toolkit.

Applying service to the IBM Integration Explorer

You can apply maintenance or fixes to the IBM Integration Explorer.

About this task

Download the latest fix pack from the IBM Integration Bus support web page:

Procedure

1. Click **Recommended fixes** in the Download section of the support page.
2. Click your IBM Integration Bus product. Available fixes are displayed.
3. Click the fix that you want to download. Details about the fix, and instructions for download and installation, are displayed.
4. Click **Problems fixed** to find out what PMRs, APARs, and defects have been fixed in the fix pack.
5. Click **Release notes** to read more detail about the fix pack contents.
6. Click **FC** in the **Download package** section to start the download process. Accept the download terms and conditions and sign in to the download site with your IBM ID. If you do not have an ID, you can register on this page. When the download completes, extract the executable file or binary file.
7. Decide whether to use the graphical installation, a console installation, or a silent installation.
 - To use the graphical installation, see “Installing IBM Integration Explorer” on page 84.
 - To use the installation wizard with a screen reader, see “Installing the IBM Integration Explorer by using console mode with a screen reader” on page 87.
 - To use a console installation on Windows, see “Installing IBM Integration Explorer in console mode on Windows” on page 88.
 - To use a console installation on Linux, see “Installing IBM Integration Explorer in console mode on Linux” on page 89.
 - To use a silent installation, see “Installing IBM Integration Explorer in silent mode” on page 91.

To apply service to any IBM Integration Bus component on Windows using the *fixpacklauncher* see “Installing service on Windows” on page 133.

8. Launch the installation wizard by running the executable file or binary file. The name of this file is in the form *V.R.M-prod-platform-fixpack.exe* on Windows and *V.R.M-prod-platform-fixpack.bin* on Linux platforms. Where *V.R.M* represents the version, release, and modification level of the product, *prod* identifies the product, *platform* identifies the platform, and *fixpack* identifies the fix pack. For example, the file name *9.0.0-WS-MBX-WINIA32-FP0001.exe* identifies the file for Fix Pack 1 on Version 9.0.0 of IBM Integration Explorer on Windows.

On Linux, you must make this file executable by using the **chmod** command. For example, use the following command: `chmod u+x 9.0.0-WS-MBX-LINUXIA32-FP0001.bin`

9. When you have launched the wizard, continue with the installation process by working through each of the panels. The installation wizard itself contains help information about selected panels. You can apply service to a new location on your system.

Related tasks:

“Applying service to the IBM Integration Toolkit” on page 141

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

“Applying service to the Integration Bus component” on page 130

Apply maintenance updates and program fixes to the Integration Bus component.

“Uninstalling service from the IBM Integration Explorer” on page 168

You must reinstall the IBM Integration Explorer to revert to a previous service level.

“Installing IBM Integration Explorer” on page 84

To use IBM Integration Explorer only, without installing the complete IBM Integration Toolkit, use the IBM Integration Explorer installation wizard to install the IBM Integration Explorer.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Applying service to IBM Integration Administration for WebSphere Application Server

Apply maintenance updates and program fixes to the IBM Integration Administration for WebSphere Application Server feature.

About this task

Updates for the feature repository are included in IBM Integration Bus fix packs. You can also apply feature updates by using the live update repository for IBM Installation Manager.

Click the following links to find out how to apply service to your system:

- “Applying service to the IBM Integration administration feature: Windows, Linux, and UNIX” on page 148
- “Removing service from the IBM Integration feature: Windows, Linux, and UNIX” on page 149
- “Applying service to the IBM Integration Administration feature: z/OS” on page 150

Related tasks:

“Installing IBM Integration Administration for WebSphere Application Server” on page 93

If you want to view or manage your IBM Integration resources from the WebSphere Application Server administrative console, use IBM Installation Manager on Windows and UNIX, or a provided script on z/OS, to install the IBM Integration feature into WebSphere Application Server. Use the **manageprofiles** command to configure the WebSphere Application Server administrative console to make available the IBM Integration Administration feature.

Applying service to the IBM Integration administration feature: Windows, Linux, and UNIX:

Learn about how to use IBM Installation Manager to apply service to IBM Integration Administration for WebSphere Application Server.

Before you begin

If you are installing the administration feature for the first time, you must retrieve the base feature repository file and add it to the IBM Installation Manager repository list. See “Installing the IBM Integration feature: Windows, Linux, and UNIX” on page 94.

About this task

You can apply service at the same time as you install the feature, or on a later occasion. You can apply local updates by using either the live update facility or a repository update file on your local system.

Procedure

To update the administration feature, complete the following steps:

1. Decide whether you want to retrieve the repository update file and perform a local update or perform a live update from a remote location. Add one or both packages to the IBM Installation Manager repositories list.
 - Optional: If you want to perform a live update, add the original installation repository to the IBM Installation Manager repository list and select the **Search service repositories during installation and updates** check box.
 - If you want to install the latest updates, but you do not want IBM Installation Manager to retrieve them automatically from a remote location, then clear the **Search service repositories during installation and updates** check box and locate or retrieve the update file supplied with your installation.

The name of the repository update file is `wmb.admin.repository_updates.zip`. You can find the file at the following location:

- On Windows: `install_dir\tools\wmb.admin.repository_updates.zip`
- On Linux, UNIX, and z/OS: `install_dir/tools/wmb.admin.repository_updates.zip`

Ensure that the new repository is selected in the repository list.

2. On the main IBM Installation Manager panel, click **Update**.
3. On the **Update Packages** panel, select the package group that you want to update; **IBM WebSphere Application Server VX.x**, where X.x is the product version. Click **Next**.
4. Select the feature versions that you want to install and click **Next**.
5. On the Licenses panel, review and accept the product license. Click **I accept the terms in the license agreement**. Click **Next**.
6. If you see a message about running processes that might interfere with the current operation, you must stop the processes before the installation can continue.
7. On the Summary panel, review the installation summary and click **Update** to start the installation.

8. Review the installation log and make note of the details of the **manageprofiles** command that is displayed in this panel.

Note: After you click **Finish**, use the **manageprofiles** command to augment your WebSphere Application Server profile; see “Configuring IBM Integration Administration for WebSphere Application Server” on page 102.

9. Click **Finish**.

Related tasks:

“Installing the IBM Integration feature: Windows, Linux, and UNIX” on page 94
Learn about how to use IBM Installation Manager to install IBM Integration Administration for WebSphere Application Server.

“Configuring IBM Integration Administration for WebSphere Application Server” on page 102

Learn about how to use the **manageprofiles** command to create or augment profiles in WebSphere Application Server, and about how to enable the IBM Integration web user interface, so that you can use IBM Integration administration for WebSphere Application Server.

Removing service from the IBM Integration feature: Windows, Linux, and UNIX:

Learn about how to use IBM Installation Manager to remove service from IBM Integration Administration for WebSphere Application Server.

About this task

You can roll back the installed fix pack to the previous version by completing the following steps:

Procedure

1. Open IBM Installation Manager and click **Roll Back** in the main window.
2. On the **Roll Back Packages** panel, select the package group that you want to modify. For example, **IBM WebSphere Application Server VX.x**, where X.x is the product version. Click **Next**.
3. Select the feature version that you want to roll back. Click **Next**.
4. If you see a message about running processes that might interfere with the current operation, you must stop the processes before the installation can continue.
5. If necessary, unaugment your WebSphere Application Server profiles. You are prompted to use the **manageprofiles** command for this purpose; see “Configuring IBM Integration Administration for WebSphere Application Server” on page 102.
6. On the Summary panel, review the summary and click **Roll Back** to start the rollback.
7. When the rollback is complete, review the log, and click **Finish**.
8. Augment the profiles that you unaugmented for the rollback; see “Configuring IBM Integration Administration for WebSphere Application Server” on page 102.

Related tasks:

“Installing the IBM Integration feature: Windows, Linux, and UNIX” on page 94
Learn about how to use IBM Installation Manager to install IBM Integration Administration for WebSphere Application Server.

“Configuring IBM Integration Administration for WebSphere Application Server” on page 102

Learn about how to use the **manageprofiles** command to create or augment profiles in WebSphere Application Server, and about how to enable the IBM Integration web user interface, so that you can use IBM Integration administration for WebSphere Application Server.

“Applying service to the IBM Integration administration feature: Windows, Linux, and UNIX” on page 148

Learn about how to use IBM Installation Manager to apply service to IBM Integration Administration for WebSphere Application Server.

Applying service to the IBM Integration Administration feature: z/OS:

Apply maintenance updates and program fixes to the IBM Integration Administration for WebSphere Application Server feature, or restore an earlier version of the feature.

Before you begin

Before you apply service, back up all associated resources.

About this task

You cannot apply or remove individual fixes for the administration feature on z/OS. If you want to update a system or restore it to a previous level, you must uninstall the entire feature and then install the required level. Complete the following steps:

Procedure

1. Optional: If you previously installed the administration feature onto this WebSphere Application Server system, unaugment your WebSphere Application Server profiles and uninstall the feature. See “Uninstalling the IBM Integration Administration feature: z/OS” on page 100.
2. Copy the required version of the repository file onto your z/OS system. You can get the repository file from an IBM Integration Bus installation.
3. Install the repository file by completing the instructions at “Installing the IBM Integration feature: z/OS” on page 97.

Related tasks:

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

“Installing the IBM Integration feature: z/OS” on page 97

Learn about how to install IBM Integration Administration for WebSphere Application Server on z/OS.

Uninstalling

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

Before you begin

Before you start:

Check that your user ID has the correct authority to complete this task. The requirements for each platform are defined in “Installation and uninstallation authorization” on page 661.

About this task

Follow the appropriate link for the task that you want to complete:

- “Uninstalling the Integration Bus component”
- “Uninstalling the IBM Integration Toolkit” on page 159
- “Uninstalling the IBM Integration Explorer” on page 164
- “Uninstalling the IBM Integration ODBC Database Extender” on page 159

What to do next

You might also want to uninstall complementary products. See the documentation provided by those products to complete this task:

- IBM DB2: Access the appropriate information for your installation:
 - DB2 V9.1 product documentation (distributed systems)
 - DB2 V9.5 product documentation (distributed systems)
 - DB2 Information Center (z/OS) (Versions 8 and 9)
- IBM License Metric Tool: access the IBM License Metric Tool product documentation.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.



Resolving problems when uninstalling

Work through the advice provided to help you to deal with problems that can arise when the product is uninstalled.

“Uninstalling the IBM Integration ODBC Database Extender” on page 159

How you uninstall the IBM Integration ODBC Database Extender.

Related reference:

“Installation,” on page 619

Use the reference information in this section to understand installation requirements, installation options, and how they affect your computer.

“Installation and uninstallation authorization” on page 661

Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

Uninstalling the Integration Bus component

You can uninstall the Integration Bus component on distributed systems in a number of ways.

About this task

This section describes this task for the following systems:

- AIX
- HP-UX
- Linux
- Solaris

- Windows

If you are uninstalling on Linux or UNIX systems, make sure that the directory /var is not full; the uninstallation wizard requires space in this directory while it is running.

Procedure

1. Check that your user ID has the correct authority to uninstall the Integration Bus component, or to uninstall service that you have applied. The requirements are defined in “Installation and uninstallation authorization” on page 661.
2. By default, the uninstaller launches in the graphical, console, or silent mode previously used to install the Integration Bus component. For example, if the installation uses the silent interface, the uninstaller defaults to using the silent interface. However, the uninstaller mode can be overridden on the command line by specifying the `-i swing`, `-i console`, or `-i silent` flags. Decide which mode of uninstallation you want to use. The choices are explained in “How to install and uninstall the Integration Bus component” on page 653.
3. If you have multiple installations of the Integration Bus component on your system, see “How to uninstall multiple installations of the Integration Bus component” on page 655.
4. Follow the uninstallation instructions provided in the appropriate topic:
 - “Uninstalling the Integration Bus component by using the graphical interface” on page 153
 - “Uninstalling the Integration Bus component by using the console interface” on page 154
 - “Uninstalling the Integration Bus component by using the silent interface” on page 156
 - “Uninstalling service for the Integration Bus component” on page 158

Results

If, in exceptional circumstances, the uninstallation wizard fails to remove the product successfully, you can remove the product manually. Contact your IBM Service Center for assistance if a failure occurs.

For information about removing service updates, see “Uninstalling service for the Integration Bus component” on page 158.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Applying service to the Integration Bus component” on page 130

Apply maintenance updates and program fixes to the Integration Bus component.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.



Resolving problems when uninstalling

Work through the advice provided to help you to deal with problems that can arise when the product is uninstalled.

Related reference:

“Installation,” on page 619

Use the reference information in this section to understand installation requirements, installation options, and how they affect your computer.

“Installation and uninstallation authorization” on page 661

Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

Uninstalling the Integration Bus component by using the graphical interface

Remove the Integration Bus component by using the graphical interface.

Before you begin

Always use the IBM Integration Bus uninstallation wizard to remove the Integration Bus component, unless otherwise stated. Do not use the uninstallation program that is provided by the operating system; for example, **geninstall** on AIX, or **swremove** on HP-UX.

About this task

For information about the graphical interface and alternative interfaces available, see “How to install and uninstall the Integration Bus component” on page 653.

To start the graphical interface of the uninstaller, complete the following steps:

Procedure

1. Stop all running processes that are associated with brokers before you uninstall. Use the **mqsilist** command to check which brokers are running. Use the **mqsistop** command to stop each broker.
2. On Windows, Linux on x86, and Linux on x86-64, end all active sessions of the IBM Integration Toolkit.
3. If you do not plan to reinstall brokers on this computer, delete all the brokers that are associated with this installation by using the **mqsdeletebroker** command. When you delete a broker, all its associated data and resources are deleted.
4. By default, the uninstaller launches in the graphical, console, or silent mode previously used to install the Integration Bus component. For example, if the installation uses the silent interface, the uninstaller defaults to using the silent interface. The uninstaller mode can be overridden on the command line by specifying the **-i swing**, **-i console**, or **-i silent** flags. However, the **-i swing** flag is optional for the GUI interface; the uninstaller defaults to GUI mode when no flag is specified. Run the following command from outside the installation directory, specifying the full path. This method of invocation ensures that the full directory structure is removed during uninstallation. Where *install_dir* is the home directory of your IBM Integration Bus installation:

- On Windows:

```
\install_dir\Uninstaller\Uninstaller.exe
```

- On Linux:

```
/install_dir/Uninstaller/Uninstaller
```

If you accepted the default location during installation, this directory is as follows. The default directory includes the version, release, modification, and fix of the product that you are installing in the format v.r.m.f (version.release.modification.fix):

Linux /opt/ibm/mqsi/v.r.m.f

UNIX /opt/IBM/mqsi/v.r.m.f

Windows 32-bit

C:\Program Files\IBM\MQSI\v.r.m.f

Windows 64-bit

C:\Program Files\IBM\MQSI\v.r.m.f for the 64-bit version of IBM Integration Bus

C:\Program Files (x86)\IBM\MQSI\v.r.m.f for the 32-bit version of IBM Integration Bus

These locations define the default value of *install_dir* on each platform.

Alternatively, if you have a single installation of the Integration Bus component on Windows, you can navigate to **Start > Settings > Control Panel >**

Add/Remove Programs. Select IBM Integration Bus and click **Change/Remove**.

The uninstaller graphical interface opens.

5. Follow the prompts to complete uninstallation.

Related tasks:

“Uninstalling the Integration Bus component by using the silent interface” on page 156

Remove the Integration Bus component without user interaction by using the silent interface.

“Uninstalling the Integration Bus component by using the console interface”

Remove the Integration Bus component by using the console interface.



Resolving problems when uninstalling

Work through the advice provided to help you to deal with problems that can arise when the product is uninstalled.

Related reference:



mqsdeletebroker command

Use the **mqsdeletebroker** command to delete a named broker. The command also deletes the queues on the associated queue manager (created when the broker was created). You can also specify that the queue manager is to be deleted.



mqsilist command

Use the **mqsilist** command to list installed brokers and their associated resources.



mqsistop command

Use the **mqsistop** command to stop the specified component.

Uninstalling the Integration Bus component by using the console interface

Remove the Integration Bus component by using the console interface.

Before you begin

Always use the IBM Integration Bus uninstallation wizard to remove the Integration Bus component, unless otherwise stated. Do not use the uninstallation program that is provided by the operating system; for example, **geninstall** on AIX, or **swremove** on HP-UX.

About this task

For information about the console interface and alternative interfaces available, see “How to install and uninstall the Integration Bus component” on page 653.

To start the console interface of the uninstaller, complete the following steps:

Procedure

1. Stop all running processes that are associated with brokers before you uninstall. Use the **mqsilist** command to check which brokers are running. Use the **mqsisstop** command to stop each broker.
2. On Windows, Linux on x86, and Linux on x86-64, end any active sessions of the IBM Integration Toolkit.
3. If you do not plan to reinstall brokers on this computer, delete all the brokers that are associated with this installation by using the **mqsideletebroker** command. When you delete a broker, all its associated data and resources are deleted.
4. By default, the uninstaller launches in the graphical, console, or silent mode previously used to install the Integration Bus component. For example, if the installation uses the silent interface, the uninstaller defaults to using the silent interface. However, the uninstaller mode can be overridden on the command line by specifying the **-i swing**, **-i console**, or **-i silent** flags. Run the uninstallation program from outside the installation directory, specifying the full path to ensure that the folders are removed during uninstallation, where *install_dir* is the home directory of your IBM Integration Bus installation:

- On Windows:

```
\install_dir\Uninstaller\Uninstaller.exe -i console
```

- On distributed operating systems:

```
/install_dir/Uninstaller/Uninstaller -i console
```

If you accepted the default location during installation, this directory is as follows. The default directory includes the version, release, modification, and fix of the product that you are installing in the format *v.r.m.f* (version.release.modification.fix):

Linux /opt/ibm/mqsi/*v.r.m.f*

UNIX /opt/IBM/mqsi/*v.r.m.f*

Windows 32-bit

C:\Program Files\IBM\MQSI*v.r.m.f*

Windows 64-bit

C:\Program Files\IBM\MQSI*v.r.m.f* for the 64-bit version of IBM Integration Bus

C:\Program Files (x86)\IBM\MQSI*v.r.m.f* for the 32-bit version of IBM Integration Bus

These locations define the default value of *install_dir* on each platform.

5. When started, the uninstaller displays the location from which the uninstaller will remove the Integration Bus component. Press **Enter** to confirm that you want to uninstall the product and that the location it is uninstalling from is correct. The uninstallation starts. A brief message displays when the uninstallation is complete.

Related tasks:

“Uninstalling the Integration Bus component by using the graphical interface” on page 153

Remove the Integration Bus component by using the graphical interface.

“Uninstalling the Integration Bus component by using the silent interface”
Remove the Integration Bus component without user interaction by using the silent interface.

“Uninstalling the Integration Bus component by using the console interface” on page 154
Remove the Integration Bus component by using the console interface.



Resolving problems when uninstalling

Work through the advice provided to help you to deal with problems that can arise when the product is uninstalled.

Related reference:



mqsdeletebroker command

Use the **mqsdeletebroker** command to delete a named broker. The command also deletes the queues on the associated queue manager (created when the broker was created). You can also specify that the queue manager is to be deleted.



mqsilist command

Use the **mqsilist** command to list installed brokers and their associated resources.



mqsistop command

Use the **mqsistop** command to stop the specified component.

Uninstalling the Integration Bus component by using the silent interface

Remove the Integration Bus component without user interaction by using the silent interface.

Before you begin

Always use the IBM Integration Bus uninstallation wizard to remove the Integration Bus component, unless otherwise stated. Do not use the uninstallation program that is provided by the operating system; for example, **geninstall** on AIX, or **swremove** on HP-UX.

If you use the silent interface to uninstall the Integration Bus component, the wizard always uninstalls the Integration Bus component from the last known Version 9.0 installation location (that is, the most recent installation), regardless of the location of the uninstallation wizard that you start. If you want to remove the Integration Bus component from an earlier Version 9.0 installation, use the console or graphical interface.

About this task

You can remove the Integration Bus component without user interaction. This process is called unattended (or silent) removal. This topic describes how to run the uninstallation program silently without a response file; the program assumes all the default values.

If you want to complete a silent uninstallation, but have non-default requirements, you can use a response file. A response file is a text file that contains values for the options that you select when you uninstall IBM Integration Bus. For more details of how to create, edit, and employ a response file to specify your requirements, see Using response files.

For information about the silent interface and alternative interfaces available, see “How to install and uninstall the Integration Bus component” on page 653.

To start the silent interface of the uninstaller, complete the following steps:

Procedure

1. Stop all running processes that are associated with brokers before you uninstall. Use the **mqsilist** command to check which brokers are running. Use the **mqsistop** command to stop each broker.
2. On Windows, Linux on x86, and Linux on x86-64, end any active sessions of the IBM Integration Toolkit.
3. If you do not plan to reinstall brokers on this computer, delete all the brokers that are associated with this installation by using the **mqsdeletebroker** command. When you delete a broker, all its associated data and resources are deleted.
4. By default, the uninstaller launches in the graphical, console, or silent mode previously used to install the Integration Bus component. For example, if the installation uses the silent interface, the uninstaller defaults to using the silent interface. However, the uninstaller mode can be overridden on the command line by specifying the **-i swing**, **-i console**, or **-i silent** flags. Run the following command from outside the installation directory, specifying the full path (this method of invocation ensures that the full directory structure is removed during uninstallation), where *install_dir* is the home directory of your IBM Integration Bus installation:

- On Windows:

```
install_dir\Uninstaller\Uninstaller.exe -i silent
```

- On distributed operating systems:

```
install_dir/Uninstaller/Uninstaller -i silent
```

If you accepted the default location during installation, this directory is as follows. The default directory includes the version, release, modification, and fix of the product that you are installing in the format *v.r.m.f* (version.release.modification.fix):

Linux /opt/ibm/mqsi/*v.r.m.f*

UNIX /opt/IBM/mqsi/*v.r.m.f*

Windows 32-bit

C:\Program Files\IBM\MQSI*v.r.m.f*

Windows 64-bit

C:\Program Files\IBM\MQSI*v.r.m.f* for the 64-bit version of IBM Integration Bus

C:\Program Files (x86)\IBM\MQSI*v.r.m.f* for the 32-bit version of IBM Integration Bus

These locations define the default value of *install_dir* on each platform. The uninstallation program runs without interaction.

Related tasks:

“Uninstalling the Integration Bus component by using the graphical interface” on page 153

Remove the Integration Bus component by using the graphical interface.

“Uninstalling the Integration Bus component by using the console interface” on page 154

Remove the Integration Bus component by using the console interface.



Resolving problems when uninstalling

Work through the advice provided to help you to deal with problems that can arise when the product is uninstalled.

Related reference:



mqsdeletebroker command

Use the **mqsdeletebroker** command to delete a named broker. The command also deletes the queues on the associated queue manager (created when the broker was created). You can also specify that the queue manager is to be deleted.



mqsilist command

Use the **mqsilist** command to list installed brokers and their associated resources.



mqsistop command

Use the **mqsistop** command to stop the specified component.

Uninstalling service for the Integration Bus component

On some platforms, you can remove fixes that you have applied.

About this task

Distributed systems

You cannot remove individual fixes that you have applied to the Integration Bus component on distributed systems. If you want to restore a system to a previous service level, you must uninstall the product before you reinstall the required level of service.

If you have a single installation on your computer, create and retain a backup image, or keep previous GA or fix pack images or media, in case you experience problems after you have installed service. If you have multiple installations, each one can be at a different service level and you can transfer the broker to work with an alternative installation at a newer or older level of service. This option is described in “Applying service to the Integration Bus component” on page 130.

When you uninstall the product, the components that you have created are not lost, and you can continue to use these components and associated resources with the reinstalled product. However, you might want to take a backup of the installation and work path directories before you start this procedure.

To restore a system on which you have a single installation to a previous service level:

1. Uninstall the entire product.
2. Reinstall the GA level of the product. This step is required to ensure that you comply with your license agreement.
3. Install the required level of fix pack.

z/OS On z/OS systems, you can uninstall service levels under some circumstances during the installation phase; for further information, see “Installing service on z/OS” on page 139.

Related concepts:

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Applying service to the Integration Bus component” on page 130
Apply maintenance updates and program fixes to the Integration Bus component.
“Uninstalling the Integration Bus component” on page 151
You can uninstall the Integration Bus component on distributed systems in a number of ways.



Resolving problems when uninstalling
Work through the advice provided to help you to deal with problems that can arise when the product is uninstalled.

Uninstalling the IBM Integration ODBC Database Extender

How you uninstall the IBM Integration ODBC Database Extender.

Procedure

To uninstall the IBM Integration ODBC Database Extender, change your directory to `Uninstall_IE02` within the installation path `IE02 ($IE02_PATH)`, and run the following command:

```
./Uninstall_IE02
```

This command launches the uninstaller in the same mode that it was previously installed. For example, in Graphical, Console, or Silent mode.
The uninstaller does not delete the `/var/mqsi/common/profiles/IE02.sh` file. You can remove this file manually after you have removed the SupportPac, although no problems are caused if the file is not removed.

Related tasks:

“Configuring the IBM Integration ODBC Database Extender” on page 78
IBM Integration ODBC Database Extender encapsulates the unixODBC driver manager, which is an implementation of the Open DataBase Connectivity interface for UNIX systems, and this topic describes how you configure it.
“Installing the IBM Integration ODBC Database Extender” on page 76
IBM Integration ODBC Database Extender encapsulates the unixODBC driver manager, which is an implementation of the Open DataBase Connectivity interface for UNIX systems. This topic describes how you install IBM Integration ODBC Database Extender.

Uninstalling the IBM Integration Toolkit

Choose the method that you want to use to uninstall the IBM Integration Toolkit and follow the instructions to remove the component.

About this task

Complete the following steps to uninstall the IBM Integration Toolkit:

Procedure

1. Check that your user ID has the correct authority to uninstall the IBM Integration Toolkit or to uninstall service that you have applied. The requirements are defined in “Installation and uninstallation authorization” on page 661.
2. If you are uninstalling the IBM Integration Toolkit component, decide which mode of uninstallation you want to use. The choices are explained in “How to install and uninstall the IBM Integration Toolkit” on page 657.

3. Follow the instructions provided in the appropriate topic:
 - “Uninstalling the IBM Integration Toolkit in graphical mode”
 - “Uninstalling the IBM Integration Toolkit in silent mode” on page 162
 - “Uninstalling service from the IBM Integration Toolkit” on page 163

Results

When you uninstall the IBM Integration Toolkit component, entries are removed from the system menus:

Linux If you have installed the IBM Integration Toolkit more than once on Linux on x86, the main menu shows a single entry, regardless of the number of installations that you have completed; the entry always accesses the last installation that you completed. When you uninstall this component, the single entry is removed, and you cannot access remaining installations using the main menu. Navigate to the correct package group installation directory and start the IBM Integration Toolkit from the command line.

Windows Entries in the Windows Start menu are grouped by package group. When you uninstall this component, the entry within the appropriate package group is correctly removed.

Related tasks:

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Applying service to the IBM Integration Toolkit” on page 141

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

“Uninstalling service from the IBM Integration Toolkit” on page 163

Remove maintenance or fixes from the IBM Integration Toolkit.

Related reference:

“Installation Guide” on page 32

Installation information for IBM Integration Bus is provided in the online information center.

“Installation and uninstallation authorization” on page 661

Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

Uninstalling the IBM Integration Toolkit in graphical mode

The graphical interface guides you through the process of uninstalling the IBM Integration Toolkit.

About this task

For information about the modes that are available, see “How to install and uninstall the IBM Integration Toolkit” on page 657.

Complete the following steps:

Procedure

1. Log on to the computer on which IBM Integration Toolkit is installed with a user ID that has the correct authority. The authority requirements are defined in “Installation and uninstallation authorization” on page 661.

2. If you choose a user ID that does not have root or administrator authority, then prior to uninstalling a user with root or administrator authority must run the following pre-uninstallation script file from the IBM Integration Toolkit home directory:
 - **Linux** <IIT90_Home>/wmbt/postconfig/preUninstall_admin.sh
 - **Windows** <IIT90_Home>\wmbt\postconfig\preUninstall_admin.bat
3. Start the IBM Installation Manager.
 - **Linux** Choose one of the following options:
 - The main menu. You can use this option only if you are logged in as root. If you have logged in as another user, you cannot use the main menu links, even if you become root.
 - On Red Hat, click **System > IBM Installation Manager**.
 - On SUSE, click **System Tools > IBM Installation Manager**.
 - Navigate to the `eclipse` subdirectory in the directory in which Installation Manager is installed and run the command `IBMIM`.
 - On the command line, start the IBM Integration Toolkit uninstallation wizard (where `package_group_dir` is the directory in which you have installed the package group):


```
package_group_dir/uninstall/uninstallMB.sh
```

 For example:


```
/opt/IBM/IntegrationToolkit90/uninstall/uninstallMB.sh
```
 - **Windows** Choose one of the following options:
 - Click **Start > Programs > IBM Installation Manager > IBM Installation Manager**.
 - Click **Start > Settings > Control Panel > Add/Remove Programs**. Open the package group and select the package group in which you have installed IBM Integration Toolkit.
 - Navigate to the `eclipse` subdirectory in the directory in which Installation Manager is installed and run the command `IBMIM.exe`.
 - On the command line, start the IBM Integration Toolkit uninstallation wizard (where `package_group_dir` is the directory in which you have installed the package group):


```
package_group_dir\uninstall\uninstallMB.bat
```
4. If you started Installation Manager from the Start or main menu, or by running `IBMIM` or `IBMIM.exe`, click **Uninstall Packages**. The Uninstall Package panel displays.
5. Select the IBM Integration Toolkit, then click **Next**. The Uninstall Summary panel is displayed.
6. Check that the items listed are those that you want to uninstall. Click **Back** if you want to make changes, or click **Uninstall** to continue. The uninstallation process starts.
7. Follow the guidance through the series of panels, providing input and responses as required.
8. When the process completes, check the log file for errors. The file `YYYYMMDD_TIME.xml` is written to the following directory:
 - **Linux** `/var/ibm/InstallationManager/logs`
 - **Windows** `C:\ProgramData\IBM\Installation Manager\logs`.
9. Restart your computer to complete the uninstallation of the IBM Integration Toolkit.

Related tasks:

“Uninstalling the IBM Integration Toolkit in silent mode”

Uninstall the IBM Integration Toolkit without user interaction.

“Uninstalling service from the IBM Integration Toolkit” on page 163

Remove maintenance or fixes from the IBM Integration Toolkit.

Related reference:

“How to install and uninstall the IBM Integration Toolkit” on page 657

Install and uninstall the IBM Integration Toolkit by using one of two interfaces.

“Installation and uninstallation authorization” on page 661

Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

Uninstalling the IBM Integration Toolkit in silent mode

Uninstall the IBM Integration Toolkit without user interaction.

About this task

For information about alternative modes that are available, see “How to install and uninstall the IBM Integration Toolkit” on page 657.

Complete the following steps:

Procedure

1. Log on to the computer on which IBM Integration Toolkit is installed. You have the option to uninstall with a user ID that does not have root or administrator authority.
2. If you choose a user ID that does not have root or administrator authority, then prior to uninstalling a user with root or administrator authority must run the following pre-uninstallation script file from the IBM Integration Toolkit home directory:
 - **Linux** `<IIT90_Home>/wmbt/postconfig/preUninstall_admin.sh`
 - **Windows** `<IIT90_Home>\wmbt\postconfig\preUninstall_admin.bat`
3. To silently uninstall the IBM Integration Toolkit, start IBM Installation Manager by using the supplied script files.
 - a. Navigate to the uninstallation directory and run the script file:
 - **Linux** `package_group_dir/uninstall/uninstallMB-silent.sh`
For example:
`/opt/IBM/IntegrationToolkit90/uninstall/uninstallMB-silent.sh`
 - **Windows** `package_group_dir\uninstall\uninstallMB-silent.bat`
`C:\Program Files\IBM\IntegrationToolkit90\uninstall\uninstallMB-silent.bat`
4. When the process completes, check the log file for errors. The file `YYYYMMDD_TIME.xml` is written to the following directory:
 - **Linux** `/var/ibm/InstallationManager/logs.`
 - **Windows** `C:\ProgramData\IBM\Installation Manager\logs.`
5. Restart your computer to complete the uninstallation of the IBM Integration Toolkit.
6. Delete the uninstallation directory if it still exists.

Related tasks:

“Uninstalling the IBM Integration Toolkit in graphical mode” on page 160
The graphical interface guides you through the process of uninstalling the IBM Integration Toolkit.

“Uninstalling service from the IBM Integration Toolkit”
Remove maintenance or fixes from the IBM Integration Toolkit.

Related reference:

“How to install and uninstall the IBM Integration Toolkit” on page 657
Install and uninstall the IBM Integration Toolkit by using one of two interfaces.

“Installation and uninstallation authorization” on page 661
Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

Uninstalling service from the IBM Integration Toolkit

Remove maintenance or fixes from the IBM Integration Toolkit.

Before you begin

Linux To remove service from the IBM Integration Toolkit on Linux on x86 and Linux on x86-64, your user ID must have root authority. Follow your local security guidelines to acquire root authority; either log in as root, or log in as another user and become root.

Windows To remove service from the IBM Integration Toolkit on Windows, your user ID must have Administrator authority.

About this task

You can remove IBM Integration Bus interim fix and service updates from the toolkit using IBM Installation Manager.

To uninstall service:

Procedure

1. Close all sessions of the IBM Integration Toolkit on this computer.
2. Start Installation Manager in one of the following ways:
 - From the Linux on x86 and Linux on x86-64 main menu, or the Windows Start menu:
 - On Linux, you can use this option only if you are logged in as root. If you have logged in as another user, you cannot use the main menu links, even if you become root.
 - On Red Hat, click **System > IBM Installation Manager**.
 - On SUSE, click **System Tools > IBM Installation Manager**.
 - On Windows, click **Start > Programs > IBM Installation Manager > IBM Installation Manager**.
 - Navigate to the `eclipse` subdirectory in the directory in which Installation Manager is installed and run the following command:
 - **Linux** `IBMIM`
 - **Windows** `IBMIM.exe`

Installation Manager starts and displays its initial panel.

3. Click **Roll Back Packages**, then select the package group in which you have installed the IBM Integration Toolkit.

4. Click **Next**. The Roll Back Packages panel displays the products that you have installed in the selected group. The service updates that you have applied to any item in the list are displayed under the item to which they apply.
5. If updates are installed, select the updates that you want to remove from each particular installation. A summary of the selected updates is displayed. Click **more details...** to open the readme file associated with the fix.
6. Click **Next** to continue. Installation Manager checks that no IBM Integration Toolkit sessions are running. If the check fails, an error message is displayed and you cannot continue with the operation. Close all active sessions, then return to this step and click **Next** again.
Installation Manager displays the exact identification of the fixes that you selected; check that what is listed is correct. If you want to change your choices, click **Back** to view the list of fixes again.
7. Click **Roll Back**. Installation Manager starts to remove the selected fixes. A progress bar is displayed so that you can monitor the request. A completion message is displayed; click **View log** to read the report that is generated by the removal.
8. If the fix has been removed successfully, click **Finish** to return to the initial panel. If the removal failed, examine the contents of the log file to determine the cause of the error.
9. Click **File > Exit** to close Installation Manager. Verify that the changes have worked; see “Verifying your installation by using the IBM Integration Toolkit” on page 105.

Related tasks:

“Applying service to the IBM Integration Toolkit” on page 141

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

“Uninstalling the IBM Integration Toolkit” on page 159

Choose the method that you want to use to uninstall the IBM Integration Toolkit and follow the instructions to remove the component.

“Verifying your installation by using the IBM Integration Toolkit” on page 105

Use the instructions in this tutorial to verify your installation of IBM Integration Bus and learn how to run samples with the IBM Integration Toolkit.

Uninstalling the IBM Integration Explorer

You can uninstall the IBM Integration Explorer by using the graphical installation wizard, by using the console mode of the installation wizard, or by using the installation wizard in silent mode.

By default, the uninstaller launches in the graphical, console, or silent mode previously used to install the IBM Integration Explorer. For example, if the installation uses the silent interface, the uninstaller defaults to using the silent interface. However, the uninstaller mode can be overridden on the command line by specifying the **-i swing**, **-i console**, or **-i silent** flags. Decide which mode of uninstallation you want to use.

To uninstall the IBM Integration Explorer by using the graphical installation wizard, click the link for your operating system and follow the steps in the topic. To uninstall the IBM Integration Explorer by using the console mode, click the appropriate link and follow the steps in the topic. The steps for console mode are applicable on Windows, Linux on x86, and Linux on x86-64. You can also uninstall the IBM Integration Explorer by using the installation wizard in silent mode.

- “Uninstalling the IBM Integration Explorer on Windows” on page 165

- “Uninstalling the IBM Integration Explorer on Linux” on page 166
- “Uninstalling IBM Integration Explorer in console mode” on page 166
- “Uninstalling IBM Integration Explorer in silent mode” on page 167

To uninstall service from the IBM Integration Explorer, you must reinstall the IBM Integration Explorer to revert to a previous service level. For more information about uninstalling service, see “Uninstalling service from the IBM Integration Explorer” on page 168.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Installation,” on page 619

Use the reference information in this section to understand installation requirements, installation options, and how they affect your computer.

“Installation and uninstallation authorization” on page 661

Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

Uninstalling the IBM Integration Explorer on Windows

You can uninstall the IBM Integration Explorer by using the graphical installation wizard.

About this task

Complete the following steps to uninstall the IBM Integration Explorer by using the graphical installation wizard on Windows.

Procedure

1. From the Windows task bar, click **Start > Control Panel**, or **Start > Settings > Control Panel**.
2. Click **Add or Remove Programs**.
3. Select **IBM Integration Explorer** and click **Change/Remove**. The IBM Integration Explorer uninstaller opens.
4. Click **Uninstall** to uninstall the IBM Integration Explorer. The uninstaller lists any items that cannot be uninstalled. You can delete these items manually.
5. When the uninstallation process has completed, click **Done** to close the uninstaller.

Related concepts:

“Uninstalling the IBM Integration Explorer” on page 164

You can uninstall the IBM Integration Explorer by using the graphical installation wizard, by using the console mode of the installation wizard, or by using the installation wizard in silent mode.

Related tasks:

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Uninstalling the IBM Integration Explorer on Linux” on page 166

You can uninstall the IBM Integration Explorer by using the graphical installation

wizard.

Uninstalling the IBM Integration Explorer on Linux

You can uninstall the IBM Integration Explorer by using the graphical installation wizard.

About this task

Complete the following steps to uninstall the IBM Integration Explorer by using the graphical installation wizard on Linux. To uninstall successfully on Linux systems, you must have write permissions to the installation directory. You might also need root administrator privileges to complete this task.

Procedure

1. Navigate to the uninstallation directory. The default location on Linux is `/opt/IBM/IBExplorer/Uninstall_IBM_Integration_Explorer`. You can use your native file explorer to run the file in this directory.
2. Run the `./Uninstall_IBM_Integration_Explorer` file. The uninstaller launches.
3. Click **Uninstall** to uninstall the IBM Integration Explorer. The uninstaller lists any items that cannot be uninstalled. You can delete these items manually.
4. When the uninstallation process is completed, click **Done** to close the uninstaller.

Related concepts:

“Uninstalling the IBM Integration Explorer” on page 164

You can uninstall the IBM Integration Explorer by using the graphical installation wizard, by using the console mode of the installation wizard, or by using the installation wizard in silent mode.

Related tasks:

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Uninstalling the IBM Integration Explorer on Windows” on page 165

You can uninstall the IBM Integration Explorer by using the graphical installation wizard.

Uninstalling IBM Integration Explorer in console mode

You can uninstall the IBM Integration Explorer using the console mode of the installation wizard.

About this task

Use the following steps to uninstall the IBM Integration Explorer using the console mode of the installation wizard on Windows and Linux. To uninstall successfully on Linux systems, you need write permissions to the installation directory. You are likely to require root administrator privileges for this task.

Procedure

1. Navigate to the uninstallation directory:
 - The default location on Windows is `C:\Program Files\IBM\IBExplorer\Uninstall_IBM_Integration_Explorer`.
 - The default location on Linux is `/opt/IBM/IBExplorer/Uninstall_IBM_Integration_Explorer`. You can also use your native file explorer to run this file.

2. Run the following command: A new uninstaller command window launches:
 - On Windows, the command to run a console uninstallation is:
`Uninstall_IBM_Integration_Explorer.exe -i console.`
 - On Linux, the command to run a console uninstallation is:
`./Uninstall_IBM_Integration_Explorer -i console.`

The uninstaller launches.

3. Press Enter to uninstall the IBM Integration Explorer. The uninstaller lists any items that could not be uninstalled. You can delete these items manually.
4. When the uninstallation has completed, press Enter to exit the uninstaller.

Related concepts:

“Uninstalling the IBM Integration Explorer” on page 164

You can uninstall the IBM Integration Explorer by using the graphical installation wizard, by using the console mode of the installation wizard, or by using the installation wizard in silent mode.

Related tasks:

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Uninstalling the IBM Integration Explorer on Windows” on page 165

You can uninstall the IBM Integration Explorer by using the graphical installation wizard.

“Uninstalling the IBM Integration Explorer on Linux” on page 166

You can uninstall the IBM Integration Explorer by using the graphical installation wizard.

Uninstalling IBM Integration Explorer in silent mode

You can uninstall the IBM Integration Explorer using the installation wizard in silent mode.

About this task

Use the following steps to uninstall the IBM Integration Explorer using the silent mode on Windows and Linux.

Procedure

1. On a command line, navigate to the uninstallation directory for IBM Integration Explorer in the file system. The default uninstallation directory for the IBM Integration Explorer on Windows is `C:\Program Files\IBM\IBExplorer\Uninstall_IBM_Integration_Explorer`. The default uninstallation directory for the IBM Integration Explorer on Linux systems is `/opt/IBM/IBExplorer/Uninstall_IBM_Integration_Explorer`.
2. Enter the following command on a command line to start the uninstaller in silent mode: `Uninstall_IBM_Integration_Explorer.exe -i silent` If you run the silent uninstallation command directly from a command line, the command prompt returns immediately, and the installation wizard completes the uninstallation as a background task. If you run the silent uninstallation command as part of a batch file or script, control is returned to the batch file or script after the uninstallation has completed.

Results

The IBM Integration Explorer files have been removed from your system.

Related concepts:

“Uninstalling the IBM Integration Explorer” on page 164

You can uninstall the IBM Integration Explorer by using the graphical installation wizard, by using the console mode of the installation wizard, or by using the installation wizard in silent mode.

Related tasks:

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Uninstalling the IBM Integration Explorer on Windows” on page 165

You can uninstall the IBM Integration Explorer by using the graphical installation wizard.

“Uninstalling the IBM Integration Explorer on Linux” on page 166

You can uninstall the IBM Integration Explorer by using the graphical installation wizard.

Uninstalling service from the IBM Integration Explorer

You must reinstall the IBM Integration Explorer to revert to a previous service level.

About this task

You cannot remove individual fixes that you have applied to the IBM Integration Explorer. If you want to restore a system to a previous service level, you must reinstall the desired level of service. If you have a single installation on your computer, create and retain a backup image, or keep previous GA or fix pack images or media, in case of problems after you have installed service. To restore a Windows or Linux system on which you have an installation of the IBM Integration Explorer to a previous service level, you have two options:

- Uninstall the current level, then reinstall the previous service level of the IBM Integration Explorer.
- Install the previous service level of the IBM Integration Explorer to a new location on your system.

To uninstall the current level of service from the IBM Integration Explorer, and return to a previous level use the following instructions:

Procedure

1. Uninstall the IBM Integration Explorer.
2. Reinstall the GA level of the IBM Integration Explorer.
3. Install the desired level of fix pack.

Related concepts:

“Uninstalling the IBM Integration Explorer” on page 164

You can uninstall the IBM Integration Explorer by using the graphical installation wizard, by using the console mode of the installation wizard, or by using the installation wizard in silent mode.

Related tasks:

“Applying service to the IBM Integration Explorer” on page 146

You can apply maintenance or fixes to the IBM Integration Explorer.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

“Uninstalling the IBM Integration Explorer on Windows” on page 165

You can uninstall the IBM Integration Explorer by using the graphical installation wizard.

“Uninstalling the IBM Integration Explorer on Linux” on page 166

You can uninstall the IBM Integration Explorer by using the graphical installation wizard.

Chapter 3. Migrating

To migrate a broker domain to IBM Integration Bus Version 9.0, plan your migration strategy, perform pre-migration tasks, migrate your domain components, then complete post-migration tasks.

Before you begin

Migration information is regularly updated on the IBM Integration Bus support web page with the latest details available. For example, see Knowledge Collection: Migration for WebSphere Message Broker for a compilation of links to technotes, webcast replays, developerWorks articles, and publications that cover the migration of WebSphere Message Broker.

Procedure

1. Check that your current installation of WebSphere Message Broker (Version 8.0, Version 7.0 or Version 6.1) is at a supported level for migration. Details are provided in “Supported migration paths” on page 177.
2. Read “Preparing for migration” on page 172 to plan your migration.

Related concepts:

“Operation modes” on page 5

The operation mode that you can use for your broker is determined by the license that you purchase.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188

Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206

Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 6.1 WebSphere Message Broker component” on page 225

Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.

Related information:

 [IBM Integration Bus requirements](#)

 [IBM Integration Bus Support](#)

[Installing and Migrating \(EPUB\)](#)

[Installing and Migrating \(Kindle MOBI\)](#)

Preparing for migration

Plan the order and extent of the migration of components and resources to IBM Integration Bus Version 9.0.

Before you begin

Before you start:

Check that your current installation of WebSphere Message Broker is at a supported level for migration. Details are provided in “Supported migration paths” on page 177.

For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

You must migrate your queue managers to a supported version of WebSphere MQ before you can migrate your brokers.

All WebSphere MQ installations must be at Version 7.5.0.1 for distributed platforms, and at Version 7.1 for z/OS. For information about WebSphere MQ migration, see the WebSphere MQ information center. If you are migrating from WebSphere Message Broker Version 6.1, see “Migrating the Version 6.1 WebSphere Message Broker component” on page 225 for more information about how to manage the WebSphere MQ migration.

If your Version 6.1 configuration includes publish/subscribe applications, support has transferred to WebSphere MQ. You must take additional steps to migrate publish/subscribe information from a WebSphere Message Broker Version 6.1 configuration. Details are provided in “Migrating publish/subscribe information to WebSphere MQ” on page 244.

About this task

To prepare for migration, complete the following steps.

Procedure

1. Understand the options that are available to install IBM Integration Bus Version 9.0 components on the same computer as components from previous versions of WebSphere Message Broker.

There are two different ways to migrate your broker components. In-place broker migration allows you to migrate your existing broker components immediately to IBM Integration Bus Version 9.0. For more information, see “In-place broker migration” on page 178.

Parallel broker migration allows you to associate the application logic on your existing broker with a new, separate Version 9.0 broker. For more information, see “Parallel broker migration” on page 180.

For more information about the options that are available for the WebSphere Message Broker and IBM Integration Bus components to coexist, see “Coexistence and migration” on page 38.

2. Migrate your WebSphere Message Broker Toolkit resources. Decide how you will use your existing development resources with Version 9.0. This can be done at any time in the migration process. You do not have to perform specific tasks to migrate your development and deployment resources, such as message flow files, message set definition files, ESQL files, XML Schema files, and broker archive files. You can use these resources with Version 9.0 immediately. However, some migration actions are performed automatically when you open or rebuild resources in the IBM Integration Toolkit; for details, see “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.
3. Decide how to migrate the WebSphere Message Broker product components.
 - a. Learn about new and changed function in Version 9.0 by reading What's new in Version 9.0?. These changes might affect how you want to use your migrated components in the future.
 - b. Review the technical changes in behavior in Version 9.0 by reading “Behavioral changes in Version 9.0” on page 174. These changes might affect your post-migration development tasks.
 - c. Plan your migration of WebSphere MQ to a supported version. It is a requirement that a supported version of WebSphere MQ is installed before you install Version 9.0.

If you are migrating from WebSphere Message Broker Version 6.1, publish/subscribe functions are no longer supported in the broker; WebSphere MQ provides this support. If your Version 6.1 applications use the publish/subscribe communication model, you must migrate your subscriptions to a supported version of WebSphere MQ before you migrate your brokers.
 - d. Check the requirements for other products on which Version 9.0 components might depend. If you configured your message flows to use external resources, such as databases, or event monitoring applications, you might have to modify your configuration. You can find details of supported versions of optional products on the IBM Integration Bus Requirements web page.

What to do next

Next: After you have planned your migration, complete the migration tasks by following the instructions in “Migration tasks” on page 182.

Related concepts:

“In-place broker migration” on page 178

IBM Integration Bus Version 9.0 supports in-place broker migration to upgrade your broker components immediately.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.

Related information:

- [📄 IBM Integration Bus requirements](#)
- [📄 IBM Integration Bus Support](#)
- [📄 WebSphere MQ Version 7.1 product documentation](#)
- [📄 WebSphere MQ Version 7.5 product documentation](#)

Behavioral changes in Version 9.0

Depending on your current version of WebSphere Message Broker, IBM Integration Bus Version 9.0 might introduce several technical changes in behavior. These changes might affect your post-migration development tasks.

If you are upgrading from Version 8.0, Version 7.0 or Version 6.1, review the following changes to see how your post-migration development tasks might be affected:

- “Web user interface not enabled on migrated brokers”
- “Actions on z/OS”

If you are upgrading from Version 7.0 or Version 6.1, also review the following changes to see how your post-migration development tasks might be affected:

- “Subflows and broker archive (BAR) files”
- “Applications and Libraries” on page 175
- “Message sets” on page 175
- “Message maps” on page 175

Web user interface not enabled on migrated brokers

The web user interface for Web administration, REST administration, and Integration Registry is not enabled on migrated brokers. The web user interface is enabled by default on new integration nodes (brokers) (and brokers that were created in Version 8.0.0.1 and later), but this interface is disabled on brokers that are migrated from Version 8.0.0.0 or earlier. Therefore, you cannot use REST applications, Web administration, or the Integration Registry with brokers that are migrated from Version 8.0.0.0 or earlier until you enable the web user interface for those brokers.

For information, see [Enabling and disabling the web user interface](#).

z/OS

Actions on z/OS

An Integration Registry is introduced in IBM Integration Bus Version 9.0. The registry requires the value of the `MAXMMAPAREA` to be set higher than the system default. The Integration Registry is always on, and so you must tune this value, even if you are not using the registry. For more information, see [Integration Registry](#)

Subflows and broker archive (BAR) files

The default option for building new BAR files in IBM Integration Bus Version 9.0 is to package message flows as source, with ESQL files also included as source. Some existing message flows might not be compatible with this default option, but you can select a compatibility option to maintain previous behavior when you

build a new BAR file. Any existing BAR files keep the previous behavior, which is to recompile ESQL into a .cmf file rather than package ESQL as separate source files.

For information about the solutions that are affected, see Adding files to a broker archive.

Applications and Libraries

You can continue to use resources from a previous version of WebSphere Message Broker by importing them into a Version 9.0 workspace. However, you cannot create a message flow project in IBM Integration Bus Version 9.0. You can continue to use message flow projects from previous versions in the same way in Version 9.0 by migrating them to integration projects. You might want to continue to use message flow projects if you are working in a team environment, for example.

For more information, see “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.

Message sets

In WebSphere Message Broker Version 8.0 and later, *message model schema* files contained in applications, integration services, and libraries are the preferred way to model messages for most data formats. Message sets are required if you use the MRM or IDOC domains. For more information about message modeling, see Message modeling concepts.

You can import message flows containing message sets from WebSphere Message Broker Version 7.0 into IBM Integration Bus Version 9.0. Your existing message sets can be viewed, compiled, and deployed. However, by default, your message sets are accessible in read-only mode and cannot be modified.

If you need to modify your existing message sets, or create new message sets or message definition files, you must first enable message set development in the IBM Integration Toolkit.

For more information, see Enabling message set development.

Message maps

In IBM Integration Bus Version 9.0, a graphical data mapping capability is used when you add a Mapping node to a message flow.

You can import message flows containing the following nodes, which use message mapping, from WebSphere Message Broker Version 6.1 or Version 7.0 into IBM Integration Bus Version 9.0:

- DataDelete
- DataInsert
- DataUpdate
- Extract
- Mapping
- Warehouse
- ESQL Compute node, which contains an ESQL CALL statement that invokes a message map .msgmap

The message map referenced on these Version 6.1 and Version 7.0 nodes can be viewed, compiled into a BAR file, deployed, and run. The following information must be considered however:

- The Version 6.1 or Version 7.0 message map operations are accessible only in read-only mode, and cannot be modified. To modify the transformation logic of an imported message flow that used Version 6.1 or Version 7.0 message mapping, you must convert the message map .msgmap file to a graphical data map .map file.
- To run an unconverted message map, the whole message flow or application must be deployed in a BAR file that is in compiled message format (a .cmf file) with the Compile and in-line option set. This deployment node might prevent the use of other new capabilities that require deploying as a source. For example: To use deployable subflows in a .subflow file, the solution must be deployed as source.
- The ESQL CALL statement cannot be used to invoke graphical data maps. When you convert flows that use the ESQL CALL statement, extra steps are required.

For more information, see [Converting a message map from a .msgmap file to a .map file](#).

From IBM Integration Bus Version 9.0.0.1, the behavior of graphical data maps that are used in the flow Mapping node is altered when a broker function level is specified. When the function level is changed to 9.0.0.1 or later using the **mqsichangebroker** command, the graphical data maps are prepared for execution on deployment instead of when the first message is flowed through the node.

For a full list of function level changes that affect graphical data maps and how this might affect migration, see [Message maps](#).

DFDL PIFs generated after Version 9.0.0.2 have a minimum runtime requirement of Version 9.0.0.0

In order to improve memory usage when deploying industry models, defunct information has been removed from the compiled grammar (PIF) format. This improvement means that compiled DFDL grammars from IBM Integration Bus Version 9.0.0.2 and later are not supported with runtimes from WebSphere Message Broker Version 8.0, even when a compatible function is used.

The following example demonstrates the scenario:

1. You complete an in-place upgrade of WebSphere Message Broker Version 8.0 to IBM Integration Bus Version 9.0.0.2 or later.
2. You redeploy one or more applications that use DFDL schemas while at the Version 9.0.0.2 or later level.
3. You roll back the upgrade and revert to WebSphere Message Broker Version 8.0.
4. You must now redeploy the applications that use the DFDL schemas that were redeployed at Version 9.0.0.2 or later, so that the DFDL grammar is recompiled at the Version 8 level. If you did not redeploy at the Version 9.0.0.2 or later level, then the grammar remained at the Version 8 compatible level.

There is no impact to users of IBM Integration Bus Version 9.0.0.1 and Version 9.0.0.0 that have moved to Version 9.0.0.2 and then back again. DFDL PIFs generated in IBM Integration Bus Version 9.0.0.2 or later have a minimum requirement of IBM Integration Bus Version 9.0.0.0.

Related tasks:

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

“Migrating deployable resources” on page 259

You can continue to use legacy resources in IBM Integration Bus. However, if you want to continue developing resources created in WebSphere Message Broker Version 6 and Version 7, you must migrate them.

Related reference:

“Reviewing technical changes in IBM Integration Bus Version 9.0” on page 253

Some minor changes in behavior are present in IBM Integration Bus Version 9.0 if you are migrating from Version 7.0 or Version 6.1; for example, those changes that are caused by defects that were fixed between versions.

Supported migration paths

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.

You cannot migrate from the Windows on x86 product (any version) to the Windows on x86-64 product. Similarly, you cannot migrate from the Linux on x86 product (any version) to the Linux on x86-64 product. You must re-create the integration nodes (brokers) from scratch.

For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

From ...	You can migrate to ...
WebSphere Message Broker Version 6.1 ¹	IBM Integration Bus Version 9.0
WebSphere Message Broker with Rules and Formatter Extension Version 6.1 ¹	IBM Integration Bus Version 9.0 ²
WebSphere Message Broker Version 7.0 ³	IBM Integration Bus Version 9.0
WebSphere Message Broker with Rules and Formatter Extension Version 7.0 ³	IBM Integration Bus Version 9.0 ²
WebSphere Message Broker Version 8.0 ⁴	IBM Integration Bus Version 9.0

You can migrate from WebSphere Message Broker Version 8.0, Version 7.0 or Version 6.1 only to a full edition of IBM Integration Bus Version 9.0, Remote Adapter Deployment, Express Edition, or Standard Edition. Do not migrate brokers from an earlier version to the Developer Edition. If you choose to migrate brokers by using the **mqsिमigratecomponents** command, the broker is migrated in its current mode. If you purchased the Remote Adapter Deployment, Express Edition, or Standard Edition, you must change the mode of all your migrated brokers to comply with the terms of your license. For more information, see “Changing the operation mode of your broker” on page 356.

Notes:

1. You can migrate Version 6.1 products at Fix Pack 3 (Version 6.1.0.3) or later. During the migration of WebSphere Message Broker Version 6.1 to IBM Integration Bus Version 9.0, you must upgrade your WebSphere MQ installation to a supported version. For more information about migrating from WebSphere Message Broker Version 6.1 to IBM Integration Bus Version 9.0 at the same time as upgrading your WebSphere MQ installation, see “Migrating the Version 6.1 WebSphere Message Broker component” on page 225.

If you are using WebSphere Application Server with IBM Integration Bus, or you have publish/subscribe applications that use the SubIdentity option, you must upgrade WebSphere Message Broker Version 6.1 to Fix Pack 9 before you can migrate to IBM Integration Bus Version 9.0.

2. You can migrate to IBM Integration Bus Version 9.0 only if you do not use the additional features that are provided by Rules and Formatter, or you choose not to use them after migration. To take advantage of equivalent functionality in Version 9.0, you must redesign your applications.
3. You can migrate Version 7.0 products at Fix Pack 5 (Version 7.0.0.5) or later, but you must first ensure your WebSphere MQ installation is at a supported version, by upgrading if necessary.
4. You do not need to apply maintenance to your WebSphere Message Broker Version 8.0 installation before you migrate to IBM Integration Bus Version 9.0, but you must first ensure your WebSphere MQ installation is at a supported version, by upgrading if necessary.

Related concepts:

Chapter 2, “Installing and uninstalling,” on page 31

Install and uninstall IBM Integration Bus components and service.

“Operation modes” on page 5

The operation mode that you can use for your broker is determined by the license that you purchase.

Related tasks:


Chapter 3, “Migrating,” on page 171


To migrate a broker domain to IBM Integration Bus Version 9.0, plan your migration strategy, perform pre-migration tasks, migrate your domain components, then complete post-migration tasks.

Related information:

 [IBM Integration Bus requirements](#)

 [IBM Integration Bus Support](#)

 [WebSphere MQ Version 7.1 product documentation](#)

 [WebSphere MQ Version 7.5 product documentation](#)

In-place broker migration

IBM Integration Bus Version 9.0 supports in-place broker migration to upgrade your broker components immediately.

IBM Integration Bus Version 9.0 requires a specific minimum supported version of WebSphere MQ. If you are migrating from Version 7.0 or Version 8.0, you must first install a supported version of WebSphere MQ and migrate your WebSphere MQ resources to this version. You can then install IBM Integration Bus Version 9.0. If you are migrating from WebSphere Message Broker Version 6.1, see “Migrating the Version 6.1 WebSphere Message Broker component” on page 225 for more information about how to manage the WebSphere MQ migration. For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

There are two different ways to migrate your broker components. By using in-place broker migration, you can migrate your existing broker components immediately to Version 9.0. You do not need to change your client configurations because the existing broker is migrated in-place to Version 9.0 on the same computer. However, you must stop your broker while migration takes place.

Alternatively, if you prefer not to stop the broker (to avoid problems for your business applications), or if you want to reproduce the broker function on another computer, you can use parallel broker migration to associate the application logic on your existing broker with a separate Version 9.0 integration node (broker). For more information, see “Parallel broker migration” on page 180.

Migrating your Broker components

If you are migrating from WebSphere Message Broker Version 8.0, select the topic that is appropriate to your environment:

- “Performing in-place broker migration for a Version 8.0 broker on distributed operating systems” on page 194
- “Performing in-place broker migration for a Version 8.0 broker on z/OS” on page 197
- “Migrating a multi-instance broker from WebSphere Message Broker Version 8.0” on page 198
- “Migrating a WebSphere Message Broker Version 8.0 broker that is configured on Microsoft Cluster Services (MSCS)” on page 200

If you are migrating from WebSphere Message Broker Version 7.0, select the topic that is appropriate to your environment:

- “Performing in-place broker migration for a Version 7.0 broker on distributed operating systems” on page 212
- “Performing in-place broker migration for a Version 7.0 broker on z/OS” on page 215
- “Migrating a multi-instance broker from WebSphere Message Broker Version 7.0” on page 217
- “Migrating a WebSphere Message Broker Version 7.0 broker that is configured on Microsoft Cluster Services (MSCS)” on page 218

If you are migrating from WebSphere Message Broker Version 6.1, select the topic that is appropriate to your environment:

- “Performing in-place broker migration for a Version 6.1 broker on distributed operating systems” on page 232
- “Performing in-place broker migration for a Version 6.1 broker on z/OS” on page 235
- “Migrating a WebSphere Message Broker Version 6.1 broker that is configured on Microsoft Cluster Services (MSCS)” on page 238

Related tasks:

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188
Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206
Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 6.1 WebSphere Message Broker component” on page 225
Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

Related reference:

“Memory and disk space requirements” on page 621

Check the memory and disk space that is required for your installation.

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.

Parallel broker migration

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

IBM Integration Bus Version 9.0 requires a specific minimum supported version of WebSphere MQ. You must first install a supported version of WebSphere MQ and migrate your WebSphere MQ resources to this version. You can then install IBM Integration Bus Version 9.0. If you are migrating from WebSphere Message Broker Version 6.1, see “Migrating the Version 6.1 WebSphere Message Broker component” on page 225 for more information about how to manage the WebSphere MQ migration. For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

There are two different ways to migrate your broker components. You can use parallel broker migration to associate the application logic on your existing broker with a separate Version 9.0 integration node (broker). Choose this option if you prefer not to stop the broker (to avoid problems for your business applications) or if you want to reproduce the broker function on another computer.

Alternatively, you can use in-place broker migration to migrate your existing broker components immediately to IBM Integration Bus Version 9.0. You do not need to change your client configurations because the existing broker is migrated in-place to Version 9.0 on the same computer. However, you must stop your broker while migration takes place. For more information, see “In-place broker migration” on page 178.

Parallel broker migration with previous versions of the product installed on the same computer

IBM Integration Bus Version 9.0 can coexist with any previous version of WebSphere Message Broker on the same computer during parallel broker migration.

Component names must be unique on the computer, regardless of version.

When you migrate from a previous version of WebSphere Message Broker to IBM Integration Bus Version 9.0, you do not have to uninstall the previous version of the product before you install Version 9.0. You can install Version 9.0 in a different location on the same computer, migrate your components and resources to Version 9.0, and uninstall the previous version of WebSphere Message Broker later, when you are sure that you no longer need it.

To migrate your brokers by using parallel migration, select the appropriate topic from the following list and complete the tasks that are described:

- “Performing parallel broker migration for a Version 8.0 broker” on page 201
- “Performing parallel broker migration for a Version 7.0 broker” on page 220
- “Performing parallel broker migration for a Version 6.1 broker” on page 239

For details of default locations for installations of each version, see “Coexistence and migration” on page 38.

If you attempt to install IBM Integration Bus Version 9.0 into a directory that already contains an installation of WebSphere Message Broker, you are prompted to cancel the installation and select a different directory to preserve your existing configuration.

If you install more than one version on a single computer, and you create multiple components, check that sufficient memory and disk space are available.

When you have installed Version 9.0, you can test your configuration.

Restrictions on coexistence of Version 9.0 components with previous versions

The following restrictions apply.

- The IBM Integration Toolkit Version 9.0 works only with brokers that you migrated to, or created in, Version 9.0. If you try to connect to brokers at an earlier version, the connection attempt fails. However, you can use the IBM Integration Explorer to connect to Version 9.0, Version 8.0 and Version 7.0 brokers.
- You can run all programs that you wrote to the CMP at Version 8.0, Version 7.0 or Version 6.1 to interact with brokers that you migrated to, or created in, Version 9.0. Programs that start, stop, delete, and monitor deployed resources continue to work with Version 9.0.
- You can deploy BAR files that you created in an earlier version of the WebSphere Message Broker Toolkit from the IBM Integration Toolkit Version 9.0 to a broker at Version 9.0, without making any changes to the BAR files.
- You cannot deploy resources that you compiled in the IBM Integration Toolkit Version 9.0 to brokers that are at a previous version.
- You cannot share the IBM Integration Toolkit Version 9.0 workspace or source projects with previous versions of the WebSphere Message Broker Toolkit. For more information, see “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.

Coexistence with SupportPac IS02

The SupportPac IS02, IBM Integration Explorer Plug-in, cannot coexist with the IBM Integration Explorer in IBM Integration Bus Version 9.0. If you previously installed SupportPac IS02, you must uninstall the SupportPac before you install the IBM Integration Explorer Version 9.0. To remove SupportPac IS02, delete or move the directory to which you extracted the IS02 zip file that was supplied with the SupportPac. Delete the BrokerExplorer.link file that you copied into the C:\Program Files\IBM\WebSphere MQ\eclipseSDK33\eclipse\links\ directory.

Related tasks:

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188
Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206
Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 6.1 WebSphere Message Broker component” on page 225
Migrate your Version 6.1 components and resources to IBM Integration Bus Version

9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

Related reference:

“Memory and disk space requirements” on page 621

Check the memory and disk space that is required for your installation.

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.

Migration tasks

IBM Integration Bus Version 9.0 consists of the IBM Integration Toolkit Version 9.0, Integration Bus component Version 9.0, and the IBM Integration Explorer Version 9.0. Read the following migration tasks to learn how to upgrade to these components.

- “Migrating development resources to IBM Integration Toolkit Version 9.0”
- “Migrating the Version 8.0 WebSphere Message Broker component” on page 188
- “Migrating the Version 7.0 WebSphere Message Broker component” on page 206
- “Migrating the Version 6.1 WebSphere Message Broker component” on page 225
- “Migrating the WebSphere Message Broker Explorer” on page 250

Related tasks:

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.



Resolving problems that occur when you start resources

Use the advice given here to help you to resolve common problems that can occur when you start resources.

“Installing IBM Integration Explorer” on page 84

To use IBM Integration Explorer only, without installing the complete IBM Integration Toolkit, use the IBM Integration Explorer installation wizard to install the IBM Integration Explorer.

Migrating development resources to IBM Integration Toolkit Version 9.0

You cannot migrate the WebSphere Message Broker Toolkit from previous versions, but you can install IBM Integration Toolkit Version 9.0 to coexist with a previous version, and migrate the development resources that you created in your workspace.

Before you begin

Before you start:

Be aware of the following restrictions that apply after you migrate resources to IBM Integration Toolkit Version 9.0.

- You cannot share the development resources with previous versions of the WebSphere Message Broker Toolkit. When you create a new project or open an existing project in IBM Integration Toolkit Version 9.0, it is converted to a new format, which cannot be used in previous versions of the WebSphere Message

Broker Toolkit. You can take the following steps to manage development with different versions of the WebSphere Message Broker Toolkit.

- If you are using a version control system, create a new stream for use with the new version of your project.
- If you expect to continue development of a project for a broker at a version before Version 9.0, you must retain a WebSphere Message Broker Toolkit at the previous version.
- If you need to continue development of the same project for brokers on both Version 9.0 and a previous version, ensure you use the WebSphere Message Broker Toolkit from the previous version for all your development.
- You cannot deploy resources from IBM Integration Toolkit Version 9.0 to brokers at a previous version to Version 9.0.
- If any of the following actions cause an error in the project, the IBM Integration Toolkit flags the error, and you can use a Quick Fix to rectify the error.
 - Creating the metadata information for the user-defined node project
 - Correcting the plug-in identifier if it does not match the project name
 - Ensuring all the user-defined nodes are in the same category

For more information, see *Applying a Quick Fix to a task list error*. However, if you have different user-defined nodes that depend on different resources that have identical names, the broker archive (BAR) file compiler produces an error that indicates a naming conflict in the dependent resources.

- If you have a deployable JAR file (for a user-defined node) that was created in a version before Version 9.0, you cannot migrate this file to Version 9.0.

About this task

You can continue to use resources from a previous version of WebSphere Message Broker by importing them into an IBM Integration Toolkit Version 9.0 workspace. After you import resources, you can no longer use them in a previous version of the WebSphere Message Broker Toolkit.

Message flow projects are replaced by integration projects in IBM Integration Bus Version 9.0. When you import message flow projects into IBM Integration Bus Version 9.0, they are converted automatically to integration projects. You cannot create a message flow project in IBM Integration Toolkit Version 9.0.

To migrate your WebSphere Message Broker Toolkit workspace or import a Project Interchange file, follow the appropriate set of steps.

- “Migrating your WebSphere Message Broker Toolkit workspace”
- “Importing resources in a Project Interchange file” on page 184

Migrating your WebSphere Message Broker Toolkit workspace

About this task

To migrate your WebSphere Message Broker Toolkit workspace to IBM Integration Bus Version 9.0, complete the following steps.

Procedure

1. Install IBM Integration Toolkit Version 9.0 in a different location from the current version of WebSphere Message Broker Toolkit. For detailed instructions, see “Installing the IBM Integration Toolkit” on page 79, or refer to the “Installation Guide” on page 32.

2. When you start IBM Integration Toolkit Version 9.0 for the first time, you are prompted to enter a workspace location. Enter the directory that contains the WebSphere Message Broker Toolkit workspace that you want to migrate and click **OK**. The workspace and resources are now available in IBM Integration Toolkit Version 9.0.
3. In the Integration Development perspective, select **Window > Reset Perspective** to ensure that all views and menus are updated for IBM Integration Toolkit Version 9.0.
4. You must migrate any user-defined node source projects that were developed in a previous version. An error is displayed on all projects that require migration. You can use a Quick Fix to clear these errors. See Applying a Quick Fix to a task list error.

Importing resources in a Project Interchange file

About this task

To import resources in a Project Interchange file, complete the following steps.

Procedure

1. Export your resources from the previous version of the WebSphere Message Broker Toolkit in a Project Interchange file.
2. Import the Project Interchange file into the Version 9.0 IBM Integration Toolkit by following the instructions in “Importing resources from previous versions” on page 185.
3. Ensure that your message flow projects have been converted to integration projects.

Some resources that were contained in message flow projects might cause errors in integration projects. Therefore, after you import message flow projects into IBM Integration Bus Version 9.0, check the converted integration project for error messages. For more information, see “Migrating message flow projects” on page 186.

Results

Results:

You can now view and modify existing resources, and create new resources. You can deploy your workspace resources to IBM Integration Bus Version 9.0 integration nodes (brokers).

After migration, your resources are contained in a integration project. You can then decide how you want to organize your resources. Applications are typically used to contain all the resources that are required for a particular solution. Libraries are typically used to contain resources that you might want to reuse. So you might decide to store your main message flow in an application, and store your reusable resources in a library. For an example of how you might convert your resources, see Example: Manual conversion of Version 7.0 resources to applications and libraries. For instructions about converting your migrated resources to applications and libraries, see Converting existing projects to applications and libraries.

What to do next

Next:

Migrate your WebSphere Message Broker component; for more information, see:

- “Migrating the Version 7.0 WebSphere Message Broker component” on page 206
- “Migrating the Version 6.1 WebSphere Message Broker component” on page 225

Related concepts:



IBM Integration Toolkit

The IBM Integration Toolkit is an integrated development environment and graphical user interface based on the Eclipse platform.

Related tasks:

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

Importing resources from previous versions

You can work with resources from previous versions of WebSphere Message Broker by using a project interchange file to import the resources into IBM Integration Bus Version 9.0.

About this task

You might have resources from previous versions of WebSphere Message Broker, such as Message Flow projects, Java projects, message set projects, and PHP projects, that you want to use in IBM Integration Bus Version 9.0. You can import these resources by migrating your WebSphere Message Broker Toolkit workspace, or by using a project interchange file.

Message flow projects are replaced by Message Broker projects in WebSphere Message Broker Version 8.0. Message Broker projects are replaced by integration projects in IBM Integration Bus Version 9.0. When you import Message Flow projects or Message Broker projects into IBM Integration Toolkit Version 9.0, they are converted automatically to integration projects. You can then use an imported project as the basis for a new application or library.

To import resources into IBM Integration Bus Version 9.0 by using a project interchange file, complete the following steps.

Procedure

1. In the previous version of WebSphere Message Broker Toolkit, export the resources that you want to use in IBM Integration Toolkit Version 9.0.
 - a. Click **File > Export**. The Export wizard opens.
 - b. Expand **Other**, click **Project Interchange**, then click **Next**.
 - c. Select the projects that you want to export, and specify the location for the compressed file that is created. You can save the file to a folder on your file system, or to a disk drive.
 - d. Click **Finish**. The project interchange file is created in the specified location.
2. Create an IBM Integration Toolkit Version 9.0 workspace, or open an existing one.
3. Import the project interchange file into the Version 9.0 workspace.
 - a. Click **File > Import**. The Import wizard opens.
 - b. Expand **Other**, click **Project Interchange**, then click **Next**.
 - c. Specify the location of the project interchange file that you created in step 1.
 - d. Specify the location of the open Version 9.0 workspace.

- e. Select the projects that you want to import into your Version 9.0 workspace, then click **Finish**.

Results

If you are importing Message Flow projects or Message Broker projects into IBM Integration Toolkit Version 9.0, they are converted automatically to integration projects.

What to do next

Next:

You can choose to convert the imported projects to applications or libraries. For an example of how you might convert your resources, see Example: Manual conversion of Version 7.0 resources to applications and libraries. For detailed instructions for how to convert your resources, see [Converting existing projects to applications and libraries](#).

Related concepts:

Integration projects

An integration project is a specialized container in which you create and maintain all the resources associated with one or more message flows.

Applications and libraries

Applications and libraries are deployable containers of resources, such as message flows, subflows, message definitions (DFDL, XSD files), JAR files, XSL style sheets, and WebSphere Adapters files.

Related tasks:

Managing message flow resources

You can use applications, libraries, and integration projects to organize your resources.

Creating an application

Create an application to contain all the resources that are required to develop your solution.

Creating a library

Create a library to contain a logical grouping of related code, data, or both, that can be reused.

Creating resources in an application

You can create resources such as message flows, maps, and ESQL files in an application.

“Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182

You cannot migrate the WebSphere Message Broker Toolkit from previous versions, but you can install IBM Integration Toolkit Version 9.0 to coexist with a previous version, and migrate the development resources that you created in your workspace.

Migrating message flow projects

When you import resources into an IBM Integration Toolkit Version 9.0 workspace, your message flow projects are converted automatically to integration projects.

Before you begin

Before you start:

Ensure that you imported resources from a previous version of WebSphere Message Broker, as described in “Importing resources from previous versions” on page 185.

About this task

You cannot create a message flow project in IBM Integration Bus Version 9.0. You might want to continue to use message flow projects if you are working in a team environment, for example.

After your message flow projects are converted to integration projects, the message flows behave in the same way as before. However, if your message flow project contains schema, adapter components, WSDL files, IDL files, or SCA definitions, you might see errors after you migrate your project to IBM Integration Bus Version 9.0.

Quick fixes are available for these errors. These resources have no effect in a message flow project; you can include them in a Message flow project for reference only. But when you include the resources in an application or library, the resources become visible to other applications and libraries. This visibility can affect the resolution of those resources in the workspace. For example, a Message flow project might contain an XSD file that defines an element that is named *element1*, and a message set also contains an XSD file that defines an element that is named *element1*. In previous versions of WebSphere Message Broker, this duplication does not cause a problem because the element in the Message flow project is not used for name resolution. However, after migration to IBM Integration Bus Version 9.0, the presence of two elements that are named *element1* causes an error.

An error is also displayed if you import a BAR file with resources from a previous version, and you then choose to refactor those resources to applications and libraries. If you try to recompile a BAR file after the resources are migrated to applications and libraries, you see an error message similar to the following example:

```
TotalPurchaseOrderFlow.msgflow belongs in an application or library and should be deployed within that container and not independently.
Create a new BAR file and select the application or library in the Prepare tab of the BAR editor, then select Build and Save.
To deploy the resource separately from the application or library, it must be moved into a Message Broker project.
```

Therefore, you cannot rebuild the BAR file but you can view the contents to see, for example, your original resources and how they were refactored.

What to do next

Next:

Decide how you want to organize your migrated resources. Applications are typically used to contain all the resources that are required for a particular solution. Libraries are typically used to contain resources that you might want to reuse. So you might decide to store your main message flow in an application, and store your reusable resources in a library. For an example of how you can convert resources to applications and libraries, see Example: Manual conversion of Version 7.0 resources to applications and libraries. For detailed instructions to convert your resources, see Converting existing projects to applications and libraries.

Related concepts:

Applications and libraries

Applications and libraries are deployable containers of resources, such as message flows, subflows, message definitions (DFDL, XSD files), JAR files, XSL style sheets, and WebSphere Adapters files.

Integration projects

An integration project is a specialized container in which you create and maintain all the resources associated with one or more message flows.

Related tasks:

Managing message flow resources

You can use applications, libraries, and integration projects to organize your resources.

“Importing resources from previous versions” on page 185

You can work with resources from previous versions of WebSphere Message Broker by using a project interchange file to import the resources into IBM Integration Bus Version 9.0.

Converting a project to an application or library

Two options are available to convert existing projects to applications or libraries.

Use **Convert Single Project** to convert a project with no references. Use **Analyse and Convert Multiple Projects** to convert one or more projects and any referenced resources.

Adding and removing library references

To include a library in an application or integration project, you can add a reference to that library to the application or integration project. You can also add a reference to a library from another library.

Applying a Quick Fix to a task list error

During the creation, migration and manipulation of message models, warnings or errors might occur; these are listed in the Problems view of the Integration Development perspective. Some of these warnings or errors can be cleared by applying a Quick Fix.

Migrating the Version 8.0 WebSphere Message Broker component

Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

Before you begin

Before you start:

Read “Preparing for migration” on page 172.

About this task

You can migrate to IBM Integration Bus Version 9.0 from WebSphere Message Broker Version 8.0 or later.

For full product version and release levels, see “Supported migration paths” on page 177.

For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

The instructions in the following section apply to all operating systems that are supported by WebSphere Message Broker Version 8.0.

Complete the following migration tasks in the order that is shown to reduce the possibility of problems.

Procedure

1. Back up your components and resources to ensure that you can return to your previous version if necessary. For more information, see “Backing up WebSphere Message Broker Version 8.0 resources” on page 190.

Consider backing up the following resources:

- Critical databases that are accessed by your message flows.
- All your development resources; for example, message flows and message sets.

If you use a repository to manage these resources, check that the product you use provides sufficient features for you to recover your resources at a specific version.

2. Migrate your WebSphere MQ installation to a supported version.
3. Migrate the IBM Integration Toolkit resource by following the instructions in “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.

If the users of your toolkit operate in a team environment and share resources with each other, upgrade all users to Version 9.0 at the same time to ensure continued access to all resources. Because toolkit resources are stored in a different format when they are first saved in Version 9.0, users might experience compatibility problems if they are still working with a previous version and some of their colleagues are using Version 9.0.

4. Migrate the broker by following the instructions in “Performing in-place broker migration for a Version 8.0 broker” on page 193 or “Performing parallel broker migration for a Version 8.0 broker” on page 201.
5. Optional: If you are migrating from a secure domain, you cannot directly migrate security settings for brokers. You must set up equivalent security by using the facilities that are provided by Version 9.0. To set up administration security for a broker, complete the following tasks.
 - a. Activate broker administration security
 - b. Set up your security based on existing ACLs
6. Optional: If you are migrating a broker that is configured by using Microsoft Cluster Services, follow the instructions in “Migrating a WebSphere Message Broker Version 8.0 broker that is configured on Microsoft Cluster Services (MSCS)” on page 200.
7. Optional: If you are migrating a multi-instance broker, follow the instructions in “Migrating a multi-instance broker from WebSphere Message Broker Version 8.0” on page 198.
8. Start the broker by using the **mqsistart** command.
9. Review the changes of behavior that are introduced in Version 9.0; for more details, see “Reviewing technical changes in IBM Integration Bus Version 9.0” on page 253.

What to do next

Next:

When you have completed migration, consider the list of tasks in “Perform post-migration tasks” on page 252, and follow the guidance that is provided if these tasks apply to your environment.

Related concepts:

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Preparing for migration” on page 172

Plan the order and extent of the migration of components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.

mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.

Backing up WebSphere Message Broker Version 8.0 resources

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.

Before you begin

Before you start:

To plan your migration strategy, read “Preparing for migration” on page 172.

About this task

Before you complete migration tasks, back up your WebSphere Message Broker Version 8.0 resources by completing the following steps.

Procedure

1. Optional: If your message flows access user databases through an ODBC connection, back up the ODBC files that you use for these connections. Take a copy of these files and store them safely in a different location.
2. Optional: If you have created or configured configurable services for your message flows, record the properties of all your configurable services. For example, a message flow might access databases through a JDBC connection, for which you have set up a JDBCProvider configurable service. Run the **mqsireportproperties** command for these services, take a copy of the output, and store it safely in a different location.

3. Back up your IBM Integration Toolkit workspace and resources; for example, message flow files, message set definition files, Java files, ESQL files, mapping files, XML Schema files, and broker archive (BAR) files.
 - Export all your projects from your Version 8.0 IBM Integration Toolkit.
 - Archive your workspace resources:
 - If you manage your workspace resources in a shared repository, for example CVS, follow standard backup procedures for safeguarding versions. Create a version for storing Version 9.0 resources.
 - If you maintain your workspace resources on a local or shared disk, copy your workspace directory to a different location.

What to do next

Next:

After you have backed up your WebSphere Message Broker Version 8.0 resources, update your ODBC definitions by following the instructions in “Updating ODBC definitions when migrating from WebSphere Message Broker Version 8.0.”

Related tasks:

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188
Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

Updating ODBC definitions when migrating from WebSphere Message Broker Version 8.0

Before you migrate a broker, create ODBC definitions for databases that specify appropriate database drivers for IBM Integration Bus Version 9.0.

About this task

The database drivers that are supported by IBM Integration Bus Version 9.0 may be at a later version than the drivers used by WebSphere Message Broker Version 8.0.

Complete this update before you run the `mqsिमigratecomponents` command for the broker that uses these ODBC connections.

Follow the instructions provided for your operating system:

Windows systems

To change the ODBC connection definitions:

1. Open the ODBC Data Source Administrator window.
2. Open the System DSN page.
3. For each Oracle and Sybase database that is accessed by the broker, compare the ODBC driver against the entries listed in the following table. If the ODBC driver does not match, you need to associate the data source name with the new ODBC driver using the following instructions:
 - a. Delete the data source by clicking **Remove**.
 - b. Re-create the data source with the new ODBC driver by clicking **Add**.

The following table displays the name of the new ODBC driver for each database management system (DBMS).

DBMS	IBM Integration Bus Version 9.0 ODBC driver name
Oracle	IBM Integration Bus DataDirect Technologies 7.0 32-BIT Oracle Wire Protocol
Sybase	IBM Integration Bus DataDirect Technologies 7.0 32-BIT Sybase Wire Protocol

To change the XA resource manager definitions:

1. Open the **Properties** window of the broker queue manager using the WebSphere MQ Services snap-in.
2. Open the **Resources** page.
3. For each Oracle and Sybase database that participates in a global unit of work, coordinated by the broker queue manager, change the contents of the **SwitchFile** field. For changes to the switch file configuration to take effect, you must restart the broker queue manager.

The following table specifies what you must change for each database management system (DBMS). *WBIMB* represents the fully qualified path name of the directory in which you have installed IBM Integration Bus.

DBMS	Change ...	To ...
Oracle	<i>WBIMB</i> \bin\ukor8dtc22.d11 or <i>WBIMB</i> \bin\ukor8dtc23.d11 or <i>WBIMB</i> \bin\ukora24.d11	<i>WBIMB</i> \bin\ukora26.d11
Sybase	<i>WBIMB</i> \bin\ukase22.d11 or <i>WBIMB</i> \bin\ukase23.d11 or <i>WBIMB</i> \bin\ukase24.d11	<i>WBIMB</i> \bin\ukase26.d11

Linux and UNIX systems

To change the ODBC connection definitions:

Create an ODBC definitions file by following the instructions in “Connecting to a database from Linux and UNIX systems by using the IBM Integration ODBC Database Extender” on page 375. Before you run the commands at the new service level, check that your ODBCINI environment variable points to the new file and not to the existing file. Check that ODBCYSINI environment variable is set to point to the directory that contains your odbcinstr.ini file.

To change the XA resource manager definitions:

To change the XA resource manager definitions, edit the queue manager configuration file (qm.ini) of the queue manager that is associated with the broker. The qm.ini file is located at /var/mqm/qmgrs/*queue_manager_name*/qm.ini, where *queue_manager_name* is the name of the queue manager that is associated with the broker.

In the *XAResourceManager* stanza for each Oracle and Sybase database that participates in a global unit of work that is coordinated by the broker queue manager, change the entry for the switch file. For changes to the switch file configuration to take effect, you must restart the broker queue manager.

The following table specifies what you must change for each broker operating system and database management system (DBMS).

DBMS	Change ...	To ...
Oracle	SwitchFile=UKor8dtc22.so orSwitchFile=UKoradtc22.so orSwitchFile=UKor8dtc23.so orSwitchFile=UKoradtc23.so orSwitchFile=UKoradtc24.so	SwitchFile=UKoradtc26.so
Sybase (not supported on Linux on System z)	SwitchFile=UKasedtc22.so orSwitchFile=UKasedtc23.so orSwitchFile=UKasedtc24.so	SwitchFile=UKasedtc26.so

To check that your ODBC environment is set up correctly on Linux and UNIX systems, run the **mqsicvp** command. This command also validates the connection to all data sources that are listed in the `odbc.ini` file that have been associated with a broker by using the **mqsisetdbparms** command. For more information, see **mqsicvp** command.

Results

If you revert to a previous version of IBM Integration Bus, you must reverse the changes that you make to the ODBC definitions. Update your ODBC files after you have run the **mqsimigratecomponents** command, but before you restart the broker at the earlier version.

Related tasks:

“Restoring components and resources to Version 8.0” on page 204

Restore components and resources that you migrated from Version 8.0 products to their original state.

“Enabling ODBC connections to the databases” on page 369

Set up the resources and environment that the broker requires for Open Database Connectivity (ODBC) connections to databases on distributed systems.

Performing in-place broker migration for a Version 8.0 broker

Migrate one or more brokers to IBM Integration Bus Version 9.0.

Before you begin

Before you start: Read “Preparing for migration” on page 172.

About this task

To migrate a broker from WebSphere Message Broker Version 8.0 to IBM Integration Bus Version 9.0, see the appropriate topic for your operating system and configuration:

- “Performing in-place broker migration for a Version 8.0 broker on distributed operating systems” on page 194
- “Performing in-place broker migration for a Version 8.0 broker on z/OS” on page 197

- Performing in-place broker migration for a Version 8.0 multi-instance broker
- Performing in-place broker migration for a Version 8.0 broker on Microsoft Cluster Services (MSCS)

Results

You do not have to redeploy resources to a broker that you have migrated. When you start the integration node (broker), it starts all existing message flows.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188

Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.

Performing in-place broker migration for a Version 8.0 broker on distributed operating systems:

Migrate a broker on a distributed operating system to use the enhanced facilities available in IBM Integration Bus Version 9.0 by using in-place migration.

Before you begin

Before you start:

- Back up your broker resources by following the instructions in “Backing up WebSphere Message Broker Version 8.0 resources” on page 190.
- For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.
- Install WebSphere MQ and migrate your queue managers and other resources, following the instructions in the WebSphere MQ information center.
- Migrate the IBM Integration Toolkit resources by following the instructions in “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.
- Check that you have no aggregations in progress on this broker. When you migrate a broker to Version 9.0, all live data that is being stored for aggregations in progress is lost.
- If the broker runs in a locale that is not listed in “Locales” on page 664, check that the code page is one of the supported code pages, and that the locale is set up correctly.

You cannot use in-place migration to migrate a broker from WebSphere Message Broker Version 8.0 on a Windows x86 operating system to IBM Integration Bus on a Windows x86-64 operating system. You must use parallel migration to migrate brokers between deployments on these operating systems, see “Parallel broker migration” on page 180

About this task

If you stop the broker, you can migrate it in-place immediately to the new version on the same computer. If you prefer not to stop the broker to avoid problems for your business applications, or if you want to reproduce the broker function on another computer, you can perform parallel broker migration instead, by associating the application logic on your Version 8.0 broker with a separate Version 9.0 broker. For more information, see “Performing parallel broker migration for a Version 8.0 broker” on page 201.

Migrating a Version 8.0 broker to Version 9.0:

About this task

To migrate a Version 8.0 broker on a distributed operating system to Version 9.0 on the same computer, complete the following steps:

Procedure

1. Install IBM Integration Bus Version 9.0 on the same computer as Version 8.0. Install at least the Integration Bus component; other components are optional. You must specify a new location for this installation.
2. Open the IBM Integration Toolkit and remove the broker from the domain configuration.
3. Stop all channels that are connected to the Version 8.0 broker.
4. Open a Version 8.0 command environment, and stop the Version 8.0 broker by using the **mqsistop** command.
5. Optional: If your message flows access user databases by using ODBC connections, update the ODBC definitions to Version 9.0 format by following the instructions in “Updating ODBC definitions when migrating from WebSphere Message Broker Version 8.0” on page 191.
6. Set up the correct Version 9.0 command environment:

- **Linux** **UNIX** On Linux and UNIX systems, open a new shell and run the environment profile **mqsiprofile** for this Version 9.0 installation.
- **Windows** On Windows, click **Start**, and open the Command Console that is associated with the Version 9.0 installation.

On Windows systems, you must open a command console with elevated privileges. To open a command console with elevated privileges, use the **mqsicommandconsole** command. For more information, see **mqsicommandconsole** command.

7. Run the **mqsimigratecomponents** command to migrate the broker. For example:

```
mqsimigratecomponents Broker1
```

All configuration data is retrieved from the Version 8.0 broker database. If you configured a default user ID and password for your Version 8.0 broker by using the **-u** and **-p** options on the **mqsicreatebroker** command, these values are migrated with the broker. You can change these values for your Version 9.0 broker by using the **mqsisetdbparms** command.

8. Copy all additional custom environment settings from your previous environment into your Version 9.0 environment; for example, `MQSI_FILENODES_ROOT_DIRECTORY`.
9. Start the Version 9.0 integration node (broker) by using the `mqsistart` command.

What to do next

Next:

When you have completed migration, see the tasks in “Perform post-migration tasks” on page 252 for information about tasks that you might want to complete after migration.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188
Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

“Backing up WebSphere Message Broker Version 8.0 resources” on page 190

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.



Starting and stopping a broker

Run the appropriate command to start or stop a broker.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.



`mqsicommandconsole` command

Use the `mqsicommandconsole` command to launch an elevated command console from which commands that require elevation on Windows can be run.



`mqsdeletebroker` command

Use the `mqsdeletebroker` command to delete a named broker. The command also deletes the queues on the associated queue manager (created when the broker was created). You can also specify that the queue manager is to be deleted.



`mqsigratecomponents` command

Use the `mqsigratecomponents` command to migrate a component from one version of the product to another version on the same computer.



`mqsistart` command

Use the `mqsistart` command to start the specified broker if all initial verification

tests complete successfully.



mqsisstop command

Use the **mqsisstop** command to stop the specified component.

Related information:



WebSphere MQ Version 7.5 product documentation

Performing in-place broker migration for a Version 8.0 broker on z/OS:

Migrate a broker on z/OS to use the enhanced facilities available in IBM Integration Bus Version 9.0 by using in-place migration.

Before you begin

Before you start:

- Back up your broker resources by following the instructions in “Backing up WebSphere Message Broker Version 8.0 resources” on page 190.
- For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.
- Install WebSphere MQ and migrate your queue managers and other resources, following the instructions that are provided in the WebSphere MQ information center.
- Ensure that you are familiar with the steps involved in creating a broker on z/OS; for more details, see “Creating a broker on z/OS” on page 313.
- Ensure that you are familiar with the **mqsimigratecomponents** command that the JCL (Job Control Language) file uses to migrate a broker.
- Check that you have no aggregations in progress. When you migrate a broker to Version 9.0, any live data that is being stored for aggregations in progress is lost.
- If the broker runs in a locale that is not listed in “Locales” on page 664, check that the code page is one of the supported code pages and that the locale is set up correctly.

About this task

To migrate a WebSphere Message Broker Version 8.0 broker to Version 9.0 on z/OS, complete the following steps.

Procedure

1. Open the WebSphere Message Broker Toolkit Version 8.0, and remove the broker from the domain configuration.
2. Stop the Version 8.0 broker.
3. Complete the steps that are defined in “Creating a broker on z/OS” on page 313 to create a new PDSE with customised Version 9.0 JCL scripts, but stop before you create the Integration Bus component.
4. Customize and submit the BIPMGCMP (**mqsimigratecomponents**) job. This job migrates the registry and queue.
5. Complete the steps that are defined in “Creating a broker on z/OS” on page 313 to replace the Version 8.0 Started Task JCL that is in the procedures library with the Version 9.0 Started Task JCL.
6. Start the Version 9.0 integration node (broker). The verification program checks the configuration of the broker.

What to do next

Next:

When you have completed migration, see “Perform post-migration tasks” on page 252 for information about tasks that you might want to perform after migration.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188

Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

“Backing up WebSphere Message Broker Version 8.0 resources” on page 190

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

“Creating the broker PDSE” on page 321

This is part of the larger task of creating a broker on z/OS.

“Customizing the broker JCL” on page 324

This subtask is part of the larger task of creating a broker on z/OS.

“Copying the broker started task to the procedures library” on page 328

This is part of the larger task of creating a broker on z/OS.



Starting and stopping a broker on z/OS

Run the appropriate command from SDSF to start or stop a broker.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.



mqsigratecomponents command

Use the **mqsigratecomponents** command to migrate a component from one version of the product to another version on the same computer.

Migrating a multi-instance broker from WebSphere Message Broker Version 8.0:

You must complete a specific set of steps to migrate your multi-instance broker to IBM Integration Bus Version 9.0.

Before you begin

Before you start:

Read the topics in the section “Migrating the Version 8.0 WebSphere Message Broker component” on page 188, and the topic “Migrating a WebSphere Message Broker Version 8.0 broker that is configured on Microsoft Cluster Services (MSCS)” on page 200.

About this task

Use the following procedure to migrate a multi-instance broker.

Migrating from WebSphere Message Broker Version 8.0 to IBM Integration Bus Version 9.0

Procedure

1. Install IBM Integration Bus Version 9.0 onto each node. You must specify a new location for this installation.
2. Stop all channels that are connected to the WebSphere Message Broker Version 8.0 broker.
3. Stop the broker on the standby node.
4. Stop the broker on the active node.
5. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1
```

where BROKER1 is the name of your broker.

6. Copy all additional custom environment settings from your previous environment into your IBM Integration Bus Version 9.0 environment; for example: MQSI_FILENODES_ROOT_DIRECTORY.
7. On the active node, in an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsistart BROKER1
```
8. On the standby node, in an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsistart BROKER1
```

What to do next

Migrating from IBM Integration Bus Version 9.0 to WebSphere Message Broker Version 8.0

1. Stop the broker on the standby node.
2. Stop the broker on the active node.
3. On the active node, in an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1 -s 9.0 -t 8.0
```

where BROKER1 is the name of your broker.

4. On the active node, in a WebSphere Message Broker Version 8.0 command console run the following command:

```
mqsistart BROKER1
```
5. On the standby node, in a WebSphere Message Broker Version 8.0 command console run the following command:

```
mqsistart BROKER1
```

Related tasks:

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188
Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

“Using a broker with an existing high availability manager” on page 571
You can use WebSphere Message Broker Version 8.0 with an existing high availability manager, for example HACMP™, HA/XD, VCS, or HP-UX Serviceguard.

Migrating a WebSphere Message Broker Version 8.0 broker that is configured on Microsoft Cluster Services (MSCS):

You must complete a specific set of steps to migrate a broker in an MSCS cluster to IBM Integration Bus Version 9.0.

Before you begin**Before you start:**

Read the topics in the section “Migrating the Version 8.0 WebSphere Message Broker component” on page 188, and the topic “Using a broker with an existing high availability manager” on page 571.

About this task

Use the following procedure to migrate a broker that is configured on MSCS.

Migrating from WebSphere Message Broker Version 8.0 to IBM Integration Bus Version 9.0**Procedure**

1. Install IBM Integration Bus Version 9.0 onto each node. You must specify a new location for this installation.
2. Stop all channels that are connected to the WebSphere Message Broker Version 8.0 broker.
3. In MSCS bring the broker resource offline, keeping the cluster disk and WebSphere MQ resource online on your primary node
4. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1
```

where BROKER1 is the name of your broker
5. With the broker resource offline, move the broker cluster group to the secondary node.
6. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1 -1
```
7. Copy all additional custom environment settings from your previous environment into your IBM Integration Bus Version 9.0 environment; for example: MQSI_FILENODES_ROOT_DIRECTORY.
8. Bring the cluster disk and WebSphere MQ resource offline.
9. Start the integration node (broker) cluster group on its primary node.

What to do next

Migrating from IBM Integration Bus Version 9.0 to WebSphere Message Broker Version 8.0

1. In MSCS bring the broker resource offline, keeping the cluster disk and WebSphere MQ resource online on your primary node.
2. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1 -s 9.0 -t 8.0
```
3. In a WebSphere Message Broker Version 8.0 command console run the following command:

```
mqsistart BROKER1
```

This step is required to update the Windows service to point at the WebSphere Message Broker Version 8.0 binary files.
4. In a WebSphere Message Broker Version 8.0 command console run the following command:

```
mqsistop BROKER1
```
5. With the broker resource offline, move the broker cluster group to the secondary node.
6. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1 -s 8 -t 9 -1
```
7. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1 -s 9.0 -t 8.0 -1
```
8. In a WebSphere Message Broker Version 8.0 command console run the following command:

```
mqsistart BROKER1
```

This step is required to update the Windows service to point at the WebSphere Message Broker Version 8.0 binary files
9. In a WebSphere Message Broker Version 8.0 command console run the following command:

```
mqsistop BROKER1
```
10. Bring the cluster disk and WebSphere MQ resource offline.
11. Start the broker cluster group on its primary node.

Related tasks:

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188
Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

“Using a broker with an existing high availability manager” on page 571
You can use WebSphere Message Broker Version 8.0 with an existing high availability manager, for example HACMP, HA/XD, VCS, or HP-UX Serviceguard.

Performing parallel broker migration for a Version 8.0 broker

Parallel broker migration allows a staged migration process by creating a new Version 9.0 broker to run in parallel with your existing Version 8.0 broker, to which you can migrate your application logic.

Before you begin

Before you start:

For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

- Backup your broker resources by following the instructions in “Backing up WebSphere Message Broker Version 8.0 resources” on page 190.
- Install a supported version of WebSphere MQ and migrate your queue managers and other resources, following the instructions that are provided in the WebSphere MQ information center.
- Migrate the IBM Integration Toolkit by following the instructions in “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.
- Check that you have no aggregations in progress on this broker. When you migrate a broker to Version 9.0, all live data that is being stored for aggregations in progress is lost.
- If the broker runs in a locale that is not listed in “Locales” on page 664, check that the code page is one of the supported code pages, and that the locale is set up correctly.

Migrating application logic to a Version 9.0 broker:

About this task

To migrate application logic from a Version 8.0 broker on a distributed operating system to a different Version 9.0 broker on the same computer, or on a different computer, complete the following steps:

Procedure

1. Install IBM Integration Bus Version 9.0 on the same computer as Version 8.0, or on a different computer. Install at least the Integration Bus component; other components are optional. If you are installing on the same computer, you must specify a different location.
2. Optional: If your message flows access databases by using ODBC connections, update the ODBC definitions to Version 9.0 format by following the instructions in “Updating ODBC definitions when migrating from WebSphere Message Broker Version 8.0” on page 191.
3. Set up the correct Version 9.0 command environment:
 - **Linux** **UNIX** On Linux and UNIX systems, open a new shell and run the environment profile **mqsiprofile** for this Version 9.0 installation.
 - **Windows** On Windows, click **Start**, and open the Command Console that is associated with this Version 9.0 installation.
On Windows systems, you must open a command console with elevated privileges. To open a command console with elevated privileges, use the **mqsicommandconsole** command. For more information, see **mqsicommandconsole** command.
4. Create a Version 9.0 broker by using the **mqsicreatebroker** command; give it a name and a queue manager name that are different from the names of the Version 8.0 broker and queue manager. If you have installed one or both of these components on this computer, you can also create a local broker by using the IBM Integration Explorer or the IBM Integration Toolkit.

5. Start the Version 9.0 broker by using the **mqsistart** command. If you have installed one or both of these components on this computer, you can also start a local broker by using the IBM Integration Explorer or the IBM Integration Toolkit.
6. Write a list of the integration servers that you have on the Version 8.0 broker, and create these same integration servers on the Version 9.0 broker. Use the IBM Integration Explorer Version 9.0, the IBM Integration Toolkit Version 9.0, or the **mqsicreateexecutiongroup** command to complete this step. On the AIX, Linux on x86-64, and Solaris on SPARC platforms, you can create only 64-bit integration servers.
7. Deploy the message flows and message sets that are in use by the Version 8.0 broker to the Version 9.0 broker from the Version 9.0 IBM Integration Toolkit. You cannot complete this step unless you have already migrated the IBM Integration Toolkit resources, as described in “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.
8. Configure all other relevant properties of the Version 8.0 broker on the Version 9.0 broker.
9. If you want to delete your Version 8.0 broker:
 - a. In a Version 8.0 command environment, stop the Version 8.0 broker by using the **mqsistop** command.
 - b. Remove the Version 8.0 broker from the Version 8.0 WebSphere Message Broker Toolkit.
 - c. In a Version 8.0 command environment, delete the Version 8.0 broker by using the **mqsdeletebroker** command.

What to do next

Next:

When you have completed migration, see the tasks in “Perform post-migration tasks” on page 252 for information about tasks that you might want to complete after migration.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188

Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

“Backing up WebSphere Message Broker Version 8.0 resources” on page 190

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.

“Creating a broker” on page 304

You can create brokers on every platform that is supported by IBM Integration Bus. The broker runs as a 64-bit application on all platforms except Linux on x86

and Windows on x86.



Starting and stopping a broker

Run the appropriate command to start or stop a broker.



Creating an integration server using the IBM Integration Toolkit or IBM Integration Explorer

Use the IBM Integration Toolkit or IBM Integration Explorer to create integration servers on your broker.



Creating an integration server using the **mqsicreateexecutiongroup** command
Use the **mqsicreateexecutiongroup** command to create integration servers on your broker.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.



mqsicommandconsole command

Use the **mqsicommandconsole** command to launch an elevated command console from which commands that require elevation on Windows can be run.



mqsdeletebroker command

Use the **mqsdeletebroker** command to delete a named broker. The command also deletes the queues on the associated queue manager (created when the broker was created). You can also specify that the queue manager is to be deleted.



mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.



mqsistop command

Use the **mqsistop** command to stop the specified component.

Restoring components and resources to Version 8.0

Restore components and resources that you migrated from Version 8.0 products to their original state.

About this task

If you migrated from WebSphere Message Broker Version 8.0 to IBM Integration Bus Version 9.0, you can restore your installation to Version 8.0, subject to the following restrictions and procedures.

When you restore brokers to Version 8.0, changes that you made that are compatible with Version 8.0 are kept. Changes that you made that are not compatible with Version 8.0 are not reflected in your broker's state, which might cause errors.

You might want to restore components if you encounter a problem after you migrate your brokers to Version 9.0. For example, if a message flow does not work as expected in Version 9.0, restore that broker until you can resolve the problem.

You can restore brokers to a previous version only if they were migrated from that version originally. You cannot create a new Version 9.0 broker, then restore it to a previous level.

When you use the **mqsigratecomponents** command, you must be logged in to a Version 9.0 command environment.

See the **mqsigratecomponents** command description for usage information and details of parameters and the format to use when you specify version numbers.

Restrictions

Message flows that you deploy to a broker after you migrate to Version 9.0 might no longer work when you restore that broker to Version 8.0; always check logs to search for errors or warnings about message flows that the broker cannot start.

Source files in the IBM Integration Toolkit Version 9.0 are maintained in a different format from that used in Version 8.0. The files are migrated to the new format when you save them in the IBM Integration Toolkit Version 9.0. After you save a resource with the IBM Integration Toolkit Version 9.0, you can no longer use the resources with the WebSphere Message Broker Toolkit Version 8.0.

Migrating resources back to Version 8.0

The following sections describe how to restore the IBM Integration Toolkit and your brokers to Version 8.0. You can reuse your original Version 8.0 Configuration Manager and User Name Server components with the brokers that you restore to Version 8.0.

Restoring the IBM Integration Toolkit Version 9.0 to WebSphere Message Broker Toolkit Version 8.0

1. Close all IBM Integration Toolkit Version 9.0 sessions.
2. Restore the Version 8.0 workspace from the backup that you took before migration.

Ensure that the workspace directories include the `.metadata` directory, which contains information that is specific to the release of Eclipse on which the toolkit is based.

3. Restart WebSphere Message Broker Toolkit Version 8.0.

Restoring brokers to Version 8.0

Use the **-s** and **-t** parameters of the **mqsigratecomponents** command to migrate brokers from Version 9.0 to Version 8.0.

- Specify the installed level of Version 9.0 for the source version parameter (**-s**), for example `9.0.0.0` for the GA (general availability) level, or allow this to take the default value.
- Specify the appropriate level of Version 8.0 for the target version parameter (**-t**), for example `8.0.0.2` for Version 8.0, Fix Pack 2.

See the **mqsigratecomponents** command for detailed information about these parameters and the format to use when you specify version numbers.

Restoring brokers on distributed systems

1. Stop the Version 9.0 broker by using the **mqsistop** command.
2. Restore the broker to Version 8.0 by using the **mqsigratecomponents** command, as shown in the following example:

```
mqsigratecomponents Broker -t 8.0.0.2
```

3. Reverse the changes that you made to the ODBC definitions when you migrated to Version 9.0; for more information, see “Updating ODBC definitions when migrating from WebSphere Message Broker Version 8.0” on page 191.
4. Open a Version 8.0 command window, and restart the broker by using the **mqsistart** command.

If you migrate to Version 9.0, deploy a message set to the Version 9.0 broker, and then migrate back to Version 8.0, Version 8.0 is unable to recognize the message set that was deployed by Version 9.0. In this case, all message sets that Version 8.0 is unable to use are deleted and the warning message BIP8688W is displayed for each message set, prompting you to redeploy it to Version 8.0 following successful migration.

Restoring brokers on z/OS

1. Stop the Version 9.0 broker by using one of the available options.
2. Submit the BIPMGCOMP job to call the **mqsimigratecomponents** command, specifying the **-s** and **-t** parameters, as described for distributed systems.
3. Restore your Version 8.0 ENVFILE from a backup that you have made, or by resubmitting the Version 8.0 BIPGEN job.
4. Replace the started task JCL file in USER.PROCLIB with the Version 8.0 copy that you backed up.

Related tasks:

Chapter 3, “Migrating,” on page 171

To migrate a broker domain to IBM Integration Bus Version 9.0, plan your migration strategy, perform pre-migration tasks, migrate your domain components, then complete post-migration tasks.

“Backing up WebSphere Message Broker Version 8.0 resources” on page 190

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

Related reference:



mqsimigratecomponents command

Use the **mqsimigratecomponents** command to migrate a component from one version of the product to another version on the same computer.

Migrating the Version 7.0 WebSphere Message Broker component

Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

Before you begin

Before you start:

Read “Preparing for migration” on page 172.

About this task

You can migrate to IBM Integration Bus Version 9.0 from the following products:

- WebSphere Message Broker Version 7.0
- WebSphere Message Broker with Rules and Formatter Extension Version 7.0

For full product version and release levels, see “Supported migration paths” on page 177.

For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

The instructions in this section apply to all operating systems that are supported by Version 7.0.

Complete the following migration tasks in the order that is shown to reduce the possibility of problems.

Procedure

1. Back up your components and resources to ensure that you can return to your previous version if necessary. For more information and instructions, see the information center for your current product.

Consider backing up the following resources:

- Critical databases that are accessed by your message flows.
- All your development resources; for example, message flows and message sets.

If you use a repository to manage these resources, check that the product you use provides sufficient features for you to recover your resources at a specific version.

2. Migrate your WebSphere MQ installation to a supported version.
3. Migrate the IBM Integration Toolkit resource by following the instructions in “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.

If the users of your toolkit operate in a team environment and share resources with each other, upgrade all users to IBM Integration Toolkit Version 9.0 at the same time to ensure continued access to all resources. Because toolkit resources are stored in a different format when they are first saved in Version 9.0, your users might experience compatibility problems in some circumstances if some of their colleagues are using Version 9.0 and they are still working with a previous version.

4. Migrate the broker by following the instructions in “Performing in-place broker migration for a Version 7.0 broker” on page 212.
5. Optional: If you are migrating from a secure domain, you cannot directly migrate security settings for brokers. You must set up equivalent security by using the facilities that are provided by IBM Integration Bus Version 9.0. To set up administration security for a broker, complete the following tasks.
 - a. Activate broker administration security

- b. Set up your security based on existing ACLs
6. Optional: If you are migrating a broker that is configured to use Microsoft Cluster Services, follow the instructions in “Migrating a WebSphere Message Broker Version 7.0 broker that is configured on Microsoft Cluster Services (MSCS)” on page 218.
7. Optional: If you are migrating a multi-instance broker, follow the instructions in “Migrating a multi-instance broker from WebSphere Message Broker Version 7.0” on page 217.
8. Start the broker by using the **mqsistart** command.
9. Review the changes of behavior that are introduced in IBM Integration Bus Version 9.0; for more details, see “Reviewing technical changes in IBM Integration Bus Version 9.0” on page 253.

What to do next

When you have completed migration, consider the list of tasks in “Perform post-migration tasks” on page 252, and follow the guidance that is provided if these tasks apply to your environment.

Related concepts:

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Preparing for migration” on page 172

Plan the order and extent of the migration of components and resources to IBM Integration Bus Version 9.0.

“Migrating publish/subscribe information to WebSphere MQ” on page 244

If you have publish/subscribe applications and you want to preserve the publish/subscribe state of these applications when you migrate, you must move the durable subscriptions, retained publications and access control lists out of WebSphere Message Broker Version 6.1 and into your broker's queue manager during migration.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.



mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.

Backing up WebSphere Message Broker Version 7.0 resources

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.

Before you begin

Before you start:

To plan your migration strategy, read “Preparing for migration” on page 172.

About this task

Before you complete migration tasks, back up your WebSphere Message Broker Version 7.0 resources by completing the following steps.

Procedure

1. Optional: If your message flows access user databases through an ODBC connection, back up the ODBC files that you use for these connections. Take a copy of these files and store them safely in a different location.
2. Optional: If you created or configured configurable services for your message flows, record the properties of all your configurable services. For example, a message flow might access databases through a JDBC connection, for which you configured a JDBCProvider configurable service. Run the **mqsireportproperties** command for these services, take a copy of the output, and store it safely in a different location.
3. Back up your IBM Integration Toolkit workspace and resources; for example, message flow files, message set definition files, Java files, ESQL files, mapping files, XML Schema files, and broker archive (BAR) files.
 - Export all your projects from your Version 7.0 IBM Integration Toolkit.
 - Archive your workspace resources:
 - If you manage your workspace resources in a shared repository, for example CVS, follow standard backup procedures for safeguarding versions. Create a version for storing Version 9.0 resources.
 - If you maintain your workspace resources on a local or shared disk, copy your workspace directory to a different location.

What to do next

Next:

After you back up your WebSphere Message Broker Version 7.0 resources, update your ODBC definitions by following the instructions in “Updating ODBC definitions when migrating.”

Related tasks:

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206
Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

Updating ODBC definitions when migrating

Before you migrate a broker, create ODBC definitions for databases that specify appropriate database drivers for IBM Integration Bus Version 9.0.

About this task

The database drivers that are supported by IBM Integration Bus Version 9.0 might be at a later version than the drivers used by WebSphere Message Broker Version 7.0 or WebSphere Message Broker Version 6.1.

Complete this update before you run the **mqsिमigratecomponents** command for the broker that uses these ODBC connections.

Follow the instructions that are provided for your operating system:

Windows systems

To change the ODBC connection definitions:

1. Open the ODBC Data Source Administrator window.
2. Open the System DSN page.
3. For each Oracle and Sybase database that is accessed by the broker, compare the ODBC driver against the entries that are listed in the following table. If the ODBC driver does not match, you must associate the data source name with the new ODBC driver by using the following instructions:
 - a. Delete the data source by clicking **Remove**.
 - b. Re-create the data source with the new ODBC driver by clicking **Add**.

The following table displays the name of the new ODBC driver for each database management system (DBMS).

DBMS	IBM Integration Bus Version 9.0 ODBC driver name
Oracle	IBM Integration Bus DataDirect Technologies 7.0 32-BIT Oracle Wire Protocol
Sybase	IBM Integration Bus DataDirect Technologies 7.0 32-BIT Sybase Wire Protocol

To change the XA resource manager definitions:

1. Open the **Properties** window of the broker queue manager by using the WebSphere MQ Services snap-in.
2. Open the **Resources** page.
3. For each Oracle and Sybase database that participates in a global unit of work that is coordinated by the broker queue manager, change the contents of the **SwitchFile** field. For changes to the switch file configuration to take effect, you must restart the broker queue manager.

The following table specifies what you must change for each database management system (DBMS). *WBIMB* represents the fully qualified path name of the directory in which you installed WebSphere Message Broker.

DBMS	Change ...	To ...
Oracle	<i>WBIMB</i> \bin\ukor8dtc22.d11 or <i>WBIMB</i> \bin\ukor8dtc23.d11 or <i>WBIMB</i> \bin\ukora24.d11	<i>WBIMB</i> \bin\ukora26.d11
Sybase	<i>WBIMB</i> \bin\ukase22.d11 or <i>WBIMB</i> \bin\ukase23.d11 or <i>WBIMB</i> \bin\ukase24.d11	<i>WBIMB</i> \bin\ukase26.d11

Linux and UNIX systems

To change the ODBC connection definitions:

Create an ODBC definitions file by following the instructions in “Connecting to a database from Linux and UNIX systems by using the IBM Integration ODBC Database Extender” on page 375. Before you run the commands at the new service level, check that your ODBCINI environment variable points to the new file and not to the existing file. Check that ODBCYSINI environment variable is set to point to the directory that contains your `odbcinst.ini` file.

To change the XA resource manager definitions:

To change the XA resource manager definitions, edit the queue manager configuration file (qm.ini) of the queue manager that is associated with the broker. The qm.ini file is located at `/var/mqm/qmgrs/queue_manager_name/qm.ini`, where *queue_manager_name* is the name of the queue manager that is associated with the broker.

In the *XAResourceManager* stanza for each Oracle and Sybase database that participates in a global unit of work that is coordinated by the broker queue manager, change the entry for the switch file. For changes to the switch file configuration to take effect, you must restart the broker queue manager.

The following table specifies what you must change for each broker operating system and database management system (DBMS).

DBMS	Change ...	To ...
Oracle	SwitchFile=UKor8dtc22.so orSwitchFile=UKoradtc22.so orSwitchFile=UKor8dtc23.so orSwitchFile=UKoradtc23.so orSwitchFile=UKoradtc24.so	SwitchFile=UKoradtc26.so
Sybase (not supported on Linux on System z)	SwitchFile=UKasedtc22.so orSwitchFile=UKasedtc23.so orSwitchFile=UKasedtc24.so	SwitchFile=UKasedtc26.so

To check that your ODBC environment is set up correctly on Linux and UNIX systems, run the **mqsicvp** command. This command also validates the connection to all data sources that are listed in the `odbc.ini` file that are associated with a broker by using the **mqsisetdbparms** command. For more information, see **mqsicvp** command.

Results

If you revert to a previous version of IBM Integration Bus, you must reverse the changes that you make to the ODBC definitions. Update your ODBC files after you run the **mqsigratecomponents** command, but before you restart the broker at the earlier version.

Related tasks:

“Restoring components and resources to Version 8.0” on page 204

Restore components and resources that you migrated from Version 8.0 products to their original state.

“Restoring components and resources to Version 7.0” on page 222

Restore components and resources that you migrated from Version 7.0 products to their original state.

“Restoring components and resources to Version 6.1” on page 250

Restore components and resources that you migrated from Version 6.1 products to their original state.

“Enabling ODBC connections to the databases” on page 369

Set up the resources and environment that the broker requires for Open Database Connectivity (ODBC) connections to databases on distributed systems.

Performing in-place broker migration for a Version 7.0 broker

Migrate one or more brokers to IBM Integration Bus Version 9.0.

Before you begin

Before you start: Read “Preparing for migration” on page 172.

About this task

To migrate a broker from WebSphere Message Broker Version 7.0 to IBM Integration Bus Version 9.0, see the appropriate topic for your operating system:

- “Performing in-place broker migration for a Version 7.0 broker on distributed operating systems”
- “Performing in-place broker migration for a Version 7.0 broker on z/OS” on page 215
- Performing in-place broker migration for a Version 7.0 multi-instance broker
- Performing in-place broker migration for a Version 7.0 broker on Microsoft Cluster Services (MSCS)

Results

You do not have to redeploy resources to a broker that you have migrated. When you start the broker, it starts all existing message flows.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206
Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.

Performing in-place broker migration for a Version 7.0 broker on distributed operating systems:

Migrate a broker to use the enhanced facilities available in IBM Integration Bus Version 9.0.

Before you begin

Before you start:

- Back up your broker resources by following the instructions in “Backing up WebSphere Message Broker Version 7.0 resources” on page 208.
- For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.
- Install WebSphere MQ and migrate your queue managers and other resources, following the instructions that are provided by WebSphere MQ.
- Migrate the IBM Integration Toolkit resources by following the instructions in “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.
- Check that you have no aggregations in progress on this broker. When you migrate a broker to Version 9.0, all live data that is being stored for aggregations in progress is lost.
- If the broker runs in a locale that is not listed in “Locales” on page 664, check that the code page is one of the supported code pages, and that the locale is set up correctly.

You cannot use in-place migration to migrate a broker from WebSphere Message Broker Version 7.0 on a Windows x86 operating system to IBM Integration Bus on a Windows x86-64 operating system. You must use parallel migration to migrate brokers between deployments on these operating systems, see “Parallel broker migration” on page 180

About this task

If you stop the broker, you can migrate it in-place immediately to the new version on the same computer. If you prefer not to stop the broker to avoid problems for your business applications, or if you want to reproduce the broker function on another computer, you can instead perform parallel broker migration by associating the application logic on your Version 7.0 broker with a separate Version 9.0 broker. For more information, see “Performing parallel broker migration for a Version 7.0 broker” on page 220.

Migrating a Version 7.0 broker to Version 9.0:

About this task

To migrate a Version 7.0 broker on a distributed operating system to Version 9.0 on the same computer, complete the following steps:

Procedure

1. Install IBM Integration Bus Version 9.0 on the same computer as Version 7.0. Install at least the Integration Bus component; other components are optional. You must specify a new location for this installation.
2. Stop all channels that are connected to the Version 7.0 broker.
3. Open a Version 9.0 command environment, and stop the Version 7.0 broker by using the **mqsistop** command.
4. Optional: If your message flows access user databases by using ODBC connections, update the ODBC definitions to Version 9.0 format by following the instructions in “Updating ODBC definitions when migrating” on page 209.
5. Set up the correct Version 9.0 command environment:

- **Linux** **UNIX** On Linux and UNIX systems, open a new shell and run the environment profile **mqsiprofile** for this Version 9.0 installation.
- **Windows** On Windows, click **Start**, and open the Command Console that is associated with this Version 9.0 installation.

On Windows systems, you must open a command console with elevated privileges. To open a command console with elevated privileges, use the **mqsicommandconsole** command. For more information, see **mqsicommandconsole** command.

6. Run the **mqsimigratecomponents** command to migrate the broker. For example:
mqsimigratecomponents Broker1
All configuration data is retrieved from the Version 7.0 broker database. If you configured a default user ID and password for your Version 7.0 broker by using the **-u** and **-p** options on the **mqsicreatebroker** command, these values are migrated with the broker. You can change these values for your Version 9.0 broker by using the **mqsisetdbparms** command.
7. Copy all additional custom environment settings from your previous environment into your Version 9.0 environment; for example, **MQSI_FILENODES_ROOT_DIRECTORY**.
8. Start the Version 9.0 broker by using the **mqsistart** command.

What to do next

Next:

When you have completed migration, see the tasks in “Perform post-migration tasks” on page 252 for information about tasks that you might want to complete after migration.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206
Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

“Backing up WebSphere Message Broker Version 7.0 resources” on page 208

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.



Starting and stopping a broker

Run the appropriate command to start or stop a broker.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.

mqsicommandconsole command

Use the **mqsicommandconsole** command to launch an elevated command console from which commands that require elevation on Windows can be run.

mqsdeletebroker command

Use the **mqsdeletebroker** command to delete a named broker. The command also deletes the queues on the associated queue manager (created when the broker was created). You can also specify that the queue manager is to be deleted.

mqsigratecomponents command

Use the **mqsigratecomponents** command to migrate a component from one version of the product to another version on the same computer.

mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.

mqsistop command

Use the **mqsistop** command to stop the specified component.

Performing in-place broker migration for a Version 7.0 broker on z/OS:

Migrate a broker to use the enhanced facilities available in IBM Integration Bus Version 9.0.

Before you begin

Before you start:

- Back up your broker resources by following the instructions in “Backing up WebSphere Message Broker Version 7.0 resources” on page 208.
- For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.
- Install WebSphere MQ and migrate your queue managers and other resources, following the instructions that are provided in the WebSphere MQ information center.
- Ensure that you are familiar with the steps involved in creating a broker on z/OS; for more details, see “Creating a broker on z/OS” on page 313.
- Ensure that you are familiar with the **mqsigratecomponents** command that the JCL (Job Control Language) file uses to migrate a broker.
- Check that you have no aggregations in progress. When you migrate a broker to Version 9.0, any live data that is being stored for aggregations in progress is lost.
- If the broker runs in a locale that is not listed in “Locales” on page 664, check that the code page is one of the supported code pages and that the locale is set up correctly.

About this task

To migrate a WebSphere Message Broker Version 7.0 broker to Version 9.0 on z/OS, complete the following steps.

Procedure

1. Open the WebSphere Message Broker Toolkit Version 7.0, and remove the broker from the domain configuration.

2. Stop the Version 7.0 broker.
3. Complete the steps that are defined in “Creating a broker on z/OS” on page 313 to create a new PDSE with customised Version 9.0 JCL scripts, but stop before you create the Integration Bus component.
4. Customize and submit the BIPMGCMP (**mqsigratecomponents**) job. This job migrates the registry and queue.
5. Complete the steps that are defined in “Creating a broker on z/OS” on page 313 to replace the Version 7.0 Started Task JCL that is in the procedures library with the Version 9.0 Started Task JCL.
6. Start the Version 9.0 integration node (broker). The verification program checks the configuration of the broker.

What to do next

Next:

When you have completed migration, see “Perform post-migration tasks” on page 252 for information about tasks that you might want to perform after migration.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206
Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

“Backing up WebSphere Message Broker Version 7.0 resources” on page 208

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

“Creating the broker PDSE” on page 321

This is part of the larger task of creating a broker on z/OS.

“Customizing the broker JCL” on page 324

This subtask is part of the larger task of creating a broker on z/OS.

“Copying the broker started task to the procedures library” on page 328

This is part of the larger task of creating a broker on z/OS.



Starting and stopping a broker on z/OS

Run the appropriate command from SDSF to start or stop a broker.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.



mqsigratecomponents command

Use the **mqsigratecomponents** command to migrate a component from one version of the product to another version on the same computer.

Migrating a multi-instance broker from WebSphere Message Broker Version 7.0:

You must complete a specific set of steps to migrate your multi-instance broker to IBM Integration Bus Version 9.0.

Before you begin

Before you start:

Read the topics in the section “Migrating the Version 7.0 WebSphere Message Broker component” on page 206, and the topic “Migrating a WebSphere Message Broker Version 7.0 broker that is configured on Microsoft Cluster Services (MSCS)” on page 218.

About this task

Use the following procedure to migrate a multi-instance broker.

Migrating from WebSphere Message Broker Version 7.0 to IBM Integration Bus Version 9.0

Procedure

1. Install IBM Integration Bus Version 9.0 onto each node. You must specify a new location for this installation.
2. Stop all channels that are connected to the WebSphere Message Broker Version 7.0 broker.
3. Stop the broker on the standby node.
4. Stop the broker on the active node.
5. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1
```

where BROKER1 is the name of your broker.

6. Copy all additional custom environment settings from your previous environment into your IBM Integration Bus Version 9.0 environment; for example: MQSI_FILENODES_ROOT_DIRECTORY.
7. On the active node, in an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsistart BROKER1
```
8. On the standby node, in an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsistart BROKER1
```

What to do next

Migrating from IBM Integration Bus Version 9.0 to WebSphere Message Broker Version 7.0

1. Stop the broker on the standby node.
2. Stop the broker on the active node.
3. On the active node, in an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsimigratecomponents BROKER1 -s 9.0 -t 7.0
```

where BROKER1 is the name of your broker.

4. On the active node, in a WebSphere Message Broker Version 7.0 command console run the following command:
5. On the standby node, in a WebSphere Message Broker Version 7.0 command console run the following command:

```
mqsisstart BROKER1
```

```
mqsisstart BROKER1
```

Related tasks:

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206
Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

“Using a broker with an existing high availability manager” on page 571
You can use WebSphere Message Broker Version 8.0 with an existing high availability manager, for example HACMP, HA/XD, VCS, or HP-UX Serviceguard.

Migrating a WebSphere Message Broker Version 7.0 broker that is configured on Microsoft Cluster Services (MSCS):

You must complete a specific set of steps to migrate a broker in an MSCS cluster to IBM Integration Bus Version 9.0.

Before you begin

Before you start:

Read the topics in the section “Migrating the Version 7.0 WebSphere Message Broker component” on page 206, and the topic “Using a broker with an existing high availability manager” on page 571.

About this task

Use the following procedure to migrate a broker that is configured on MSCS.

Migrating from WebSphere Message Broker Version 7.0 to IBM Integration Bus Version 9.0

Procedure

1. Install IBM Integration Bus Version 9.0 onto each node. You must specify a new location for this installation.
2. Stop all channels that are connected to the WebSphere Message Broker Version 7.0 broker.
3. In MSCS bring the broker resource offline, keeping the cluster disk and WebSphere MQ resource online on your primary node
4. In an IBM Integration Bus Version 9.0 command console run the following command: .

```
mqsigratecomponents BROKER1
```

where BROKER1 is the name of your broker

5. With the broker resource offline, move the broker cluster group to the secondary node.
6. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1 -1
```
7. Copy all additional custom environment settings from your previous environment into your IBM Integration Bus Version 9.0 environment; for example: MQSI_FILENODES_ROOT_DIRECTORY.
8. Bring the cluster disk and WebSphere MQ resource offline.
9. Start the broker cluster group on its primary node.

What to do next

Migrating from IBM Integration Bus Version 9.0 to WebSphere Message Broker Version 7.0

1. In MSCS bring the broker resource offline, keeping the cluster disk and WebSphere MQ resource online on your primary node.
2. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1 -s 9.0 -t 7.0
```
3. In a WebSphere Message Broker Version 7.0 command console run the following command:

```
mqsistart BROKER1
```

This step is required to update the Windows service to point at the WebSphere Message Broker Version 7.0 binary files.
4. In a WebSphere Message Broker Version 7.0 command console run the following command:

```
mqsistop BROKER1
```
5. With the broker resource offline, move the broker cluster group to the secondary node.
6. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1 -s 7 -t 9 -1
```
7. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1 -s 9.0 -t 7.0 -1
```
8. In a WebSphere Message Broker Version 7.0 command console run the following command:

```
mqsistart BROKER1
```

This is required to update the Windows service to point at the WebSphere Message Broker Version 7.0 binary files
9. In a WebSphere Message Broker Version 7.0 command console run the following command:

```
mqsistop BROKER1
```
10. Bring the cluster disk and WebSphere MQ resource offline.
11. Start the broker cluster group on its primary node.

Note, that, because of changes in the file security model that is used in IBM Integration Bus Version 9.0, you might need to alter the file permissions of the shared resource.

If necessary, create a domain group, add the userID of your broker to this group, and give the group full access to the shared files.

Related tasks:

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206
Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

“Using a broker with an existing high availability manager” on page 571
You can use WebSphere Message Broker Version 8.0 with an existing high availability manager, for example HACMP, HA/XD, VCS, or HP-UX Serviceguard.

Performing parallel broker migration for a Version 7.0 broker

Parallel broker migration allows a staged migration process by creating a new Version 9.0 broker to run in parallel with your existing Version 7.0 broker, to which you can migrate your application logic.

Before you begin

Before you start:

For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

- Backup your broker resources by following the instructions in “Backing up WebSphere Message Broker Version 7.0 resources” on page 208.
- Install a supported version of WebSphere MQ and migrate your queue managers and other resources, following the instructions that are provided in the WebSphere MQ information center.
- Migrate the WebSphere Message Broker Toolkit by following the instructions in “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.
- Check that you have no aggregations in progress on this broker. When you migrate a broker to Version 9.0, all live data that is being stored for aggregations in progress is lost.
- If the broker runs in a locale that is not listed in “Locales” on page 664, check that the code page is one of the supported code pages, and that the locale is set up correctly.

Migrating application logic to a Version 9.0 broker:

About this task

To migrate application logic from a Version 7.0 broker on a distributed operating system to a different Version 9.0 broker on the same computer, or on a different computer, complete the following steps:

Procedure

1. Install IBM Integration Bus Version 9.0 on the same computer as Version 7.0, or on a different computer. Install at least the Integration Bus component; other components are optional. If you are installing on the same computer, you must specify a different location.

2. Optional: If your message flows access databases by using ODBC connections, update the ODBC definitions to Version 9.0 format by following the instructions in “Updating ODBC definitions when migrating” on page 209.
3. Set up the correct Version 9.0 command environment:
 - **Linux** **UNIX** On Linux and UNIX systems, open a new shell and run the environment profile **mqsiprofile** for this Version 9.0 installation.
 - **Windows** On Windows, click **Start**, and open the Command Console that is associated with this Version 9.0 installation.
 On Windows systems, you must open a command console with elevated privileges. To open a command console with elevated privileges, use the **mqsicommandconsole** command. For more information, see **mqsicommandconsole** command.
4. Create a Version 9.0 broker by using the **mqsicreatebroker** command; give it a name and a queue manager name that are different from the names of the Version 7.0 broker and queue manager. If you installed one or both of these components on this computer, you can also create a local broker by using the IBM Integration Explorer or the IBM Integration Toolkit.
5. Start the Version 9.0 broker by using the **mqsistart** command. If you installed one or both of these components on this computer, you can also start a local broker by using the IBM Integration Explorer or the IBM Integration Toolkit.
6. Write a list of the integration servers that you have on the Version 7.0 broker, and create these same integration servers on the Version 9.0 broker. Use the IBM Integration Explorer, the IBM Integration Toolkit Version 9.0, or the **mqsicreateexecutiongroup** command to complete this step. On the AIX, Linux on x86-64, and Solaris on SPARC platforms, you can create only 64-bit integration servers.
7. Deploy the message flows and message sets that are in use by the Version 7.0 broker to the Version 9.0 broker from the IBM Integration Toolkit Version 9.0. You cannot complete this step unless you migrated the IBM Integration Toolkit resources, as described in “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.
8. Configure all other relevant properties of the Version 7.0 broker on the Version 9.0 broker.
9. If you want to delete your Version 7.0 broker, complete the following steps:
 - a. In a Version 7.0 command environment, stop the Version 7.0 broker by using the **mqsistop** command.
 - b. Remove the Version 7.0 broker from the Version 7.0 IBM Integration Toolkit.
 - c. In a Version 7.0 command environment, delete the Version 7.0 broker by using the **mqsideletebroker** command.

What to do next

When you have completed migration, see the tasks in “Perform post-migration tasks” on page 252 for information about tasks that you might want to complete after migration.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous

versions, with some restrictions.

Related tasks:

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206
Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

“Backing up WebSphere Message Broker Version 7.0 resources” on page 208

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.

“Creating a broker” on page 304

You can create brokers on every platform that is supported by IBM Integration Bus. The broker runs as a 64-bit application on all platforms except Linux on x86 and Windows on x86.



Starting and stopping a broker

Run the appropriate command to start or stop a broker.



Creating an integration server using the IBM Integration Toolkit or IBM Integration Explorer

Use the IBM Integration Toolkit or IBM Integration Explorer to create integration servers on your broker.



Creating an integration server using the **mqsicreateexecutiongroup** command

Use the **mqsicreateexecutiongroup** command to create integration servers on your broker.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.



mqsicommandconsole command

Use the **mqsicommandconsole** command to launch an elevated command console from which commands that require elevation on Windows can be run.



mqsdeletebroker command

Use the **mqsdeletebroker** command to delete a named broker. The command also deletes the queues on the associated queue manager (created when the broker was created). You can also specify that the queue manager is to be deleted.



mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.



mqsistop command

Use the **mqsistop** command to stop the specified component.

Restoring components and resources to Version 7.0

Restore components and resources that you migrated from Version 7.0 products to their original state.

About this task

If you migrated from WebSphere Message Broker Version 7.0 to IBM Integration Bus Version 9.0, you can restore your installation to Version 7.0, subject to the following restrictions and procedures.

When you restore brokers to Version 7.0, changes that you made that are compatible with Version 7.0 are kept. Changes that you made that are not compatible with Version 7.0 are not reflected in your broker's state, which might cause errors.

You might want to restore components if you encounter a problem after you migrate your brokers to Version 9.0. For example, if a message flow does not work as expected in Version 9.0, you can restore that broker until you can resolve the problem.

You can restore brokers to a previous version only if they were migrated from that version originally. You cannot create a new Version 9.0 broker, then restore it to a previous level.

When you use the `mqsigratecomponents` command, you must be logged in to a Version 9.0 command environment.

See the `mqsigratecomponents` command description for usage information and details of parameters and the format to use when you specify version numbers.

Restrictions

Message flows that you deployed to a broker after migration to Version 9.0 might no longer work when you restore that broker to Version 7.0; always check logs to search for errors or warnings about message flows that the broker cannot start.

Source files in the IBM Integration Toolkit Version 9.0 are maintained in a different format from the format used in Version 7.0. The files are migrated to the new format when you save them in the IBM Integration Toolkit Version 9.0. After you save the files, you can no longer use the resources with the IBM Integration Toolkit Version 7.0.

In Version 9.0, you can write CMP applications to modify message flows. You can use message flow files from Version 7.0 in these applications. When you save message flow files in a Version 8.0 CMP application, these files are stored in Version 9.0 format and cannot be used with earlier versions of the IBM Integration Toolkit.

Migrating resources back to Version 7.0

The following sections describe how to restore the WebSphere Message Broker Toolkit and your brokers to Version 7.0. You can reuse your original Version 7.0 Configuration Manager and User Name Server components with the brokers that you restore to Version 7.0.

Restoring the IBM Integration Toolkit to Version 7.0

1. Close all IBM Integration Toolkit Version 9.0 sessions.
2. Restore the Version 7.0 workspace from the backup that you took before migration.

Ensure that the workspace directories include the `.metadata` directory, which contains information that is specific to the release of Eclipse on which the toolkit is based.

3. Restart WebSphere Message Broker Toolkit Version 7.0.

Restoring brokers to Version 7.0

Use the **-s** and **-t** parameters of the **mqsigratecomponents** command to migrate brokers from Version 9.0 to Version 7.0.

- Specify the installed level of Version 9.0 for the source version parameter (**-s**), for example 9.0.0.0 for the GA (general availability) level, or allow this to take the default value.
- Specify the appropriate level of Version 7.0 for the target version parameter (**-t**), for example 7.0.0.3 for Version 7.0, Fix Pack 3.

See the **mqsigratecomponents** command for detailed information about these parameters and the format to use when you specify version numbers.

Restoring brokers on distributed systems

1. Stop the Version 9.0 broker by using the **mqsistop** command.
2. Restore the broker to Version 7.0 by using the **mqsigratecomponents** command, as shown in the following example:
mqsigratecomponents Broker -t 7.0.0.3
3. Reverse the changes that you made to the ODBC definitions when you migrated to Version 9.0; for more information, see “Updating ODBC definitions when migrating” on page 209.
4. Open a Version 7.0 command window, and restart the broker by using the **mqsistart** command.

If you migrate to Version 9.0, deploy a message set to the Version 9.0 broker, and then migrate back to Version 7.0, Version 7.0 is unable to recognize the message set that was deployed by Version 9.0. In this case, all message sets that Version 7.0 is unable to use are deleted and the warning message BIP8688W is displayed for each message set, prompting you to redeploy it to Version 7.0 following successful migration.

Restoring brokers on z/OS

1. Stop the Version 9.0 broker by using one of the available options.
2. Submit the BIPMGCMP job to call the **mqsigratecomponents** command, specifying the **-s** and **-t** parameters, as described for distributed systems.
3. Restore your Version 7.0 ENVFILE from a backup, or by resubmitting the Version 7.0 BIPGEN job.
4. Replace the started task JCL file in USER.PROCLIB with the Version 7.0 copy that you backed up.

Related tasks:

Chapter 3, “Migrating,” on page 171

To migrate a broker domain to IBM Integration Bus Version 9.0, plan your migration strategy, perform pre-migration tasks, migrate your domain components, then complete post-migration tasks.

“Backing up WebSphere Message Broker Version 7.0 resources” on page 208

Back up your resources before you start to migrate components to IBM Integration

Bus Version 9.0.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

Related reference:



mqsigratecomponents command

Use the **mqsigratecomponents** command to migrate a component from one version of the product to another version on the same computer.

Migrating the Version 6.1 WebSphere Message Broker component

Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

Before you begin

Before you start:

Read “Preparing for migration” on page 172.

About this task

You can migrate to IBM Integration Bus Version 9.0 from the following products:

- WebSphere Message Broker Version 6.1
- WebSphere Message Broker with Rules and Formatter Extension Version 6.1

For full product version and release levels, see “Supported migration paths” on page 177.

For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

The architecture of IBM Integration Bus Version 9.0 is simplified:

- The broker has no requirement for a database. When you migrate from WebSphere Message Broker Version 6.1, the following changes apply:
 - Because a database is no longer a mandatory requirement for a broker, DB2 and DB2 Runtime Client (the Derby database supported only on Windows systems) are not supplied when you purchase IBM Integration Bus Version 9.0.
 - The **mqscreatebroker** and **mqschangebroker** commands are updated to remove the specification of database parameters.
 - The DatabaseInstanceMgr that was available on Windows systems to manage Derby and DB2 databases is removed from Version 9.0. Commands that control the DatabaseInstanceMgr and the Derby database are removed:
 - mqscreatedb**
 - mqsdeletedb**
 - mqschangedbimgr**
- The broker security model is implemented by using WebSphere MQ queues. The Configuration Manager component is removed and you cannot migrate existing components of this type. All operational data that is maintained by the Configuration Manager component is retrieved when you migrate your brokers.

- The commands that controlled the Configuration Manager component are removed:
 - mqsichangeconfigmgr**
 - mqsicreateconfigmgr**
 - mqsideleteconfigmgr**
 - mqsireportconfigmgr**
- All topic-based publish/subscribe operations are handled by WebSphere MQ. Because access control for publishers and subscribers is managed by WebSphere MQ, the User Name Server component is no longer required, and is removed; you cannot migrate existing components of this type. The following commands are no longer available:
 - mqsichangeusernameserver**
 - mqsicreateusernameserver**
 - mqsideleteusernameserver**
- In IBM Integration Bus Version 9.0, publish/subscribe is controlled by WebSphere MQ. If your applications use publish/subscribe functions, and you have message flows that include one or more Publication nodes, you must migrate your subscriptions to WebSphere MQ after you migrate your queue manager but before you migrate your broker. Information about how to run the **migmqbrk** command to migrate the subscriptions and perform other tasks that might be required, is provided in “Migrating publish/subscribe information to WebSphere MQ” on page 244. After migration, you must use WebSphere MQ facilities to change subscriptions.

If you are using cloned WebSphere Message Broker Version 6.1 support, you must change to use WebSphere MQ Clustering for Publish and Subscribe.

The instructions in this section apply to all operating systems that are supported by Version 6.1.

Complete the following migration tasks in the order that is shown to reduce the possibility of problems.

Procedure

1. Back up your components and resources to ensure that you can return to your previous version if necessary. For more information and instructions, see the information center for your current product.

Consider backing up the following resources:

- The internal configuration repository that is maintained by the Configuration Manager.
- The broker database.
- Other critical databases that are accessed by your message flows.
- All your development resources; for example, message flows and message sets.

If you use a repository to manage these resources, check that the product you use provides sufficient features for you to recover your resources at a specific version.

2. Install a version of WebSphere MQ that is supported by IBM Integration Bus Version 9.0 but do not attempt to migrate your existing WebSphere MQ installation. WebSphere Message Broker Version 6.1 cannot run on WebSphere MQ Version 7.5.0.1 for distributed platforms, or Version 7.1 for z/OS, which are the minimum supported versions that are required for IBM Integration Bus Version 9.0.

3. Migrate the IBM Integration Toolkit by following the instructions in “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.
If the users of your toolkit operate in a team environment and share resources with each other, upgrade all users to Version 9.0 at the same time to ensure continued access to all resources. Because toolkit resources are stored in a different format when they are first saved in Version 9.0, your users might experience compatibility problems in some circumstances if some of their colleagues are using Version 9.0 and they are still working with a previous version.
4. Migrate the broker to the new installation of WebSphere MQ and IBM Integration Bus Version 9.0 by following the instructions in “Performing in-place broker migration for a Version 6.1 broker” on page 231.
5. Optional: If you are migrating from a secure domain, you cannot directly migrate security settings for brokers. You must set up equivalent security by using the facilities that are provided by Version 9.0. To set up administration security for a broker, complete the following tasks.
 - a. Enabling administration security
 - b. “Migrating Configuration Manager ACLs” on page 242
6. Optional: If you are migrating a broker that is configured on Microsoft Cluster Services, follow the instructions in “Migrating a WebSphere Message Broker Version 6.1 broker that is configured on Microsoft Cluster Services (MSCS)” on page 238.
7. Start the broker by using the `mqsistart` command.
8. Review the technical changes that are introduced in Version 9.0; for more information, see “Reviewing technical changes in IBM Integration Bus Version 9.0” on page 253.

What to do next

When you have completed migration, consider the list of tasks in “Perform post-migration tasks” on page 252, and follow the guidance that is provided if these tasks apply to your environment.

Related concepts:

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Preparing for migration” on page 172

Plan the order and extent of the migration of components and resources to IBM Integration Bus Version 9.0.

“Migrating publish/subscribe information to WebSphere MQ” on page 244

If you have publish/subscribe applications and you want to preserve the publish/subscribe state of these applications when you migrate, you must move the durable subscriptions, retained publications and access control lists out of WebSphere Message Broker Version 6.1 and into your broker's queue manager during migration.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the

product.



mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.

Related information:



WebSphere Message Broker product documentation

Backing up WebSphere Message Broker Version 6.1 resources

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.

Before you begin

Before you start:

Read “Preparing for migration” on page 172 to plan your migration strategy.

About this task

Before you carry out migration tasks, back up your WebSphere Message Broker Version 6.1 resources by completing the following steps.

Note: It is not possible to revert an installation that was migrated to IBM Integration Bus Version 9.0 so you must back up everything that you require to rebuild your WebSphere Message Broker Version 6.1 environment from scratch.

Procedure

1. Back up the broker database tables. On z/OS, customize and submit the JCL member BIPBUDB. On other platforms, use the documented procedures that are provided by your database supplier.
2. Optional: If your message flows access user databases through an ODBC connection, back up the ODBC files that you use for these connections. Take a copy of these files and store them safely in a different location.
3. Optional: If you created or configured configurable services for your message flows, record the properties of all your configurable services. For example, a message flow might access databases through a JDBC connection, for which you configured a JDBCProvider configurable service. Run the **mqsireportproperties** command for these services, take a copy of the output, and store it safely in a different location.
4. Back up your WebSphere Message Broker Toolkit workspace and resources; for example, message flow files, message set definition files, Java files, ESQL files, mapping files, XML Schema files, and broker archive (BAR) files.
 - Export all your projects from your WebSphere Message Broker Toolkit Version 6.1.
 - Archive your workspace resources:
 - If you manage your workspace resources in a shared repository, for example CVS, follow standard backup procedures for safeguarding versions. Create a version for storing Version 9.0 resources.
 - If you maintain your workspace resources on a local or shared disk, copy your workspace directory to a different location.

Results

For detailed instructions on how to back up these resources, see the WebSphere Message Broker Version 6.1 information center.

What to do next

Next: After you back up your WebSphere Message Broker Version 6.1 resources, update your ODBC definitions by following the instructions in “Updating ODBC definitions when migrating” on page 209.

Related tasks:

“Migrating the Version 6.1 WebSphere Message Broker component” on page 225
Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

Related information:

 [WebSphere Message Broker product documentation](#)

Updating ODBC definitions when migrating

Before you migrate a broker, create ODBC definitions for databases that specify appropriate database drivers for IBM Integration Bus Version 9.0.

About this task

The database drivers that are supported by IBM Integration Bus Version 9.0 might be at a later version than the drivers used by WebSphere Message Broker Version 7.0 or WebSphere Message Broker Version 6.1.

Complete this update before you run the `mqsिमigratecomponents` command for the broker that uses these ODBC connections.

Follow the instructions that are provided for your operating system:

Windows systems

To change the ODBC connection definitions:

1. Open the ODBC Data Source Administrator window.
2. Open the System DSN page.
3. For each Oracle and Sybase database that is accessed by the broker, compare the ODBC driver against the entries that are listed in the following table. If the ODBC driver does not match, you must associate the data source name with the new ODBC driver by using the following instructions:
 - a. Delete the data source by clicking **Remove**.
 - b. Re-create the data source with the new ODBC driver by clicking **Add**.

The following table displays the name of the new ODBC driver for each database management system (DBMS).

DBMS	IBM Integration Bus Version 9.0 ODBC driver name
Oracle	IBM Integration Bus DataDirect Technologies 7.0 32-BIT Oracle Wire Protocol
Sybase	IBM Integration Bus DataDirect Technologies 7.0 32-BIT Sybase Wire Protocol

To change the XA resource manager definitions:

1. Open the **Properties** window of the broker queue manager by using the WebSphere MQ Services snap-in.
2. Open the **Resources** page.
3. For each Oracle and Sybase database that participates in a global unit of work that is coordinated by the broker queue manager, change the contents of the **SwitchFile** field. For changes to the switch file configuration to take effect, you must restart the broker queue manager.

The following table specifies what you must change for each database management system (DBMS). *WBIMB* represents the fully qualified path name of the directory in which you installed WebSphere Message Broker.

DBMS	Change ...	To ...
Oracle	<i>WBIMB</i> \bin\ukor8dtc22.dll or <i>WBIMB</i> \bin\ukor8dtc23.dll or <i>WBIMB</i> \bin\ukora24.dll	<i>WBIMB</i> \bin\ukora26.dll
Sybase	<i>WBIMB</i> \bin\ukase22.dll or <i>WBIMB</i> \bin\ukase23.dll or <i>WBIMB</i> \bin\ukase24.dll	<i>WBIMB</i> \bin\ukase26.dll

Linux and UNIX systems

To change the ODBC connection definitions:

Create an ODBC definitions file by following the instructions in “Connecting to a database from Linux and UNIX systems by using the IBM Integration ODBC Database Extender” on page 375. Before you run the commands at the new service level, check that your ODBCINI environment variable points to the new file and not to the existing file. Check that ODBCYSINI environment variable is set to point to the directory that contains your *odbcinst.ini* file.

To change the XA resource manager definitions:

To change the XA resource manager definitions, edit the queue manager configuration file (*qm.ini*) of the queue manager that is associated with the broker. The *qm.ini* file is located at */var/mqm/qmgrs/queue_manager_name/qm.ini*, where *queue_manager_name* is the name of the queue manager that is associated with the broker.

In the *XAResourceManager* stanza for each Oracle and Sybase database that participates in a global unit of work that is coordinated by the broker queue manager, change the entry for the switch file. For changes to the switch file configuration to take effect, you must restart the broker queue manager.

The following table specifies what you must change for each broker operating system and database management system (DBMS).

DBMS	Change ...	To ...
Oracle	SwitchFile=UKor8dtc22.so orSwitchFile=UKoradtc22.so orSwitchFile=UKor8dtc23.so orSwitchFile=UKoradtc23.so orSwitchFile=UKoradtc24.so	SwitchFile=UKoradtc26.so
Sybase (not supported on Linux on System z)	SwitchFile=UKasedtc22.so orSwitchFile=UKasedtc23.so orSwitchFile=UKasedtc24.so	SwitchFile=UKasedtc26.so

To check that your ODBC environment is set up correctly on Linux and UNIX systems, run the **mqsicvp** command. This command also validates the connection to all data sources that are listed in the `odbc.ini` file that are associated with a broker by using the **mqsisetdbparms** command. For more information, see **mqsicvp** command.

Results

If you revert to a previous version of IBM Integration Bus, you must reverse the changes that you make to the ODBC definitions. Update your ODBC files after you run the **mqsimigratecomponents** command, but before you restart the broker at the earlier version.

Related tasks:

“Restoring components and resources to Version 8.0” on page 204

Restore components and resources that you migrated from Version 8.0 products to their original state.

“Restoring components and resources to Version 7.0” on page 222

Restore components and resources that you migrated from Version 7.0 products to their original state.

“Restoring components and resources to Version 6.1” on page 250

Restore components and resources that you migrated from Version 6.1 products to their original state.

“Enabling ODBC connections to the databases” on page 369

Set up the resources and environment that the broker requires for Open Database Connectivity (ODBC) connections to databases on distributed systems.

Performing in-place broker migration for a Version 6.1 broker

Migrate one or more brokers to IBM Integration Bus Version 9.0.

Before you begin

Before you start:

Read “Preparing for migration” on page 172.

About this task

To migrate a broker from WebSphere Message Broker Version 6.1 to IBM Integration Bus Version 9.0, see the appropriate topic for your operating system:

- “Performing in-place broker migration for a Version 6.1 broker on distributed operating systems”
- “Performing in-place broker migration for a Version 6.1 broker on z/OS” on page 235
- Performing in-place broker migration for a Version 6.1 broker on Microsoft Cluster Services (MSCS)

Results

You do not have to redeploy resources to a broker that you migrated. When you start the broker, it starts all existing message flows.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Migrating the Version 6.1 WebSphere Message Broker component” on page 225
Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.

Performing in-place broker migration for a Version 6.1 broker on distributed operating systems:

Migrate a broker to use the enhanced facilities available in IBM Integration Bus Version 9.0.

Before you begin

Before you start:

- Back up your broker resources by following the instructions in “Backing up WebSphere Message Broker Version 6.1 resources” on page 228.
- For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.
- Install a version of WebSphere MQ that supports IBM Integration Bus Version 9.0 alongside your existing WebSphere MQ deployment. Do not migrate your broker's queue manager until you are ready to migrate your broker.
- Optional: If your applications use publish/subscribe functions, and you have message flows that include one or more Publication nodes, you must migrate your subscriptions to WebSphere MQ before you migrate the broker; for more information, see “Migrating publish/subscribe information to WebSphere MQ” on page 244.

- Migrate the IBM Integration Toolkit by following the instructions in “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.
- Check that you have no aggregations in progress on this broker. When you migrate a broker to IBM Integration Bus Version 9.0, all live data that is being stored for aggregations in progress is lost.
- If the broker runs in a locale that is not listed in “Locales” on page 664, check that the code page is one of the supported code pages, and that the locale is set up correctly.

You cannot use in-place migration to migrate a broker from WebSphere Message Broker Version 6.1 on a Windows x86 operating system to IBM Integration Bus on a Windows x86-64 operating system. You must use parallel migration to migrate brokers between deployments on these operating systems, see “Parallel broker migration” on page 180

About this task

If you stop the broker, you can migrate it immediately in-place to the new version on the same computer. If you prefer not to stop the broker to avoid problems for your business applications, or if you want to reproduce the broker function on another computer, you can instead associate the application logic on your Version 6.1 broker with a separate Version 9.0 broker. For more information, see “Performing parallel broker migration for a Version 6.1 broker” on page 239.

Migrating a Version 6.1 broker to Version 9.0:

About this task

To migrate a WebSphere Message Broker Version 6.1 broker to IBM Integration Bus Version 9.0 on the same computer on a distributed operating system, complete the following steps.

Procedure

1. Install IBM Integration Bus Version 9.0 on the same computer as WebSphere Message Broker Version 6.1. Install at least the Integration Bus component; other components are optional. You must specify a new location for this installation.
2. Open the WebSphere Message Broker Toolkit Version 6.1, and remove the broker from the domain configuration.
3. Stop all channels that are connected to the Version 6.1 broker.
4. Open a Version 6.1 command environment, and stop the Version 6.1 broker by using the **mqsisstop** command.
5. Migrate your broker's queue manager to a version of WebSphere MQ that supports IBM Integration Bus Version 9.0. For information about migrating queue managers, see "multi-stage migration" in the WebSphere MQ Version 7.5 Information Center online.

Note: Do not attempt to restart your Version 6.1 broker after you upgrade your queue manager because WebSphere Message Broker Version 6.1 brokers are not supported on WebSphere MQ 7.5.0.1 or later, which is the minimum version of WebSphere MQ supported by IBM Integration Bus Version 9.0.

6. Optional: If you have publish/subscribe applications and you want to preserve the publish/subscribe state of these applications, migrate the publish/subscribe information; see “Migrating publish/subscribe information to WebSphere MQ” on page 244.

7. Optional: If your message flows access user databases by using ODBC connections, update the ODBC definitions to Version 9.0 format; see “Updating ODBC definitions when migrating” on page 209.
8. Set up the correct Version 9.0 command environment:
 - **Linux** **UNIX** On Linux and UNIX systems, open a new shell and run the environment profile **mqsiprofile** for the Version 9.0 installation.
 - **Windows** On Windows, click **Start**, and open the Command Console that is associated with the Version 9.0 installation.
On Windows systems, you must open a command console with elevated privileges. To open a command console with elevated privileges, use the **mqsicommandconsole** command. For more information, see **mqsicommandconsole** command.

9. Run the **mqsimigratecomponents** command to migrate the broker. For example:
mqsimigratecomponents Broker1

All configuration data is retrieved from the Version 6.1 broker database. If you configured a default user ID and password for your Version 6.1 broker by using the **-u** and **-p** parameters on the **mqsicreatebroker** command, these values are migrated with the broker. You can change these values for your Version 9.0 broker by using the **mqsisetdbparms** command.

10. On AIX, Linux on x86-64, and Solaris on SPARC systems, support has changed from both 32-bit and 64-bit support to 64-bit support only. 32-bit integration servers are converted to 64-bit. These systems now require 64-bit libraries and 64-bit ODBC configurations for databases.

- a. Recompile C/C++ user-defined nodes, parsers, and exits to 64-bit, if the existing ones are 32-bit only.
- b. Configure the broker to point at these recompiled extensions by using the following command:

```
mqsichangebroker -l userLilPath -x userExitPath
```

where *userLilPath* defines one or more paths to your LIL files, and *userExitPath* defines one or more paths to your user exit programs.

The following changes are made to environment variables, for all systems:

Previous version	Version 9.0
MQSI_LILPATH64 and MQSI_LILPATH32	MQSI_LILPATH
MQSI_USER_EXIT_PATH64 and MQSI_USER_EXIT_PATH	MQSI_USER_EXIT_PATH
MQSI_SECURITY_PROVIDER_PATH64 and MQSI_SECURITY_PROVIDER_PATH32	MQSI_SECURITY_PROVIDER_PATH

Update all custom profiles that set these environment variables.

11. Copy all additional custom environment settings from your previous environment into your Version 9.0 environment; for example, MQSI_FILENODES_ROOT_DIRECTORY.
12. Start the Version 9.0 broker by using the **mqsistart** command.

What to do next

Next:

When you have completed the migration, see the tasks in “Perform post-migration tasks” on page 252 for information about tasks that you might want to perform after migration.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Migrating the Version 6.1 WebSphere Message Broker component” on page 225

Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

“Backing up WebSphere Message Broker Version 6.1 resources” on page 228

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.



Starting and stopping a broker

Run the appropriate command to start or stop a broker.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.



mqsicommandconsole command

Use the **mqsicommandconsole** command to launch an elevated command console from which commands that require elevation on Windows can be run.



mqsdeletebroker command

Use the **mqsdeletebroker** command to delete a named broker. The command also deletes the queues on the associated queue manager (created when the broker was created). You can also specify that the queue manager is to be deleted.



mqsigratecomponents command

Use the **mqsigratecomponents** command to migrate a component from one version of the product to another version on the same computer.



mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.



mqsistop command

Use the **mqsistop** command to stop the specified component.

Performing in-place broker migration for a Version 6.1 broker on z/OS:

Migrate a broker to use the enhanced facilities available in IBM Integration Bus Version 9.0.

Before you begin

Before you start:

- Back up your broker resources by following the instructions in “Backing up WebSphere Message Broker Version 6.1 resources” on page 228.
- For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.
- Install a version of WebSphere MQ that supports IBM Integration Bus Version 9.0 alongside your existing WebSphere MQ deployment. Do not migrate your broker's queue manager until you are ready to migrate your broker.
- Optional: If your applications use publish/subscribe functions, and you have message flows that include one or more Publication nodes, you must migrate your subscriptions to a supported version of WebSphere MQ before you migrate the broker; for more information, see “Migrating publish/subscribe information to WebSphere MQ” on page 244.
- Ensure that you are familiar with the steps involved in “Creating a broker on z/OS” on page 313.
- Ensure that you are familiar with the **mqsigratecomponents** command, which the JCL (Job Control Language) file uses to migrate a broker.
- Check that you have no aggregations in progress. When you migrate a broker to IBM Integration Bus Version 9.0, any live data that is being stored for aggregations in progress is lost.
- If the broker runs in a locale that is not listed in “Locales” on page 664, check that the code page is one of the supported code pages and that the locale is set up correctly.

About this task

To migrate a WebSphere Message Broker Version 6.1 broker to IBM Integration Bus Version 9.0 on z/OS, complete the following steps.

Procedure

1. Open the WebSphere Message Broker Toolkit Version 6.1, and remove the broker from the domain configuration.
2. Stop the Version 6.1 broker and the Version 6.1 Configuration Manager.
3. Migrate your broker's queue manager to a version of WebSphere MQ that supports IBM Integration Bus Version 9.0. For information about migrating queue managers, see WebSphere MQ Version 7.1 Information Center online

Note: Do not attempt to restart your Version 6.1 broker after you upgrade your queue manager because WebSphere Message Broker Version 6.1 on z/OS brokers are not supported on WebSphere MQ 7.1 or later, which is the minimum version of WebSphere MQ supported by IBM Integration Bus Version 9.0 on z/OS.

4. Optional: If you have publish/subscribe applications and you want to preserve the publish/subscribe state of these applications, migrate the publish/subscribe information; see “Migrating publish/subscribe information to WebSphere MQ” on page 244.
5. Complete the steps that are defined in “Creating a broker on z/OS” on page 313 to create a new PDSE with customised Version 9.0 JCL scripts, but stop before you create the Integration Bus component.
6. Customize and submit the BIPMGCMP (**mqsigratecomponents**) job. This job migrates the registry, queue, and broker database; connection to the broker database of the previous broker is required to complete this action.

After migration, all information in the broker database is stored in an internal repository in Version 9.0; the Version 9.0 integration node (broker) does not require a database.

7. Complete the steps that are defined in “Creating a broker on z/OS” on page 313 to replace the Version 6.1 Started Task JCL that is in the procedures library with the Version 9.0 Started Task JCL.
8. Start the Version 9.0 integration node (broker). The verification program checks the configuration of the broker.

What to do next

Next:

When you have completed migration, see “Perform post-migration tasks” on page 252 for information about tasks that you might want to perform after migration.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Migrating the Version 6.1 WebSphere Message Broker component” on page 225
Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

“Backing up WebSphere Message Broker Version 6.1 resources” on page 228

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

“Creating the broker PDSE” on page 321

This is part of the larger task of creating a broker on z/OS.

“Customizing the broker JCL” on page 324

This subtask is part of the larger task of creating a broker on z/OS.

“Copying the broker started task to the procedures library” on page 328

This is part of the larger task of creating a broker on z/OS.



Starting and stopping a broker on z/OS

Run the appropriate command from SDSF to start or stop a broker.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.



mqsimigratecomponents command

Use the **mqsimigratecomponents** command to migrate a component from one version of the product to another version on the same computer.

Migrating a WebSphere Message Broker Version 6.1 broker that is configured on Microsoft Cluster Services (MSCS):

You must complete a specific set of steps to migrate a broker on an MSCS cluster to IBM Integration Bus Version 9.0.

Before you begin

Before you start:

- Read the topics in the section “Migrating the Version 6.1 WebSphere Message Broker component” on page 225, and the topic “Using a broker with an existing high availability manager” on page 571.
- Install a version of WebSphere MQ that supports IBM Integration Bus Version 9.0 alongside your existing WebSphere MQ deployment. Do not migrate your broker's queue manager until you are ready to migrate your broker.

About this task

Use the following procedure to migrate a broker that is configured on MSCS from WebSphere Message Broker Version 6.1 to IBM Integration Bus Version 9.0.

Procedure

1. Install IBM Integration Bus Version 9.0 on the same computer as WebSphere Message Broker Version 6.1. Install at least the Integration Bus component; other components are optional. You must specify a new location for this installation.
2. Open the IBM Integration Toolkit Version 6.1, and remove the broker from the domain configuration.
3. Stop all channels that are connected to the Version 6.1 broker.
4. In MSCS bring the broker resource and WebSphere MQ offline, keeping the cluster disk online on your primary node
5. Migrate your broker's queue manager to a version of WebSphere MQ that supports IBM Integration Bus Version 9.0. For information about migrating queue managers, see "multi-stage migration" in the WebSphere MQ Version 7.5 Information Center online.

Note: Do not attempt to restart your Version 6.1 broker after you upgrade your queue manager because WebSphere Message Broker Version 6.1 brokers are not supported on WebSphere MQ 7.5.0.1 or later, which is the minimum version of WebSphere MQ supported by IBM Integration Bus Version 9.0.

6. Optional: If you have publish/subscribe applications and you want to preserve the publish/subscribe state of these applications, migrate the publish/subscribe information; see “Migrating publish/subscribe information to WebSphere MQ” on page 244.
7. Optional: If your message flows access user databases by using ODBC connections, update the ODBC definitions to Version 9.0 format; see “Updating ODBC definitions when migrating” on page 209.
8. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsimigratecomponents BROKER1 -m
```

where BROKER1 is the name of your broker

9. With the broker resource offline, move the broker cluster group to the secondary node.
10. In an IBM Integration Bus Version 9.0 command console run the following command:

```
mqsigratecomponents BROKER1 -m -1
```
11. Copy all additional custom environment settings from your previous environment into your IBM Integration Bus Version 9.0 environment; for example: MQSI_FILENODES_ROOT_DIRECTORY.
12. Bring the cluster disk and WebSphere MQ resource offline.
13. Start the broker cluster group on its primary node.

Example

What to do next

Next:

Due to changes in the file security model used in IBM Integration Bus Version 9.0, you might need to alter the file permissions of the shared resource.

If necessary, create a domain group, add the userID of your broker to this group, and give the group full access to the shared files.

When you have completed the migration, see the tasks in “Perform post-migration tasks” on page 252 for information about tasks that you might want to perform after migration.

Related tasks:

“Migrating the Version 6.1 WebSphere Message Broker component” on page 225
Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

“Using a broker with an existing high availability manager” on page 571
You can use WebSphere Message Broker Version 8.0 with an existing high availability manager, for example HACMP, HA/XD, VCS, or HP-UX Serviceguard.

Performing parallel broker migration for a Version 6.1 broker

You can perform a parallel broker migration by creating a new IBM Integration Bus Version 9.0 integration node (broker) to run in parallel with your existing WebSphere Message Broker Version 6.1 broker, to which you can migrate your application logic.

Before you begin

Before you start:

- For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.
- Back up your broker resources by following the instructions in “Backing up WebSphere Message Broker Version 6.1 resources” on page 228.
- Install a version of WebSphere MQ that supports IBM Integration Bus Version 9.0 alongside your existing WebSphere MQ deployment.
- Optional: If your applications use publish/subscribe functions, and you have message flows that include one or more Publication nodes, you must redefine

your subscriptions against WebSphere MQ before you migrate the broker; for more information, see “Migrating publish/subscribe information to WebSphere MQ” on page 244.

- Migrate the IBM Integration Toolkit by following the instructions in “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.
- Check that you have no aggregations in progress on this broker. When you migrate a broker to Version 9.0, all live data that is being stored for aggregations in progress is lost.
- If the broker runs in a locale that is not listed in “Locales” on page 664, check that the code page is one of the supported code pages, and that the locale is set up correctly.

Migrating application logic to an IBM Integration Bus Version 9.0 broker:

About this task

To migrate application logic from a WebSphere Message Broker Version 6.1 broker to a different IBM Integration Bus Version 9.0 broker on the same computer, or on a different computer, on a distributed operating system, complete the following steps.

Procedure

1. Install IBM Integration Bus Version 9.0 on the same computer as Version 6.1, or on a different computer. Install at least the Integration Bus component; other components are optional. You must specify a different location if you are installing on the same computer.
2. Optional: If your message flows access user databases by using ODBC connections, update the ODBC definitions to Version 9.0 format by following the instructions in “Updating ODBC definitions when migrating” on page 209.
3. Set up the correct Version 9.0 command environment:
 - **Linux** **UNIX** On Linux and UNIX systems, open a new shell and run the environment profile **mqsiprofile** for the Version 9.0 installation.
 - **Windows** On Windows, click **Start**, and open the Command Console that is associated with the Version 9.0 installation.
On Windows systems, you must open a command console with elevated privileges. To open a command console with elevated privileges, use the **mqsicommandconsole** command. For more information, see **mqsicommandconsole** command.
4. Create a queue manager, with a unique name, on your new WebSphere MQ deployment, ensuring that you source the correct WebSphere MQ environment. Because your existing WebSphere MQ deployment is the primary installation, you must configure your environment by using the WebSphere MQ **setmqenv** command before you can run commands against the new WebSphere MQ deployment. For information about the **setmqenv** command, see WebSphere MQ Version 7.5 Information Center online.
5. Create a Version 9.0 broker by using the **mqsicreatebroker** command; give it a name that is different from the name of the Version 6.1 broker and give the name of the queue manager you created on your new WebSphere MQ deployment.
6. Start the Version 9.0 broker by using the **mqsistart** command.
7. Write a list of the integration servers that you have on the Version 6.1 broker, and create these same integration servers on the Version 9.0 broker.

To complete this step, use the IBM Integration Explorer, the Version 9.0 IBM Integration Toolkit, or the **mqsicreateexecutiongroup** command. On the AIX, Linux on x86-64, and Solaris on SPARC systems, you can create only 64-bit integration servers.

8. Deploy the message flows and message sets that are in use by the Version 6.1 broker to the Version 9.0 broker from the Version 9.0 IBM Integration Toolkit. You cannot complete this step unless you already migrated the IBM Integration Toolkit resources, as described in “Migrating development resources to IBM Integration Toolkit Version 9.0” on page 182.
9. Configure all other relevant properties of the Version 6.1 broker on the Version 9.0 broker.
10. If you want to delete your Version 6.1 broker, complete the following steps.
 - a. In a Version 6.1 command environment, stop the Version 6.1 broker by using the **mqsistop** command.
 - b. Remove the Version 6.1 broker from the Version 6.1 IBM Integration Toolkit.
 - c. Redeploy the Version 6.1 topology from the Version 6.1 IBM Integration Toolkit to update the configuration data that is held by the Configuration Manager.
 - d. In a Version 6.1 command environment, delete the Version 6.1 broker by using the **mqsdeletebroker** command.

What to do next

When you have completed migration, see the tasks in “Perform post-migration tasks” on page 252 for information about tasks that you might want to perform after migration.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

“Backing up WebSphere Message Broker Version 6.1 resources” on page 228

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.

“Creating a broker” on page 304

You can create brokers on every platform that is supported by IBM Integration Bus. The broker runs as a 64-bit application on all platforms except Linux on x86 and Windows on x86.



Starting and stopping a broker

Run the appropriate command to start or stop a broker.



Creating an integration server using the IBM Integration Toolkit or IBM Integration Explorer

Use the IBM Integration Toolkit or IBM Integration Explorer to create integration

servers on your broker.



Creating an integration server using the **mqscreateexecutiongroup** command
Use the **mqscreateexecutiongroup** command to create integration servers on your broker.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.



mqscommandconsole command

Use the **mqscommandconsole** command to launch an elevated command console from which commands that require elevation on Windows can be run.



mqsdeletebroker command

Use the **mqsdeletebroker** command to delete a named broker. The command also deletes the queues on the associated queue manager (created when the broker was created). You can also specify that the queue manager is to be deleted.



mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.



mqsistop command

Use the **mqsistop** command to stop the specified component.

Migrating Configuration Manager ACLs

If you are migrating from WebSphere Message Broker Version 6.1, you can use the Access Control Lists (ACLs) that you set up in the Configuration Manager as the basis for your security model in IBM Integration Bus Version 9.0.

Before you begin

Before you start:

- Migrate your broker by following the instructions in “Performing in-place broker migration for a Version 6.1 broker” on page 231.
- Activate broker administration security by following the instructions in Enabling administration security.

Administration security for a broker in IBM Integration Bus Version 9.0 is disabled by default. Because no security details are migrated, when you migrate a broker from a previous version security is disabled, even if you had set up security in the previous version.

About this task

The Configuration Manager has been removed as a result of the architectural and security model changes that were made in Version 7.0, therefore the commands that controlled this component and its ACLs have been removed:

```
mqschangeconfigmgr  
mqscreateconfigmgr  
mqsdeleteconfigmgr  
mqsireportconfigmgr  
mqscreateaclentry  
mqsdeleteaclentry
```

mqsilistaclentry

Although you can use the ACLs that you set up in previous versions as the basis for your security model in Version 9.0, you might choose to reconsider your security model, and set up different levels of control that match the facilities available in this version.

If you choose to use your existing ACLs as a basis for your security model in Version 8.0, consider the following factors:

- You can obtain a list of existing ACLs maintained by your Configuration Manager by using the **mqsilistaclentry** command.
- You can migrate existing groups, users, or both, subject to the following factors:
 - You must define the user ID on the same computer as the queue manager associated with the Version 9.0 broker.
 - On Linux and UNIX systems, you can grant authority only to the primary group of a user. All users that you have defined with the same primary group automatically get the same level of security access. You must therefore consider the membership of all your groups to ensure that you give the required level of control to each user.
- You can grant only broker and execution level authorities for Version 9.0 brokers.
- If you grant read, write, and execute authority to a user ID for a Version 9.0 broker, this permission is equivalent to full control access in previous versions.
- If you grant read authority to a user ID for a Version 9.0 broker, this permission is equivalent to view access in previous versions.
- Check the authority that write and execute permissions grant for a Version 9.0 broker to determine the best match for edit and deploy access levels in previous versions.
- Although you can set up access for a particular computer or domain name in previous versions, you can use only user IDs and groups in Version 9.0. If you want to establish a more secure environment, consider the use of WebSphere MQ security exits and SSL.

To set up authorization for your Version 9.0 broker, complete the following steps.

Procedure

1. Review the authorizations that are required for specific tasks and commands. Details are provided in Tasks and authorizations for administration security and Commands and authorizations for broker administration security.
2. Grant the authorities that your users require.
 - For details of this task on distributed systems, with examples, see Granting and revoking authority on Linux, UNIX, and Windows systems.
 - For details of this task on z/OS, with examples, see Granting and revoking authority on z/OS systems.

What to do next

Next: Start the broker in the IBM Integration Explorer, or run the **mqsistart** command.

Related concepts:

Security overview

When you are designing an IBM Integration Bus application it is important to consider the security measures that are needed to protect the information in the system.

Related tasks:

Enabling administration security

Enable administration security on a broker to control which users can complete specific tasks against that broker and its resources.

Authorizing users for administration

Grant authority to one or more groups or users to authorize them to complete specific tasks against a broker and its resources.

Enabling SSL on the WebSphere MQ Java Client

The WebSphere MQ Java Client supports SSL-encrypted connections over the server-connection (SVRCONN) channel between an application and the queue manager. Configure SSL support for connections between applications that use the CMP (including the IBM Integration Toolkit and the IBM Integration Explorer) and a broker.

Related reference:

Commands

All IBM Integration Toolkit and runtime commands that are provided on distributed systems are listed, grouped by function, with references to command details.

Related information:

 [WebSphere MQ Version 7 product documentation](#)

Migrating publish/subscribe information to WebSphere MQ

If you have publish/subscribe applications and you want to preserve the publish/subscribe state of these applications when you migrate, you must move the durable subscriptions, retained publications and access control lists out of WebSphere Message Broker Version 6.1 and into your broker's queue manager during migration.

About this task

You must migrate publish/subscribe information to WebSphere MQ after you migrate your queue manager but before you migrate your broker.

To migrate your existing publish/subscribe information, complete the following steps:

1. Read the following topics to help you plan the migration of your publish/subscribe information:
 - ACL migration
 - “Retained publications with headers in MQRFH format” on page 247
 - Migrating a collective
2. Complete the migration steps; see “Migrating publish/subscribe from WebSphere Message Broker Version 6.1 to WebSphere MQ” on page 245

Access Control List (ACL) migration - publish/subscribe:

The function that migrates publish/subscribe configuration data from WebSphere Message Broker Version 6.1 to WebSphere MQ produces a file containing suggested security commands, and creates topic objects as required.

On WebSphere Message Broker Version 6.1, the default behavior is for all user IDs to have access to any topic unless the ACL explicitly restricts access. In WebSphere MQ the default behavior is for no user ID to have access to any topic unless the ACL explicitly authorizes access, and it is not possible to explicitly restrict access. Because of this difference in security approaches, the migration process cannot directly migrate WebSphere Message Broker Version 6.1 ACLs to the WebSphere MQ queue manager.

If the rehearsal or initial phase of the migration finds an ACL entry that denies access, the process cannot produce a WebSphere MQ equivalent command. Instead, it reports it in the security command file and advises that the ACL migration must be performed manually.

You can modify the:

- Broker security settings to match the WebSphere MQ security approach, and run the rehearsal or initial phase of the migration again to produce a new security command file, or
- Security command file as needed. You must set up a security environment in the queue manager, equivalent to the one that existed in the broker, before you run the completion phase of the migration.

WebSphere Message Broker Version 6.1 provides the capability to define topic trees, but there is no capability to set specific attributes for a particular individual topic in a topic tree. WebSphere MQ supports the concept of topic objects that allow you to set specific, nondefault attributes for a topic. An Access Control List is a property of a topic object. The initial phase of the migration creates topic objects speculatively, based on the ACL entries that are defined in the broker and in anticipation of you subsequently running the security commands to create ACLs for the topic objects. When you have resolved what security settings you need, you might need to delete the topic objects that you do not require.

Migrating publish/subscribe from WebSphere Message Broker Version 6.1 to WebSphere MQ:

You can migrate publish/subscribe configuration data from WebSphere Message Broker Version 6.1 to WebSphere MQ.

Before you begin

Before you start:

For the latest details of all supported levels of hardware and software, visit the [IBM Integration Bus Requirements website](#).

This topic assumes the following prerequisite steps:

- You are currently using the broker as an interface for functions other than, but in addition to, publishing and subscribing, and you want to continue to use those other functions after you migrate publish/subscribe information to WebSphere MQ.

- On distributed systems, you have set up and initialized a command environment in which WebSphere MQ and WebSphere Message Broker commands can run.
- You have installed a version of WebSphere MQ that supports IBM Integration Bus and you have migrated your Version 6.1 broker's queue manager.
- The queue manager attribute **PSMODE** is set to **COMPAT**; run the command **ALTER QMGR PSMODE(COMPAT)**.
- You have not migrated your Version 6.1 broker to Version 9.0 and the broker's database definitions are still valid for WebSphere Message Broker Version 6.1.
- Your Version 6.1 broker is stopped.

Note: WebSphere MQ publish/subscribe functionality might alter the header chains in your published message. The message body is not affected, but avoid assumptions about the value of the **CodedCharSetId** field in MQRFH2 headers when subscribing.

About this task

To migrate publish/subscribe configuration data from a WebSphere Message Broker Version 6.1 broker to WebSphere MQ, complete the following steps:

Procedure

1. Start the broker on your WebSphere Message Broker Version 6.1 system but do not attempt to migrate the broker yet.
2. Run the migration process with the **-r** parameter.

For information about the **migmbbrk** command, see the WebSphere MQ information center. This option rehearses the migration of the publish/subscribe configuration data from the broker to its underlying queue manager without changing either of the configurations. For example, on supported systems other than z/OS, use the following command to rehearse the migration from a broker named BRK1:

```
migmbbrk -r -b BRK1
```

On z/OS use the equivalent parameters with the **CSQUMGMB** utility.

3. Run the following command:

```
migmbbrk -t -b brokername
```

then run the security commands that set up the security environment. If you do not run these commands, migration fails.

4. Run the migration process with the **-o** parameter.

The migration process migrates the publish/subscribe configuration data to the queue manager, and creates a log file and a new security commands file.

For example, on supported systems other than z/OS, use the following command:

```
migmbbrk -c -o -b BRK1
```

This command completes the following actions:

- Migrates the publish/subscribe configuration data from broker BRK1
- Overwrites any subscription or retained publication that exists in the queue manager
- Creates a file that has the same name as a migrating subscription or retained publication

- On z/OS use the equivalent parameters with the CSQUMGMB utility.
5. Stop the broker and check the Administration log to confirm that the broker has stopped.

Note: After you have stopped the broker, the queue manager is in COMPAT mode. As a result, the publish/subscribe state cannot be changed in WebSphere MQ or WebSphere Message Broker, allowing the migration to proceed.

6. Run the commands that are in your copy of the security commands file.
7. To set the **PSMODE** queue manager attribute to ENABLED, run the following command:

```
ALTER QMGR PSMODE(ENABLED)
```

This command starts the queued publish/subscribe interface so that the queue manager deals with all subsequent publish/subscribe processing.

8. Continue with the migration of your Version 6.1 broker. When migration is complete, enable content-based filtering on your Version 9.0 broker (if you are using that feature). For more information, see Enabling content-based filtering with publish/subscribe .


Related tasks:

“Migrating publish/subscribe information to WebSphere MQ” on page 244

If you have publish/subscribe applications and you want to preserve the publish/subscribe state of these applications when you migrate, you must move the durable subscriptions, retained publications and access control lists out of WebSphere Message Broker Version 6.1 and into your broker's queue manager during migration.

Related information:

 [WebSphere MQ Version 7.1 product documentation](#)

 [WebSphere MQ Version 7.5 product documentation](#)

Retained publications with headers in MQRFH format:

Retained publications in MQRFH format might lose data when migrated to WebSphere MQ.

IBM Integration Bus applications that communicate with one another using publish/subscribe can do so regardless of the message format that they use. IBM Integration Bus delivers the message in the format of the subscription and provides automatic conversion to ensure that a subscriber receives messages in the requested format.

IBM Integration Bus applications generally use the MQRFH2 message header, but it is possible that an application might have used the MQRFH format.

The migration of publish/subscribe information from WebSphere Message Broker to WebSphere MQ requests messages in MQRFH2 format. It is rare for IBM Integration Bus client applications to use messages in MQRFH format. However, if an application does use retained messages in MQRFH format, it is possible that some truncation of data might occur upon migration. In particular, data passed using the MQPSSStringData and MQPSIntData name/value pairs is not migrated.

The migration function checks for two conditions in the data that is returned from the broker:

1. That there is at least one retained message stored in the broker

2. That there is at least one MQRFH subscription

If both these conditions are true, the migration function displays a warning message and writes a warning message in the migration log stating that MQRFH retained publications have been migrated with a possible loss of data.

MQRFH2 retained publications are migrated correctly.

Migrating publish/subscribe collectives from WebSphere Message Broker Version 6.1 to WebSphere MQ:

Complete these tasks to migrate the configuration data from publish/subscribe collectives in WebSphere Message Broker Version 6.1 to WebSphere MQ queue-manager clusters.

Before you begin

Before you start:

For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.

This task assumes the following prerequisites:

- You want to migrate publish/subscribe collectives from WebSphere Message Broker Version 6.1 to WebSphere MQ queue manager clusters.
- On distributed systems, you have set up and initialized a command environment in which WebSphere MQ and WebSphere Message Broker commands can run.
- You have installed a version of WebSphere MQ that supports IBM Integration Bus and you have migrated your Version 6.1 broker's queue manager.
- You have not migrated your Version 6.1 broker to Version 9.0 and the broker's database definitions are still valid for WebSphere Message Broker Version 6.1.
- Your Version 6.1 broker is stopped.

This process is for migration only. You cannot administer new IBM Integration Bus Version 9.0 brokers or update, for example, publish/subscribe topologies, or Access Control Lists

If you inadvertently initiate the deployment of a publish/subscribe topology, any brokers that are already migrated to IBM Integration Bus Version 9.0 ignore this part of the deployment and issue a BIP2082 warning message to the system log.

If you use this mixed environment, you must use the **migmbbrk** command as part of the migration for each broker. For information about the **migmbbrk** command, see the WebSphere MQ information center.

About this task

To migrate a publish/subscribe collective, complete the following steps:

Procedure

1. Remove each broker from the collective. For more information, see the WebSphere Message Broker Version 6.1 information center.

2. Migrate the publish/subscribe information from each broker, as described in “Migrating publish/subscribe from WebSphere Message Broker Version 6.1 to WebSphere MQ” on page 245.

What to do next

Next:

When you have migrated all the brokers, you can place the brokers in a queue manager cluster. For information about setting up a new queue manager cluster, see the WebSphere MQ Version 7.5 Information Center online

Related tasks:

“Migrating publish/subscribe information to WebSphere MQ” on page 244

If you have publish/subscribe applications and you want to preserve the publish/subscribe state of these applications when you migrate, you must move the durable subscriptions, retained publications and access control lists out of WebSphere Message Broker Version 6.1 and into your broker's queue manager during migration.

Related information:

[➤ WebSphere Message Broker product documentation](#)

[➤ WebSphere MQ Version 7.1 product documentation](#)

[➤ WebSphere MQ Version 7.5 product documentation](#)

Migrating existing z/OS applications

You must migrate your existing z/OS applications to be 64-bit before you can use them in IBM Integration Bus Version 9.0.

You must recompile and link your C code and user exits on z/OS by using XPLINK, ensuring that you use the following elements:

- LP64 flag
- IEEE Floating Point format

Therefore, you require the following options:

- LP64
- FLOAT (IEEE) - the default when you use LP64
- XPLINK - the default when you use LP64

When you make WebSphere MQ calls from within your C code or user exits, you need to link against the following stub:

```
++WMQHLQ++.SCSQDEFS(CSQBMB2X)
```

where ++WMQHLQ++ refers to the high-level qualifier for the WebSphere MQ datasets.

You must use the `cmqcmh.h` header file shipped in WebSphere Message Broker Version 8.0 to recompile code that uses WebSphere MQ structures.

Using the `cmqc.h` header file that is provided by WebSphere MQ results in compilation errors, as `cmqc.h` explicitly disallows 64-bit compilation.

Note, that compilation by using the `cmqcmh.h` header file, and linking against CSQBMB2X are supported only for user extensions that run within the IBM Integration Bus Version 9.0 address space.

See [Compiling a C user-defined extension](#) for examples of 64-bit compilation.

Related tasks:

Chapter 3, “Migrating,” on page 171

To migrate a broker domain to IBM Integration Bus Version 9.0, plan your migration strategy, perform pre-migration tasks, migrate your domain components, then complete post-migration tasks.



[Compiling a C user-defined extension](#)

Compile user-defined extensions in C for all supported operating systems.

Restoring components and resources to Version 6.1

Restore components and resources that you migrated from Version 6.1 products to their original state.

About this task

If you migrated a broker from WebSphere Message Broker Version 6.1 to IBM Integration Bus Version 9.0, you cannot revert that broker to Version 6.1. Instead you must restore the broker and its queue manager from a backup.

Other brokers on the system do not need to be changed. Keep your WebSphere Message Broker Version 6.1 and WebSphere MQ Version 7.0.1 installations on your system in their original locations until you know that you do not need to restore any brokers to Version 6.1.

Related tasks:

Chapter 3, “Migrating,” on page 171

To migrate a broker domain to IBM Integration Bus Version 9.0, plan your migration strategy, perform pre-migration tasks, migrate your domain components, then complete post-migration tasks.

“Backing up WebSphere Message Broker Version 6.1 resources” on page 228

Back up your resources before you start to migrate components to IBM Integration Bus Version 9.0.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

Migrating the WebSphere Message Broker Explorer

Migrate from a previous version of the WebSphere Message Broker Explorer to the IBM Integration Explorer Version 9.0.

Before you begin

Before you start:

About this task

One version only of the WebSphere Message Broker Explorer or IBM Integration Explorer can be active at any one time because one version only of the WebSphere MQ Explorer can be installed on any one system.

If you already have WebSphere Message Broker Explorer installed, and you decide to install IBM Integration Explorer, you must replace the existing version.

If you add another installation of WebSphere MQ after you install IBM Integration Explorer, you must install IBM Integration Explorer again to enable it in all WebSphere MQ installations.

To migrate from a previous version of WebSphere Message Broker Explorer to IBM Integration Explorer Version 9.0, complete the following steps:

1. Uninstall your current version of WebSphere Message Broker Explorer. You can uninstall the WebSphere Message Broker Explorer only if you are the administration user (for example, root on Linux on x86 and Linux on x86-64, or Administrator on Windows). For more information about uninstalling WebSphere Message Broker Explorer, see one of the following topics:
 - For WebSphere Message Broker Explorer Version 7.0, see [Uninstalling WebSphere Message Broker Explorer Version 7.0](#)
 - For WebSphere Message Broker Explorer Version 8.0, see [Uninstalling WebSphere Message Broker Explorer Version 8.0](#)
2. Check that no files exist in the previous installation directory.
3. Install IBM Integration Explorer.

After you install the new version of the IBM Integration Explorer, if you are upgrading from an existing version of the WebSphere Message Broker Explorer, you must complete the following steps:

1. To initialize the new version of the IBM Integration Explorer, run the following command from a command line in which the **mqsiprofile** command has not run:

```
strmqcfg -i
```

To run this command, you must be the administration user.
2. Start IBM Integration Explorer either by using the IBM Integration Explorer Windows shortcut or by running the following command from a command line in which the **mqsiprofile** command has not run:

```
strmqcfg -c -d
```

Windows On Windows, the **Add/Remove** program panels point to the new installation location. To remove the older version, navigate to the old location on the file system and run the uninstaller that is located there.

If you previously installed the IS02: IBM Integration Explorer plug-in SupportPac, you must complete the following steps before you install IBM Integration Explorer:

1. Move or delete the location to which you extracted the IS02 SupportPac.
2. Delete the `IntegrationExplorer.link` file, which you copied into the `links` directory. On Windows the directory name is `C:\Program Files\IBM\WebSphere MQ\eclipseSDK33\eclipse\links\`. On Linux on x86 and Linux on x86-64, the directory name is `/opt/mqm/eclipseSDK33/eclipse/links`.
3. Start IBM Integration Explorer either by using the IBM Integration Explorer Windows shortcut or by running the following command from a command line in which the **mqsiprofile** command has not run:

```
strmqcfg -c -d
```
4. Optional: If you want a new installation of your IBM Integration Explorer, complete one of the following steps:
 - Move or delete the IBM Integration Explorer metadata directory. On Windows, the directory name is `C:\Documents and Settings\<user>\IBM\MQ Explorer\.metadata`. On Linux on x86 and Linux on x86-64, the directory name is `/home/user/.mqdata/.metadata`.

- Alternatively, switch your Eclipse workspace.

Moving or deleting the metadata directory, or switching your Eclipse workspace, resets your IBM Integration Explorer to its initial startup state. You must then reconnect to any remote queue managers. Any local queue manager is discovered automatically.

Results

You can now use the new version of IBM Integration Explorer. To start the IBM Integration Explorer, complete one of the following steps:

- On Windows:
Click **Start > All Programs > IBM Integration Bus 9.0.0.0 > IBM Integration Explorer**.
- On Linux on x86 and Linux on x86-64:
Open a command shell in which the **mqsiprofile** command has not run and enter the **strmqcfg** command, or run **/usr/bin/strmqcfg**.

Related concepts:



IBM Integration Explorer

The IBM Integration Explorer is a graphical user interface based on the Eclipse platform for administering your integration nodes (brokers).

Related tasks:

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Perform post-migration tasks

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

About this task

Test the IBM Integration Bus Version 9.0 broker resources and components to verify that you experience no loss or unexpected change of functionality. Some changes in behavior might be caused by defects that were fixed between versions.

The following topics describe further tasks that you can complete after migration:

Procedure

- “Reviewing technical changes in IBM Integration Bus Version 9.0” on page 253
- “Migrating deployable resources” on page 259
- “Setting up a command environment” on page 119
- “Migrating a flow that contains JMS nodes” on page 263
- “Migrating a flow that contains File nodes” on page 264
- “Migrating a flow supporting ?wsdl queries” on page 265
- “Migrating CMP applications” on page 265
- “Updating error processing routines” on page 266

What to do next

Next: Complete the following tasks as required:

- Delete the Version 8.0, Version 7.0 or Version 6.1 components.

If the broker that you migrated shares a queue manager with the Version 6.1 Configuration Manager, the default server connection channel `SYSTEM.BKR.CONFIG`, which was created when you created the Configuration Manager, is deleted when you delete the Configuration Manager. The Version 9.0 broker requires this channel for connections from the IBM Integration Explorer, the IBM Integration Toolkit, and CMP applications. You must re-create this channel on the broker queue manager after you delete the Configuration Manager.

- Remove the installed code for Version 8.0, Version 7.0 or Version 6.1.
- Delete the broker database from Version 6.1. You can also remove the installed database product, if you do not use it for other purposes.

Access the Version 8.0, Version 7.0 or Version 6.1 information center for details of these tasks.

Related concepts:

“Parallel broker migration” on page 180

IBM Integration Bus Version 9.0 supports parallel broker migration from previous versions, with some restrictions.

Related tasks:

“Migrating publish/subscribe information to WebSphere MQ” on page 244

If you have publish/subscribe applications and you want to preserve the publish/subscribe state of these applications when you migrate, you must move the durable subscriptions, retained publications and access control lists out of WebSphere Message Broker Version 6.1 and into your broker's queue manager during migration.

“Migrating the Version 6.1 WebSphere Message Broker component” on page 225

Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206

Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

Related reference:

“Supported migration paths” on page 177

You can migrate to IBM Integration Bus Version 9.0 from previous versions of the product.

Related information:

 [WebSphere Message Broker product documentation](#)

 [WebSphere Message Broker 7.0 product documentation](#)

 [WebSphere Message Broker 8.0 product documentation](#)

Reviewing technical changes in IBM Integration Bus Version 9.0

Some minor changes in behavior are present in IBM Integration Bus Version 9.0 if you are migrating from Version 7.0 or Version 6.1; for example, those changes that are caused by defects that were fixed between versions.

If you are migrating from Version 8.0, Version 7.0, or Version 6.1, read the following sections to understand the potential effects on your broker and message flows:

- “XMLNSC timezone handling” on page 254

If you are migrating from Version 7.0 or Version 6.1, read the following sections to understand the potential effects on your broker and message flows:

- “CMP applications”
- “Record and replay” on page 255
- “Resource statistics and message flow accounting and statistics XML publication messages” on page 255
- “Upgrade to ICU V4.8” on page 255

If you are migrating from Version 6.1, read the following sections to understand the potential effects on your broker and message flows:

- “Changes to user ID and password requirements” on page 255
- “Changes to location of description properties” on page 256
- “Default Configuration wizard and database usage on Windows” on page 256
- “Default integration server when you create brokers” on page 256
- “Starting and stopping integration servers” on page 256
- “Using SOAPAsyncRequest, SOAPInput, and SOAPRequest nodes” on page 257
- “Using HTTPS with HTTPInput and HTTPReply nodes” on page 257
- “Monitoring message flows” on page 257
- “ESQL field references with an index of zero” on page 258
- “Using RegistryLookup nodes” on page 258
- “Interfaces in the IBM Integration Toolkit” on page 258

XMLNSC timezone handling

In versions before IBM Integration Bus Version 9.0, timezones are not passed between the XMLNSC parser and ESQL or other parsers. This behavior is corrected in Version 9.0. Timezones might now display in additional places but you can revert to the original behavior if required:

- To revert to the original behavior for parsing XML documents, run the following command:

```
mqsichangeproperties NODENAME -e ISNAME -o ComIbmGenericXmlParserFactory -n storeTimeZoneInValue -v
```

where *NODENAME* is the name of your integration node and *ISNAME* is the name of your integration server.

- To revert to the original behavior for writing XML documents, run the following command:

```
mqsichangeproperties NODENAME -e ISNAME -o ComIbmGenericXmlParserFactory -n writeTimeZone -v OPTION
```

where *OPTION* has one of the following values:

Table 8. Options for the `-v` flag

Value	Description
never	Do not display any timezones.
nonUTC	Display all timezones except for UTC
nonLocal	Display all timezones except for the local timezone

CMP applications

CMP applications that you develop in Version 9.0 can connect to your existing Version 8.0 and Version 7.0 brokers, and your existing Version 8.0 and Version 7.0 CMP applications can connect to Version 9.0 brokers.

However, if you are migrating from Version 6.1, you must update your CMP applications to use the file that is supplied by Version 9.0 before you can connect to a Version 9.0 broker.

For more information, see “Migrating CMP applications” on page 265.

Record and replay

The record and replay feature can be used only with Version 9.0 or Version 8.0 brokers, or brokers that you migrated to Version 9.0. It cannot be used with existing Version 7.0 or Version 6.1 brokers.

For more information, see Record and replay.

Resource statistics and message flow accounting and statistics XML publication messages

Resource statistics and message flow accounting and statistics XML publication messages are now published with an MQMD header, which has a FORMAT of MQSTR. This indicates that the publication message is composed entirely of character data.

If an WebSphere MQ JMS application is used to subscribe to the publication topic and read the messages, these messages are represented as a JMS TextMessage and not as a JMS BytesMessage.

Note: If required, you can revert to using a JMS BytesMessage format by setting the following environment variable: `MQSI_STATS_MQSTR=false`.

Upgrade to ICU V4.8

WebSphere Message Broker Version 8.0 and IBM Integration Bus Version 9.0 ship with ICU V4.8 (International Components for Unicode) for date, time, and codepage conversions. This is an upgrade from ICU 3.8.1, which was shipped with WebSphere Message Broker Version 7.0.

This upgrade results in the following responses:

- Time zones that are invalid receive a BIP2319 response from the broker.
- Time zones that are now obsolete in ICU V4.8.1 are considered invalid. For example, "Alpha Time" and "Bravo Time".
- Date and time formatting that uses the pattern "zzz" might now return a time zone ID that implicitly represents a time zone and offset from GMT rather than "GMT" plus an offset.
- Codepages 897, 1277 and 5297 are no longer supported.

The following behavioral changes apply only if you are migrating from Version 6.1.

Changes to user ID and password requirements

User ID and password management in Version 9.0 is updated:

- The requirement for a service user ID and password is removed on all systems except Windows. These parameters (`-i`, `-a`) are no longer used when you migrate your brokers to Version 9.0.

- Because the Version 9.0 broker has no requirement for a database, the parameters to define and change database user IDs and passwords are removed. If you configured database user IDs and passwords for the brokers that you are migrating, these parameters (**-u**, **-p**) are migrated with the broker, and are used as default values for data sources (databases) for which you do not set explicit values. If you have not configured **-u**, **-p**, the values for **-i**, **-a** are migrated. In Version 9.0, you can manage these user IDs and passwords for your databases by using the **mqsisetdbparms** command.
- The requirement for the broker service user ID to be a member of the mqm group is removed.
- If you change the database user ID and password by using the **mqsisetdbparms** command, you no longer need to restart the broker. You can instead restart the affected integration server by using the **mqsireload** command.

Changes to location of description properties

Long and short description properties of deployed IBM Integration artifacts were not held in the deployed integration server repository, so they are not migrated to the Version 9.0 broker.

If the following fields were used to hold keywords, they are not displayed in the migrated artifacts:

```
$MQSI name = value MQSI$
```

To correct this behavior, redeploy the artifacts directly to the Version 9.0 broker.

For more information about defining keywords, see [Guidance for defining keywords](#).

Default Configuration wizard and database usage on Windows

Some of the sample programs use a database; for example, the Airline sample. If you used the Default Configuration wizard to set up a default configuration on Windows, and deploy samples to the default broker, the samples that require a database use the Derby database that is embedded in the broker. Version 9.0 does not ship or support the Derby database. You must reconfigure your database samples by following the updated instructions in the samples documentation.

Default integration server when you create brokers

When you create a broker by using the **mqsicreatebroker** command, a default integration server is no longer created.

If you use either the IBM Integration Toolkit or the IBM Integration Explorer to create a broker, you can select an option to create a default integration server with the name `default` (unless you specify another name).

You can also create integration servers by using the **mqsicreateexecutiongroup** command.

Starting and stopping integration servers

Starting and stopping integration server behavior is updated in Version 9.0. When you start or stop an integration server by using the **mqsistartmsgflow** or **mqsistopmsgflow** commands without the **-m** parameter, the integration server

process is stopped or started. When you stop the integration server in this way, or by using the IBM Integration Toolkit, or the IBM Integration Explorer, the run state of the message flows deployed to the integration server is recorded. When you next start the integration server only those message flows that were running when the integration server was stopped are restarted, unless you specifically request all flows to be started, or use the `-j` parameter on the command.

Using SOAPAsyncRequest, SOAPInput, and SOAPRequest nodes

The Failure action property of the SOAPAsyncRequest, SOAPInput, and SOAPRequest nodes is changed to be not configurable. If you configured this property, for example in a BAR file, the setting is ignored.

Using HTTPS with HTTPInput and HTTPReply nodes

The Version 9.0 broker checks for required SSL configuration when you run the `mqsistart` command.

If you deployed a message flow that includes HTTPInput or HTTPReply nodes to a Version 6.1 broker, and you migrate the broker to Version 9.0 and start the broker again, you might see the following error message generated. (Message lines are continuous, but are split to improve readability).

```
BIP3135S: An exception occurred while starting the servlet engine connector.
Exception text is HTTP Listener LifecycleException:
Protocol handler start failed: java.io.FileNotFoundException: /home/leed/.keystore
(No such file or directory)
at org.apache.coyote.tomcat5.CoyoteConnector.start(CoyoteConnector.java:1529)
at com.ibm.broker.httplistener.ConnectorWrapper.start(ConnectorWrapper.java:166)
at com.ibm.broker.httplistener.TomcatWrapper.startSecureHTTPSConnector
(TomcatWrapper.java:146)
at com.ibm.broker.httplistener.HTTPListenerManager.ensureServletContainer
(HTTPListenerManager.java:290)
at com.ibm.broker.httplistener.HTTPListenerManager.run(HTTPListenerManager.java:153)
at java.lang.Thread.run(Thread.java:735) :
DANBRK.httplistener: /build/S000_P/src/DataFlowEngine/NativeTrace/ImbNativeTrace.cpp: 732:
ensureServletContainer: :
Oct 13 13:47:16 partick user:err|error WebSphere Broker v8000[303572]:
(DANBRK.default)[1]BIP2275E: Error loading message flow 'ef2a0606-2401-0000-0080-984a4915984c'. :
DANBRK.de427601-2401-0000-0080-d525e90f1528: /build/S000_P/src/DataFlowEngine/ImbDataFlowDirector.cpp:
2957: ImbDataFlowDirector::loadAllDataFlowsFromDatabase:
ExecutionGroup: de427601-2401-0000-0080-d525e90f1528
```

This error is generated because the Version 9.0 broker detects that you configured the HTTP nodes in the message flow to use HTTPS, but you did not configure the required SSL configuration; the broker does not load the message flow. In previous versions, this check is not performed and no error is generated.

To resolve this error, configure your HTTP nodes to use SSL, and redeploy the message flow. For SSL configuration information, see [Configuring HTTPInput and HTTPReply nodes to use SSL \(HTTPS\)](#).

Monitoring message flows

The default behavior for publishing monitoring events is changed. In versions before Version 9.0, monitoring events are emitted out of sync point. Now, the default for all events except transaction rollback is that events are emitted only if

the message flow commits its unit of work successfully. By default, transaction rollback events are emitted in a second unit of work, independent of the main unit of work.

These changes mean that you no longer see events that are backed out because of a failed message flow; you see only the transaction start event and the transaction rollback event, if these events are defined. You also see all other events that are defined to be in an independent unit of work. See *Monitoring basics* for more information.

A sequence number is added to the eventSequence element of the monitoring event. Because both the creation time and sequence number are always emitted in the monitoring event, the Sequence tab is removed from the monitoring tab in the IBM Integration Toolkit.

ESQL field references with an index of zero

The validity of using a field reference index of zero is corrected. If you have statements in your ESQL modules that include an index of zero, error BIP3226E is generated when you deploy the message flow.

For example, if you have code that contains the statement:

```
SET OutputRoot.XMLNSC.Top.A[0].B = 42;
```

You must update the code to contain the following content:

```
SET OutputRoot.XMLNSC.Top.A[1].B = 42;
```

Using RegistryLookup nodes

The default for the Depth Policy property of the RegistryLookup node is changed from the value Return matched showing immediate relationships (for compatibility only) in Version 6.1 to the value Return matched only (Depth = 0) in Version 9.0.

If you do not explicitly set this property on a RegistryLookup node, it uses the default value Return matched only (Depth = 0) to determine the depth of the WSRR query and the contents of the entity data to be returned.

If you want to use the node in deprecated mode in Version 9.0, you must explicitly set the Depth Policy property to the value Return matched showing immediate relationships (For compatibility only), and rebuild the BAR file.

For more information about the RegistryLookup node and its properties, see RegistryLookup node.

Interfaces in the IBM Integration Toolkit

The following changes are present in the IBM Integration Toolkit:

Problems view

In WebSphere Message Broker Toolkit Version 6.1, you can configure the list of problems that are shown in the Problems view by clicking either the icon on the Problems view pane bar, or the down arrow next to the icon, and selecting **Configure filter** from the list of options displayed. In IBM Integration Toolkit Version 9.0, the icon is no longer shown. Click the down arrow that is shown at the right end of the bar, and select **Configure contents**.

Application Development view

In IBM Integration Toolkit Version 9.0, the Application Development view shows pattern instance projects in a separate pane, in addition to other projects in your workspace.

Broker administration perspective

In WebSphere Message Broker Toolkit Version 6.1, you can connect, configure, and deploy to brokers by using the Broker administration perspective in the WebSphere Message Broker Toolkit. In Version 9.0, the Broker Administration perspective is removed, and you can now connect, configure, and deploy to brokers by using the Integration Nodes view in the Integration Development perspective. For more advanced configuration tasks, you can use the IBM Integration Explorer.

Event Log viewer

In IBM Integration Toolkit Version 6.1, deployment responses and messages from the broker are displayed in the Event Log viewer. In Version 9.0, deployment messages from your instance of the IBM Integration Toolkit are displayed in the Deployment Log view, in the Integration Development perspective.

Command assistants

In WebSphere Message Broker Toolkit Version 6.1, you can use the command assistants to create, change, and delete components such as brokers on your local system. In Version 9.0, you can use the Integration Nodes view to create and delete components. Alternatively, you can use the IBM Integration Explorer to create, change, and delete brokers on your local system.

XPath Expression Builder

In WebSphere Message Broker Toolkit Version 6.1, the Data Types Viewer shows two top-level categories, Data Types and Variables. In Version 9.0, you can find the variables under the single top-level category Data Types.

Related tasks:

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188
Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206
Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 6.1 WebSphere Message Broker component” on page 225
Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

“Perform post-migration tasks” on page 252

After you migrate to IBM Integration Bus Version 9.0, finish setting up your environment.

Migrating deployable resources

You can continue to use legacy resources in IBM Integration Bus. However, if you want to continue developing resources created in WebSphere Message Broker Version 6 and Version 7, you must migrate them.

Before you begin

Before you start:

- Read the concept information about applications and libraries: Applications and libraries.
- Read the concept information about subflows: Subflows.
- You can migrate Message Flow projects or Message Broker projects by following the instructions in “Importing resources from previous versions” on page 185.

About this task

You can import Message Flow projects or Message Broker projects into the IBM Integration Toolkit Version 9.0 to migrate your resources from earlier versions. These resources are converted automatically to integration projects. You can then use an imported project as the basis for a new application or library. Some of your migrated resources need further work before you can continue to develop them.

Note: Message flow projects are replaced by Message Broker projects in WebSphere Message Broker Version 8.0.

Note: Message Broker projects are replaced by integration projects in IBM Integration Bus Version 9.0.

The following list outlines the types of resources that require additional migration steps if you want to continue developing them in IBM Integration Bus:

- Message sets: For more information, see “Migrating message sets” on page 261.
- Maps: For more information, see “Migrating message maps” on page 261.
- ESQL files: For more information, see “Migrating ESQL files” on page 261.
- Subflows: For more information, see “Migrating subflows” on page 262.

The following table summarizes the type of resources that you must migrate into IBM Integration Bus Version 9.0 if you want to continue developing them:

Table 9. Legacy resources that require additional migration tasks

To continue developing in IBM Integration Bus Version 9.0	Resource created in Version 6	Resource created in Version 7	Resource created in Version 8
Message sets	You must complete the migration of a message set to a message model schema for all domains except the MRM domain and the IDOC domain.	You must complete the migration of a message set to a message model schema for all domains except the MRM domain and the IDOC domain.	You must complete the migration of a message set to a message model schema for all domains except the MRM domain and the IDOC domain.
Message maps	You must complete the migration of message maps to graphical data maps.	You must complete the migration of message maps to graphical data maps.	You must complete the migration of message maps to graphical data maps for those legacy message maps that you did not convert when you migrated to Version 8. New maps created in Version 8 are graphical data maps.
ESQL files	You must complete the migration tasks.	You must complete the migration tasks.	You must complete the migration tasks.
Subflows	You must complete the migration of subflows created as .msgflow files into .subflow files.	You must complete the migration of subflows created as .msgflow files into .subflow files.	You must complete the migration of the legacy subflows that you did not convert when you migrated to Version 8. New subflows created in Version 8 are created as .subflow files.

Migrating message sets

About this task

In WebSphere Message Broker Version 8.0 and later, *message model schema* files contained in applications, integration services, and libraries are the preferred way to model messages for most data formats. Message sets are required if you use the MRM or IDOC domains. For more information about message modeling, see Message modeling concepts.

You can import message flows containing message sets from WebSphere Message Broker Version 7.0 into IBM Integration Bus Version 9.0. Your existing message sets can be viewed, compiled, and deployed. However, by default, your message sets are accessible in read-only mode and cannot be modified.

If you need to modify your existing message sets, or create new message sets or message definition files, you must first enable message set development in the IBM Integration Toolkit.

For more information, see Enabling message set development.

Migrating message maps

About this task

IBM Integration Bus Version 9.0 includes a graphical data mapping capability, which is used when you add a Mapping node to a message flow.

You can import message flows containing the following nodes, which use message mapping, from WebSphere Message Broker Version 7.0 into IBM Integration Bus Version 9.0:

- DataDelete
- DataInsert
- DataUpdate
- Extract
- Mapping
- Warehouse

The message mapping (.msgmap) on these Version 7.0 nodes can be viewed, compiled into a BAR file, and deployed to run the same transformation logic that was used in Version 7.0. However, the Version 7.0 message mapping operations are accessible only in read-only mode and cannot be modified. If you want to modify the transformation logic of an imported message flow that used Version 7.0 message mapping operations, you must replace the node with a new Mapping node and build a new graphical data map file (.map).

For information about converting Version 7.0 message maps (.msgmap) to Version 9.0 graphical data maps (.map), see Converting a message map from a .msgmap file to a .map file.

Migrating ESQL files

About this task

The required character set encoding for ESQL files that are used in IBM Integration Bus Version 9.0 is UTF-8. When a message flow project is migrated to an integration project, any ESQL files that the message flow project contains are read

by using the character set encoding that was used in the previous version of WebSphere Message Broker, and are rewritten in UTF-8. When importing your message flow projects, note that the file encoding of the workspace and host operating system must match the file encoding with which the message flow projects were created.

Migrating subflows

About this task

Since WebSphere Message Broker Version 8, you create subflows as `.subflow` files.

When you import subflows created in earlier versions of the product into IBM Integration Bus, that is, subflows created as `.msgflow` files, you must convert them into `.subflow` files if you want to continue developing them. You convert these subflows by using the function **Convert to subflow**.

For more information on how to convert a legacy subflow into a `.subflow` file, see [Converting between message flows and subflows](#).

Related concepts:



Applications and libraries

Applications and libraries are deployable containers of resources, such as message flows, subflows, message definitions (DFDL, XSD files), JAR files, XSL style sheets, and WebSphere Adapters files.



Integration projects

An integration project is a specialized container in which you create and maintain all the resources associated with one or more message flows.

Related tasks:



Managing message flow resources

You can use applications, libraries, and integration projects to organize your resources.

[“Importing resources from previous versions” on page 185](#)

You can work with resources from previous versions of WebSphere Message Broker by using a project interchange file to import the resources into IBM Integration Bus Version 9.0.



Converting a project to an application or library

Two options are available to convert existing projects to applications or libraries. Use **Convert Single Project** to convert a project with no references. Use **Analyze and Convert Multiple Projects** to convert one or more projects and any referenced resources.



Converting between message flows and subflows

Convert a `.msgflow` file to a `.subflow` file, or a `.subflow` file to a `.msgflow` file.

Setting up a command environment

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

About this task

Also complete this task if you have migrated to IBM Integration Bus Version 9.0 from an earlier version. A profile is provided to help you set up the environment.

If appropriate, you can extend the initialization performed by this profile; for example, for databases, or for other products that you want to use within the broker.

Ensure that you use this environment each time you run an administrative command, or start a broker.

For information about setting up your command and runtime environment on a Windows system, see “Command environment: Windows systems” on page 120.

For information on setting up your command and runtime environment on Linux and UNIX systems, see “Command environment: Linux and UNIX systems” on page 124.

For information about setting up your command and runtime environment for brokers on Linux and UNIX systems, see “Broker-specific command environment: Linux and UNIX systems” on page 127.

For information about setting up your command and runtime environment for integration servers on Linux and UNIX systems, see “Integration server-specific command environment: Linux and UNIX systems” on page 128.

For information about setting up your command and runtime environment for brokers on Windows systems, see “Broker-specific command environment: Windows systems” on page 122.

For information about setting up your command and runtime environment for integration servers on Windows systems, see “Integration server-specific command environment: Windows systems” on page 123.

Related tasks:

“Migrating the Version 8.0 WebSphere Message Broker component” on page 188
Migrate your Version 8.0 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 7.0 WebSphere Message Broker component” on page 206
Migrate your Version 7.0 components and resources to IBM Integration Bus Version 9.0.

“Migrating the Version 6.1 WebSphere Message Broker component” on page 225
Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

“Enabling ODBC connections to the databases” on page 369
Set up the resources and environment that the broker requires for Open Database Connectivity (ODBC) connections to databases on distributed systems.

“Command environment: Windows systems” on page 120
Set up the Windows environment to run IBM Integration Bus commands.

“Command environment: Linux and UNIX systems” on page 124
Set up the Linux or UNIX environment to run IBM Integration Bus commands.

Related reference:

 Runtime commands

The topics in this section describe the IBM Integration Bus runtime commands.

Migrating a flow that contains JMS nodes

When you migrate a message flow from Version 7.0 that contains JMS nodes, you might have to update the value of the Transaction mode property.

About this task

In IBM Integration Bus Version 9.0, the valid values for Transaction mode on the JMS nodes are Yes and No. In versions before WebSphere Message Broker Version 8.0, the valid values were Local, Global and None.

When you migrate a message flow that contains JMS nodes with the value of Transaction mode set to None, the value is automatically migrated to No.

When you migrate a message flow that contains JMS nodes with the value of Transaction mode set to Local or Global, a migration warning is shown that prompts you to change the value to Yes. If you require XA coordinated transactions, you must also select the message flow Coordinated Transaction property.

The migration warning is displayed until you change the value appropriately. The warning does not prevent deployment of message flows.

Related reference:



JMSInput node

Use the JMSInput node to receive messages from JMS destinations. JMS destinations are accessed through a connection to a JMS provider.



JMSOutput node

Use the JMSOutput node to send messages to JMS destinations.



JMSReply node

Use the JMSReply node to send messages to JMS destinations.



JMSReceive node

Use the JMSReceive node to receive messages from JMS queues in the middle of a message flow. JMS queues are accessed through a connection to a JMS provider.

Migrating a flow that contains File nodes

If you migrate message flows that contain File nodes to IBM Integration Bus Version 9.0, ensure that your NFS server appropriately supports file locking.

About this task

If you use an NFS server, and have File nodes in different integration servers in IBM Integration Bus Version 9.0 that access the same directory on the NFS server, ensure that you are using NFS version 4 to correctly support file locking.

Related tasks:



Working with files



Problems when developing message flows with file nodes

Use the advice given here to help you to resolve some common problems that can arise when you develop message flows that contain file nodes.

Related reference:



FileInput node

Use the FileInput node to process messages that are read from files.



FileOutput node

Use the FileOutput node to write messages to files.



FileRead node

Use the FileRead node to read one record, or the entire contents of a file, from within a message flow.

Migrating a flow supporting ?wsdl queries

If you want WSDL and XML Schema information to be made available for existing SOAPInput-SOAPReply flows that implement web services, you must explicitly set the SOAPInput node property Enable support for ?wsdl and redeploy the flow.

About this task

In the unlikely event you have implemented an HTTPInput-HTTPReply flow to support ?wsdl queries related to a SOAPInput-SOAPReply flow, you should now deprecate the HTTP flow. By sending ?wsdl requests directly to the endpoint exposed by the SOAPInput node, this capability will continue to work even if the internal WSDL deployment details change in the future.

If your HTTP flow and your SOAP flow are using the same listener (either the broker listener or the same embedded listener), the HTTPInput URL will clash with the SOAPInput URL, and a warning is written to the event log. In this case the ?wsdl request is always serviced correctly, but web service requests could be sent to either the SOAPInput or the HTTPInput node.

Related concepts:



Configuring message flows by using a WSDL

You can use a WSDL in a variety of different ways to configure message flows.

Related reference:



SOAPInput node

Use the SOAPInput node to process client SOAP messages, so that the broker operates as a SOAP Web Services provider.



HTTPInput node

Use the HTTPInput node to receive an HTTP message from an HTTP client for processing by a message flow.

Migrating CMP applications

If you have applications that use the IBM Integration API (CMP), check that they access the correct resources.

Procedure

1. If you are migrating from WebSphere Message Broker Version 6.1, update your CMP applications to use the file that is supplied by IBM Integration Bus Version 9.0. You cannot use the ConfigManagerProxy.jar file from Version 6.1 to work with a Version 9.0 broker or to modify message flow files.

You do not have to change the methods that the application uses; the Version 9.0 IBM Integration API includes all deprecated classes and takes appropriate action.

- If you are migrating from Version 7.0 or Version 8.0, you can use the ConfigManagerProxy.jar file from Version 9.0 to connect to a Version 7.0 broker.
 - You can also continue to use the ConfigManagerProxy.jar file from Version 7.0 or Version 8.0 to connect to a Version 9.0 broker.
 - However, if you use the Version 9.0 file with your Version 7.0 or Version 8.0 application, you should recompile your application against the new file to take advantage of any deprecation warnings.
2. For CMP applications that connect to a broker, update the connection details that are used by your CMP applications to connect to the appropriate broker.

Related concepts:



The IBM Integration API

The IBM Integration API is a programming interface that your applications can use to control brokers and their resources through a remote interface.



The IBM Integration API samples

Explore the samples to learn the basic features that are provided by the IBM Integration API (also known as the CMP). Run the samples to deploy a BAR file or manage a broker, or use the IBM Integration API (CMP) Exerciser to implement various tasks.

Related tasks:



Developing applications that use the IBM Integration API

Develop Java applications that use the IBM Integration API (also known as the CMP) to communicate with, deploy to, and manage brokers and their associated resources.



Configuring an environment for developing and running CMP applications

Prepare the environment in which you want to run your CMP applications.

Updating error processing routines

Update your message processing to handle new or reused messages, and to delete obsolete messages.

About this task

IBM Integration Bus Version 9.0 components generate diagnostic messages (BIP messages), and no longer generate others that were reported in previous versions. The text of some messages is updated to reflect change in behavior and function in later versions, although the meaning is retained. One message, BIP8663, is reused and has different content and meaning.

If you have applications that check for specific messages, you must update these routines. For example, you might have programs that automate error reporting, or identify the occurrence of specific error conditions.

For details of the content of all messages that are generated by Version 9.0, see Diagnostic messages.

Version 7.0 generated the following new messages:

- **Configuration:** messages in the range 1000 - 1999:

BIP1014, BIP1054, BIP1060, BIP1115, BIP1117
BIP1181 to BIP1191
BIP1251 to BIP1269
BIP1280 to BIP1295
BIP1810

- **Broker:** messages in the range 2000 - 2999:

BIP2082, BIP2289, BIP2319, BIP2358
BIP2372, BIP2676, BIP2829, BIP2830
BIP2850 to BIP2866
BIP2870 to BIP2872
BIP2880 to BIP2883
BIP2890 to BIP2894

- **Built-in nodes:** messages in the range 3000 - 3999:

BIP3226
BIP3432 to BIP3439
BIP3451 to BIP3463
BIP3466 to BIP3471
BIP3524 to BIP3526
BIP3574, BIP3635, BIP3687
BIP3747 to BIP3760
BIP3769 to BIP3831
BIP3915 to BIP3925

- **Built-in nodes:** messages in the range 4000 - 4999:

BIP4015 to BIP4017
BIP4071 to BIP4076
BIP4251 to BIP4254
BIP4359
BIP4513 to BIP4516
BIP4678
BIP4821 to BIP4829
BIP4833 to BIP4865

- **Parsers:** messages in the range 5000 - 5999:

BIP5031, BIP5637

- **WebSphere MQ parsers:** messages in the range 6000 - 6999:

BIP6069, BIP6265, BIP6266

- **Publish/subscribe:** messages in the range 7000 - 7999:

BIP7036 to BIP7038
BIP7042 to BIP7045
BIP7099
BIP7120 to BIP7123

- **Commands:** messages in the range 8000 - 8999:

BIP8157, BIP8200
BIP8231 to BIP8239
BIP8769 to BIP8783
BIP8919 to BIP8923
BIP8934 to BIP8940
BIP8990 to BIP8993

- **z/OS:** messages in the range 9000 - 9999:

BIP9284
BIP9286 to BIP9290

Version 7.0 reused the message BIP8663.

Version 8.0 components no longer generate the following messages:

- **Configuration:** messages in the range 1000 - 1999:

BIP1002 to BIP1013
BIP1015 to BIP1016
BIP1020 to BIP1023
BIP1029, BIP1032, BIP1037, BIP1040
BIP1042 to BIP1045
BIP1055 to BIP1057
BIP1059, BIP1070
BIP1072 to BIP1088
BIP1090, BIP1094
BIP1096 to BIP1107
BIP1109 to BIP1112
BIP1126, BIP1128, BIP1131, BIP1132, BIP1145, BIP1146
BIP1151 to BIP1165
BIP1170 to BIP1173
BIP1201 to BIP1211
BIP1213 to BIP1217
BIP1221 to BIP1230
BIP1301 to BIP1306
BIP1350 to BIP1353
BIP1355 to BIP1363
BIP1401 to BIP1415
BIP1449, BIP1450, BIP1461, BIP1462
BIP1502 to BIP1528
BIP1530 to BIP1552
BIP1557 to BIP1571
BIP1581 to BIP1589
BIP1702 to BIP1703
BIP1705 to BIP1708
BIP1711 to BIP1712
BIP1751 to BIP1760
BIP1762, BIP1764, BIP1765
BIP1767 to BIP1773
BIP1775 to BIP1781
BIP1806 to BIP1807

- **Broker:** messages in the range 2000 - 2999:

BIP2012, BIP2013
BIP2025 to BIP2028
BIP2032, BIP2035, BIP2036, BIP2040
BIP2048, BIP2049, BIP2053
BIP2072, BIP2073, BIP2090
BIP2092 to BIP2098
BIP2220, BIP2221, BIP2223, BIP2224, BIP2317
BIP2814 to BIP2816

- **Built-in nodes:** messages in the range 4000 - 4999:

BIP4046

- **Parsers:** messages in the range 5000 - 5999:

BIP5315

- **Publish/subscribe:** messages in the range 7000 - 7999:

BIP7003, BIP7007, BIP7011, BIP7012

BIP7028, BIP7029, BIP7060, BIP7090

- **Commands:** messages in the range 8000 - 8999:

BIP8058, BIP8060, BIP8061, BIP8066, BIP8077, BIP8078

BIP8088, BIP8089, BIP8090

BIP8102, BIP8103, BIP8106, BIP8107

BIP8109, BIP8110, BIP8112

BIP8118, BIP8119, BIP8148, BIP8180

BIP8201 to BIP8210, BIP8214,

BIP8250 to BIP8263, BIP8286

BIP8301 to BIP8311

BIP8380 to BIP8382

BIP8390 to BIP8394

BIP8617, BIP8618, BIP8620

BIP8664 to BIP8668

BIP8672, BIP8674, BIP8675, BIP8678, BIP8795, BIP8796

BIP8829 to BIP8832

BIP8834 to BIP8839

BIP8842 to BIP8845

BIP8925

BIP8930 to BIP8933

BIP8995

- **z/OS:** messages in the range 9000 - 9999:

BIP9144 to BIP9146

BIP9801 to BIP9842

Version 9.0 components no longer generate the following messages:

- **Parsers:** messages in the range 5000 - 5999:

BIP5002

Version 9.0 components generate the following new messages:

- **Configuration:** messages in the range 1000 - 1999:

BIP1037, BIP1055, BIP1145,

BIP1838 to BIP1870

BIP1875I to BIP1876

BIP1880

BIP1884 to BIP1898

BIP1900I to BIP1901

- **Broker:** messages in the range 2000 - 2999:

BIP2018 to BIP2021

BIP2146 to BIP2149

BIP2156

BIP2158 to BIP2159

BIP2398
BIP2831 to BIP2839
BIP2867
BIP2898 to BIP2899

- **Built-in nodes:** messages in the range 3000 - 3999:

BIP3182 to BIP3191
BIP3305
BIP3644 to BIP3660
BIP3667 to BIP3670
BIP3688 to BIP3690
BIP3832 to BIP3834
BIP3841 to BIP3853
BIP3966 to BIP3977

- **Built-in nodes:** messages in the range 4000 - 4999:

BIP4305 to BIP4310
BIP4319 to BIP4329
BIP4792
BIP4866 to BIP4869
BIP4978
BIP4993 to BIP4999

- **Parsers:** messages in the range 5000 - 5999:

BIP5000
BIP5050 to BIP5052
BIP5847 to BIP5849

- **WebSphere MQ parsers:** messages in the range 6000 - 6999:

BIP6123 to BIP6124
BIP6277

- **Publish/subscribe:** messages in the range 7000 - 7999:

BIP7127 to BIP7149
BIP7152 to BIP7199
BIP7413
BIP7566 to BIP7582
BIP7975 to BIP7995

- **Commands:** messages in the range 8000 - 8999:

BIP8251 to BIP8252
BIP8287 to BIP8297
BIP8500 to BIP8507
BIP8955
BIP8965 to BIP8973
BIP8994

- **z/OS:** messages in the range 9000 - 9999:

BIP9261
BIP9279 to BIP9280
BIP9844

- **Activity Log:** messages in the range 11000 - 12999:

BIP11051 to BIP11202
BIP12062 to BIP12091

Related tasks:

Chapter 3, “Migrating,” on page 171

To migrate a broker domain to IBM Integration Bus Version 9.0, plan your migration strategy, perform pre-migration tasks, migrate your domain components, then complete post-migration tasks.

Chapter 4. Configuring brokers for development environments

Set up application development environments on Linux on x86 or Windows to create, test, and deploy message flows and associated resources.

About this task

If you are new to IBM Integration Bus, you can set up a basic development environment on either Windows or Linux on x86 by installing the Integration Bus component and the IBM Integration Toolkit, and creating a default configuration. Run the Default Configuration wizard to create a broker, and try the sample programs that are supplied in the IBM Integration Toolkit.

Some of the samples depend on the default configuration, which creates a broker that has the fixed name, IB9NODE, and a queue manager for the broker that has the fixed name, IB9QMGR.

Details of the default configuration are provided in “Creating a default configuration” on page 274.

When you have tried out your first few samples, you can create and configure more brokers, and set up an environment that can support your application developers. Many samples are supplied, and provide education and guidance about how to use many of the facilities of IBM Integration Bus.

See the full list of samples in Samples. The samples are grouped in sections according to the features that they use, and the operations that they perform. For example, the section “Web services samples” lists the samples that use Web services nodes to communicate with WebSphere Service Registry and Repository (WSRR) and other Web applications.

You can view information about samples only when you use the information center that is integrated with the IBM Integration Toolkit or the online information center. You can run samples only when you use the information center that is integrated with the IBM Integration Toolkit.

To create and configure brokers in addition to, or instead of, the default configuration, follow the instructions in “Creating a development environment” on page 277.

When you start the IBM Integration Toolkit, a workbench session opens, which you can use to create, configure, and manage your application development resources. You can configure your workbench session in various ways to suit your working environment and preferences. These options are described in “Configuring the Toolkit” on page 280.

Related tasks:

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.



Administering brokers and broker resources

Administering brokers and associated integration node resources includes the tasks that you perform frequently to activate and manage those resources. Choose the method you prefer to administer your brokers and associated resources.



Developing integration solutions

IBM Integration Bus provides a flexible environment in which you can develop integration solutions to transform, enrich, route, and process your business messages and data. You can integrate client applications that use different protocols and message formats.

Creating a default configuration

Use the Default Configuration wizard to create and test a basic broker configuration.

Before you begin

Before you run the Default Configuration wizard, complete the following steps:

- Install the IBM Integration Toolkit, the Integration Bus component, and the Integration Bus component prerequisite products on this computer.
- Access the Default Configuration wizard through the IBM Integration Toolkit, which is available only on Linux on x86 and Windows.

About this task

By using the Default Configuration wizard, you create a default configuration on your local computer so that you can explore the product and run the samples. You can also remove the default configuration, if it exists, that has been created on your logon account.

- “Creating the default configuration”
- “Removing the default configuration” on page 275

The default configuration is described in more detail in “Verifying your IBM Integration Bus installation” on page 104. This topic describes how to verify your installation on Linux on x86, Linux on x86-64, or Windows by using either the IBM Integration Toolkit or the IBM Integration Explorer.

Creating the default configuration

About this task

The wizard creates the following resources:

- A sample broker named IB9NODE.
- A WebSphere MQ queue manager named IB9QMGR.

Procedure

1. Start the Default Configuration wizard from the IBM Integration Toolkit Welcome page, which is displayed the first time you start the IBM Integration Toolkit. If the Welcome page is not displayed, open it in the IBM Integration Toolkit by clicking **Help > Welcome**.
2. Click **Get Started** on the Welcome page, then click **Create the Default Configuration**.

3. Click **Start the Default Configuration wizard**. The Default Configuration wizard is displayed.
4. The Welcome page of the wizard describes what is about to happen, click **Next** to continue. You can click **Cancel** at any time to cancel the creation of the default configuration.
The wizard checks that the default configuration is not already created.
5. The Default Configuration Summary page lists the resources that will be created. Click **Next** to continue.
6. The Default Configuration Progress page lists the background configuration actions as they occur, and indicates successful completion. You can cancel the creation of the default configuration by clicking **Cancel**. The wizard backs out all the configuration tasks and displays the progress and success of the process. The configuration process is written to a log file in the Eclipse workspace directory:

- **Linux** /home/*user_name*/IBM/IntegrationToolkit90/workspace/.metadata/DefaultConfigurationWizard.log
- **Windows** C:\Documents and settings*user_name*\IBM\IntegrationToolkit90\workspace\.metadata\DefaultConfigurationWizard.log

If the default configuration is created successfully, you see an appropriate message. If errors occur, you see an appropriate message and the wizard backs out all the configuration tasks. If an error occurs during the back out process, the wizard displays a list of resources that you must remove manually.

7. You can use the samples to verify the default configuration. **Launch Samples Wizard when finished** is selected by default. Click **Finish** to open the Prepare the Samples wizard.

If you do not want to open the Prepare the Samples wizard, clear **Launch Samples Wizard when finished** before clicking **Finish**.

If you are viewing this information from within the IBM Integration Toolkit, you can open the samples manually by clicking the following sample:

- Pager

Alternatively, click **Product Overview > Samples** in the IBM Integration Bus information center in the IBM Integration Toolkit, to display a list of the available samples.

You can view information about samples only when you use the information center that is integrated with the IBM Integration Toolkit or the online information center. You can run samples only when you use the information center that is integrated with the IBM Integration Toolkit.

Removing the default configuration

Procedure

1. Start the Default Configuration wizard from the IBM Integration Toolkit Welcome page, which is displayed after you start the IBM Integration Toolkit. If the Welcome page is not displayed, open it in the IBM Integration Toolkit by clicking **Help > Welcome**.
2. Click **Get Started** on the Welcome page, then click **Create the Default Configuration**.
3. Click **Remove the Default Configuration wizard**. The Default Configuration wizard is displayed.

4. The Welcome page of the wizard describes what is about to happen. You can click **Cancel** at any time to cancel the removal of the default configuration. The wizard checks that the default configuration is already created.
5. The Remove Default Configuration Summary page lists the resources that will be removed. Click **Next** to continue.
6. The Default Configuration Progress page lists the removal actions as they occur, and indicates successful completion. The removal process is written to a log file in the Eclipse workspace directory:
 - **Linux** /home/user_name/IBM/IntegrationToolkit90/workspace/.metadata/DefaultConfigurationWizard.log
 - **Windows** C:\Documents and settings\user_name\IBM\IntegrationToolkit90\workspace\.metadata\DefaultConfigurationWizard.log
7. A message confirms that the default configuration has been removed successfully. Click **Finish** to close the wizard.

If errors occur during the removal of the default configuration, the wizard displays the errors, and also writes them to the log file. Follow the advice in the log, and try each step again.

Results

If you experience problems when you are using the wizard to remove the default configuration, you might have to remove the default configuration manually. For more information, see [You experience problems with the default configuration](#).

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.



Samples

The IBM Integration Toolkit provides samples that show the features that are available in IBM Integration Bus, and how to use them. This topic provides links to the information about the individual samples.

Related tasks:

“Creating a development environment” on page 277

Create a development environment on Linux on x86 or Windows to develop your message flows.

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.



Resolving problems that occur when you start resources

Use the advice given here to help you to resolve common problems that can occur when you start resources.



Resolving problems when running samples

Use the advice given here to help you to resolve common problems that can arise when you run or remove samples.

Related reference:

“Installation Guide” on page 32

Installation information for IBM Integration Bus is provided in the online information center.

“WebSphere MQ resources for brokers” on page 17

Each broker depends on a number of WebSphere MQ resources: some are required, others are optional, and depend on your environment and requirements. Some of these resources are created for you, but others you must define for yourself.

Creating a development environment

Create a development environment on Linux on x86 or Windows to develop your message flows.

Before you begin

Before you start:

- “Installing the Integration Bus component” on page 69
- “Installing the IBM Integration Toolkit” on page 79

On Windows, you can use the Launchpad to install both these components. For details, see “Installing by using the Windows Launchpad” on page 60.

If you are new to IBM Integration Bus, and have not created a default configuration, you might want to use that option to set up your first configuration. You can use the default environment to run some of the supplied sample programs, and explore the features that are supported by IBM Integration Bus. Follow the instructions in “Creating a default configuration” on page 274.

About this task

To create your own development environment on a computer that is running Linux on x86 or Windows, complete the following steps:

Procedure

1. Start the IBM Integration Toolkit in one of the following ways:

- **Linux** From the main menu:
 - On Red Hat, click **Programming > IBM Integration Toolkit**.
 - On SUSE Linux, click **All Applications > IBM Integration Toolkit**.
- **Windows** Click **Start > All Programs > IBM Integration Toolkit > IBM Integration Toolkit 9.0 > IBM Integration Toolkit 9.0**, or double-click the shortcut on your desktop.
- Use the following commands in a command prompt from their location in the root directory for the package group:
 - **Linux**

```
./eclipse -product com.ibm.etools.msgbroker.tooling.ide
```
 - **Windows**

```
eclipse.exe -product com.ibm.etools.msgbroker.tooling.ide
```

A new session of the IBM Integration Toolkit opens, and prompts you for the name of a workspace in which to store the resources that you create in this session.

You can accept the default value, workspace, type your preferred name into the text field, or click Browse to select the location of your choice. Select **Use this as the default and do not ask again** to inhibit the display of the workspace dialog next time you start the IBM Integration Toolkit.

2. Click **OK** to confirm the location of your workspace. Your IBM Integration Toolkit session opens, and displays the Welcome page. You can tailor various settings in the IBM Integration Toolkit to suit your requirements and your working environment; for example, the colors and fonts. These options are described in “Configuring the Toolkit” on page 280.
3. Close the Welcome page. You can return to the Welcome page later by clicking **Help > Welcome**. If more than one option is listed, select IBM Integration Bus.
4. In the Integration Development perspective, select the Integration Nodes view. Alternatively, click **Window > Show view > Other > Integration Development > Integration Nodes** to show the Integration Nodes view.
5. Right-click **WebSphere Message Brokers**, and click **New > Local Broker** to create a broker, and complete the wizard for this task.

If you want more detailed instructions, see “Creating a broker for a development environment” on page 279.

Your configuration might be affected by the operation mode of your broker. The default operation mode is advanced mode, in which your broker runs with no restrictions. Check with your broker administrator which operation mode applies to your organization; you might have to change the mode of your broker after you have created it. For more information about operation modes, see “Operation modes” on page 5.

6. You must create at least one integration server on your broker. To add additional integration servers to your broker, right-click the broker, and click **New > Integration Server**. Enter a name for your integration server, and click **OK**.

The integration server is the runtime environment in which your message flows run. You can create many integration servers on a single broker, and you can deploy your message flows to one or more integration servers on one or more brokers.

For more detailed instructions about this task, see Creating an integration server using the IBM Integration Toolkit or IBM Integration Explorer.

Results

Your configuration is now ready to use.

What to do next

Next: You can start to develop resources to deploy to your broker. You can create your own message flows, or you can use the patterns and samples that are provided to get you started. For details of these options, see Developing integration solutions.

This task has covered the minimum set of steps that you must complete to create a broker and configure your development environment. Typically, as an application developer, you are working in a single platform environment to create, deploy, and test your message flows before they are ready for use in a test of production environment.

More options are available for enhanced configurations, including configurable services. When the requirements of your message flows extend beyond this basic

configuration, and you need additional configuration to support those requirements, you can find details of these more advanced options in Chapter 5, “Configuring brokers for test and production environments,” on page 289.

Related tasks:

Chapter 4, “Configuring brokers for development environments,” on page 273
Set up application development environments on Linux on x86 or Windows to create, test, and deploy message flows and associated resources.

“Creating a default configuration” on page 274

Use the Default Configuration wizard to create and test a basic broker configuration.

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.



Administering brokers and broker resources

Administering brokers and associated integration node resources includes the tasks that you perform frequently to activate and manage those resources. Choose the method you prefer to administer your brokers and associated resources.



Developing integration solutions

IBM Integration Bus provides a flexible environment in which you can develop integration solutions to transform, enrich, route, and process your business messages and data. You can integrate client applications that use different protocols and message formats.

Creating a broker for a development environment

Create an integration node (broker) by using the IBM Integration Toolkit on Linux on x86 or Windows.

Before you begin

Before you start:

If you are using the IBM Integration Explorer, see “Creating an integration node using the IBM Integration Explorer” on page 311.

- On Windows, you must have administrator access rights to create integration nodes by using the IBM Integration Toolkit or IBM Integration Explorer. When creating an integration node, you might be prompted to agree to the use of administrator rights or be prompted to enter an administrator user ID and password.
- If you want to configure the integration node as a WebSphere MQ trusted application, see “Using WebSphere MQ trusted applications” on page 306.
- Read Considering security for a broker.
- Check which operation mode you are licensed to use. If you do not set a mode, the automatic default is advanced mode; see “Operation modes” on page 5.

About this task

When you create a broker, if the WebSphere MQ queue manager does not exist, the queue manager is automatically created. If WebSphere MQ Version 7.1, or later, has been selected for the queue manager, the channel auth security is automatically disabled.

If the specified queue manager already exists when creating the broker, then it is assumed that the user has applied the appropriate security configuration to meet their requirements, and therefore channel auth security is not disabled.

To create a broker by using the IBM Integration Toolkit:

Procedure

1. In the Integration Development perspective, select the Integration Nodes view.
2. Right-click the Integration Nodes folder, and click **New > Local Integration Node**. The New Local Integration Node wizard is displayed.
3. Enter a name for the broker and queue manager. You cannot create a broker on a queue manager that is already associated with a broker.
4. On Windows, enter your user name and password. These parameters are not required on Linux on x86.
5. Optional: On Windows, select whether to start the broker automatically when Windows starts.
6. Enter a name for the default integration server, or accept the value “default”.
7. Optional: Enter a value for the flow debug port for the default integration server.
8. Click **Finish**.

Results

You have created and started a broker.

What to do next

Return to the instructions in “Creating a development environment” on page 277.

Related concepts:



IBM Integration Toolkit

The IBM Integration Toolkit is an integrated development environment and graphical user interface based on the Eclipse platform.



IBM Integration Explorer

The IBM Integration Explorer is a graphical user interface based on the Eclipse platform for administering your integration nodes (brokers).

Related tasks:

Chapter 4, “Configuring brokers for development environments,” on page 273
Set up application development environments on Linux on x86 or Windows to create, test, and deploy message flows and associated resources.

“Creating a broker” on page 304

You can create brokers on every platform that is supported by IBM Integration Bus. The broker runs as a 64-bit application on all platforms except Linux on x86 and Windows on x86.

“Creating a default configuration” on page 274

Use the Default Configuration wizard to create and test a basic broker configuration.

Configuring the Toolkit

You can configure various settings in the Toolkit to suit your requirements and your working environment.

About this task

The following topics show you how to configure aspects of the Toolkit:

Procedure

- “Changing IBM Integration Toolkit preferences”
- “Changing workbench capabilities” on page 282
- “Changing IBM Integration Explorer preferences” on page 356
- “Configuring CVS to run with the IBM Integration Toolkit” on page 283
- “Configuring the IBM Integration Toolkit to run Rational ClearCase” on page 284
- “Creating a working set” on page 285
- “Integrating the Rational Team Concert client with the IBM Integration Toolkit” on page 287

Results

A minimum display resolution of at least 1024 x 768 is required for some dialog boxes, such as the Preferences dialog box.

Related concepts:



IBM Integration Toolkit

The IBM Integration Toolkit is an integrated development environment and graphical user interface based on the Eclipse platform.

Related reference:



IBM Integration Toolkit

The IBM Integration Toolkit provides your application development environment on Windows and Linux on x86.

Changing IBM Integration Toolkit preferences

The IBM Integration Toolkit has a large number of preferences that you can change to suit your requirements. Some of these are specific to the product plug-ins that you have installed, including those for IBM Integration Bus. Others control more general options, such as the colors and fonts in which information is displayed.

About this task

To access the IBM Integration Toolkit preferences:

Procedure

1. Select **Window > Preferences**.
2. Click the plus sign associated with **General**, typically the first entry in the left pane. An expanded list of options appears. Select the aspect of the IBM Integration Toolkit that you want to modify. These options might be of interest:

Startup and shutdown

Switch on, or off, the prompt at toolkit startup that asks you to confirm the workspace location. Typically you switch this prompt off, so that it does not appear, but you can force it to appear next time you start the IBM Integration Toolkit if you want to specify a different location.

You can also specify whether to display the dialog box that asks you to confirm shutdown of the IBM Integration Toolkit.

Appearance

Change the default fonts and colors that appear in the IBM Integration Toolkit.

Perspectives

On this dialog, your choices include the option to open a new perspective in a new window.

3. When you have made your changes, click **OK** to close the Preferences dialog.

What to do next

Below the General category in the Preference dialog are items that refer specifically to IBM Integration Bus resources, such as message flows. Review the following topics for information about setting preferences and other values that are specific to your use of these resources:

- Message flow preferences
- Changing ESQL preferences
- Message Sets: Configuring message set preferences
- Testing and debugging message flows
- Changing trace settings

Related concepts:



IBM Integration Toolkit

The IBM Integration Toolkit is an integrated development environment and graphical user interface based on the Eclipse platform.

Related reference:



IBM Integration Toolkit

The IBM Integration Toolkit provides your application development environment on Windows and Linux on x86.

Changing workbench capabilities

You can configure the workbench to disable access to some of the functional capabilities of IBM Integration Bus.

About this task

Capabilities is an Eclipse concept that allows you to enable or disable the components of a product. By default, all components of the workbench are enabled.

To access the workbench capabilities:

Procedure

1. Select **Window > Preferences**.
2. Click the plus sign associated with **General**. An expanded list of options appears.
3. Click **Capabilities**. You can use the capabilities that are listed to enable or disable various product components; the capabilities are grouped according to a set of predefined categories.
4. Select **IBM Integration Toolkit** from the list of capabilities that is displayed, and select the **Advanced** button. A window opens that has a check box for each of the predefined categories.

5. Select the check boxes for the categories that you want to either enable or disable; click either the **Enable All** or **Disable All** button and click **OK**. A pane describes the functionality that is enabled following this action.

What to do next

The predefined categories for the workbench are listed together with a reference to more information about the relevant functional area of IBM Integration Bus:

- **IBM Integration Toolkit - Administration.** See Administering brokers and broker resources.
- **IBM Integration Toolkit - Core.** See IBM Integration Toolkit.
- **IBM Integration Toolkit - Development.** See Developing integration solutions.

Related concepts:



IBM Integration Toolkit

The IBM Integration Toolkit is an integrated development environment and graphical user interface based on the Eclipse platform.

Related tasks:



Developing integration solutions

IBM Integration Bus provides a flexible environment in which you can develop integration solutions to transform, enrich, route, and process your business messages and data. You can integrate client applications that use different protocols and message formats.

Related reference:



IBM Integration Toolkit

The IBM Integration Toolkit provides your application development environment on Windows and Linux on x86.

Configuring CVS to run with the IBM Integration Toolkit

Install CVS as a normal program by following the usual prompts. Not all versions of *CVSNT* are supported by Eclipse.

Procedure

1. Configure CVS by carrying out the following tasks:
 - a. Create a directory on your computer, for example, on Windows - `c:\CVSRepository`.
 - b. Start the *CVSNT* control panel. Select **Start > Programs > CVSNT** to see the icon on the desktop.
 - c. Stop both the CVS Service and the CVS Lock Service.
 - d. Select the Repositories tag, click **Add** and create a new repository. Note that no entry appears on the screen the first time that you do this.
 - e. Use the **...** button on the next window to select the directory that you created in step 1a and click **OK**. Note that when CVS has finished formatting its repository the backslash in the directory name is changed to a forward slash.
 - f. Select the Service Status tab and restart both the CVS Service and the CVS Lock Service.
2. Enable the CVS Revision tag to be populated in the Eclipse Version fields in the IBM Integration Toolkit. To do this on Windows:
 - a. Select Window->Preferences

- b. Expand the Team section and click **CVS**
 - c. Use the drop down in the **Default keyword substitution:** field and set the value to ASCII with keyword expansion(-kkv)
3. Add the IBM Integration Bus file types to the Eclipse CVS configuration. To do this:
 - a. Select **File Content** in the Team section of the window you opened in step 2 on page 283
 - b. Click **Add** and add msgflow as an allowable file extension. Ensure that the value is set to ASCII.
 - c. Repeat the above procedure for the following file extensions that the broker uses:
 - esql
 - mset
 - mxsd

If you use CVS to store other file types, for example, COBOL copybooks add the appropriate file types as well.
 - d. Click **OK** when you have finished.

Related concepts:



Version and keyword information for deployable objects

Use the Broker Archive editor to view the version and keyword information of deployable objects.

Related reference:



Description properties for a message flow

The description properties for a message flow include the Version, Short Description and Long Description. To view and edit the properties of a message flow click **Flow > Properties**.



Message flow preferences

You can change preferences that determine properties of message flows when you create them.



Message Sets: Message set preferences

Preferences for message sets.

Configuring the IBM Integration Toolkit to run Rational ClearCase

To use Rational ClearCase[®] with the IBM Integration Toolkit, enable the capability in the Preferences page.

About this task

To enable Rational ClearCase in the IBM Integration Toolkit:

Procedure

1. Click **Window > Preferences** to open the Preferences window.
2. Expand **General** in the left pane, and click **Capabilities**.
3. In the Capabilities pane, click **Advanced**. The Advanced window opens.
4. Expand **Team**, and ensure that **ClearCase SCM Adapter** is selected.
5. Click **OK** to close the Advanced window.

6. Click **OK** or **Apply** to apply your changes.

Results

After you enable the ClearCase capability, the ClearCase menu is displayed in the Integration Development perspective.

What to do next

To work with ClearCase:

1. Click **ClearCase > Connect to Rational ClearCase**.
2. Right-click your project and click **Team > Add to Version Control** to add your projects to the ClearCase source control.
3. After you have added your projects to the ClearCase source control, you can perform ClearCase operations.

Related concepts:



Development repository

Use a development repository to benefit from features such as version control and access control of files, which make it easier for teams to work on shared resources.



Version and keyword information for deployable objects

Use the Broker Archive editor to view the version and keyword information of deployable objects.

Related tasks:

“Configuring CVS to run with the IBM Integration Toolkit” on page 283
Install CVS as a normal program by following the usual prompts. Not all versions of *CVSNT* are supported by Eclipse.

Related reference:



Message flow preferences

You can change preferences that determine properties of message flows when you create them.



Message Sets: Message set preferences

Preferences for message sets.

Creating a working set

Create a working set to limit the number of resources that are displayed in the Application Development view.

Before you begin

Before you start:


To read about working sets, see *Working sets*.

About this task

By creating and using a working set, you can reduce the visual complexity of what is displayed in the Application Development view, making it easier to manage and work with your projects.

To create a new working set, complete the following steps:

Procedure


1. Click the arrow  on the toolbar of the Application Development view, then click **Select Working Set**.
A list is displayed containing existing working sets. A working set is created automatically when you create an application or library. The names of these working sets are enclosed by brackets; for example [Application1].
2. To open the New Working Set wizard, click **New**.
3. Select a type for the new working set (for example, Broker), then click **Next**.
4. Enter a name for the working set.
5. Select the resources that you want to include in this working set. You can also include all the projects that are dependent on your selected resources by selecting **Automatically include dependent projects in this working set**.
6. Click **Finish**.

Results

The new working set is created.

What to do next

Next:

- To show only the resources in a working set, click the arrow on the toolbar of the Application Development view, click **Select Working Set**, then select the relevant working set. If you select the working set that is created automatically for an application or library (as indicated by brackets; for example, [Application1]), the Application Development view shows only the selected container and its contents.
- To edit the selected working set, click the arrow on the toolbar of the Application Development view, then click **Edit Active Working Set**. When you delete the active working set, all resources are shown automatically. You cannot edit the working sets that are created automatically for an application or library, as indicated by brackets; for example, [Application1].
- To delete the selected working set, click the arrow on the toolbar of the Application Development view, click **Select Working Set**, then click **Remove**.
- To show all resources, click the arrow on the toolbar of the Application Development view, then click **Deselect Working Set**. Alternatively, click the **Remove all filters** icon  on the toolbar of the Application Development view.

Related concepts:



Resources

The projects, folders, and files that you work with in the IBM Integration Toolkit workspace are called *resources*. By default, these resources are stored with their metadata in the workspace directory in your local file system. The workspace directory is created the first time that you start the IBM Integration Toolkit.

Related tasks:



Creating an application

Create an application to contain all the resources that are required to develop your solution.



Creating a library

Create a library to contain a logical grouping of related code, data, or both, that can be reused.



Creating an application based on an existing message set

Create a new application that is based on an existing message set.

Related reference:



Integration Development perspective

The Integration Development perspective is the default perspective that is displayed when you start the IBM Integration Toolkit.

Integrating the Rational Team Concert client with the IBM Integration Toolkit

How you integrate the Rational Team Concert™ client with the IBM Integration Toolkit.

About this task

To integrate the Rational Team Concert client with the IBM Integration Toolkit, you must have the Rational Team Concert client in the same package group as the IBM Integration Toolkit.

The supported version of Rational Team Concert is V3.0.

Procedure

Use the Installation Manager to install the Rational Team Concert client into the same package group as the IBM Integration Toolkit.

Using the Installation Manager adds all the Rational Team Concert client capability into the IBM Integration Toolkit workspace.

You can also install the IBM Integration Toolkit into the Rational Team Concert client package group.

If the products are in the same package group, they can be installed in any order.

What to do next

To work with Rational Team Concert source control:

1. Start the IBM Integration Toolkit.
2. Open the Work Items perspective and click **Window > Open Perspective > Other > Work Items**.
3. Click the **Create Repository Connection** link in the Team Artifacts view.
4. Follow the dialog and enter the information given to you by your Jazz™ administrator.
5. To add your project to the repository, right-click the project and select **Team > Share Project**.
6. In the Share Project window, select Jazz Source Control as the repository type.

See Getting started in your Rational Team Concert source control workspace for more details about the Rational Team Concert source control operations.

You can also use the Jazz Team Build as your build engine for the IBM Integration Toolkit; see Building with Jazz Team Build for more information.

Related reference:



Message flow preferences

You can change preferences that determine properties of message flows when you create them.



Message Sets: Message set preferences

Preferences for message sets.

Chapter 5. Configuring brokers for test and production environments

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

About this task

On Linux on x86 and Windows, you can install the broker and the IBM Integration Toolkit, and create an initial configuration to learn about IBM Integration Bus, and start to develop your applications. You can also configure an environment for more advanced application development and unit test. These tasks are described in Chapter 4, “Configuring brokers for development environments,” on page 273.

To configure your test and production environments, use the information in this section to plan and configure the resources that you want:

Procedure

1. Plan the system.
2. If you are configuring a broker on z/OS, Customize the z/OS environment.
3. Ensure that you have the correct authorization and permissions to create and access components. For more information, see Authorization for configuration tasks and Integration Bus server security.
4. Create the components.
5. If you are using the IBM Integration Toolkit or IBM Integration Explorer, Configure brokers in the developer or administrator workbench.
6. If you want to ensure that you are using the correct operation mode for your license, Check the operation mode of your broker.
7. Create and configure the databases.
8. If you want to ensure the data integrity of transactions, Configure global coordination of transactions.
9. If you want to connect to external resources such as Enterprise Information Systems, IMS™, or JMS, Configure properties to connect to external resources.
10. If you want to configure the storage of events for aggregation, Collector, or timeout nodes, or configure monitoring event sources, Configure internal resources that are required by message flows.
11. If you want to view objects in a different language or code page, Change the locale.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.



Integration servers

An integration server is a named grouping of message flows that have been assigned to a broker. The broker enforces a degree of isolation between message flows in distinct integration servers by ensuring that they run in separate address spaces, or as unique processes.



Authorization for configuration tasks

Authorization is the process of granting or denying access to a system resource.

Related tasks:



Administering brokers and broker resources

Administering brokers and associated integration node resources includes the tasks that you perform frequently to activate and manage those resources. Choose the method you prefer to administer your brokers and associated resources.



Integration Bus server security

You must consider several security aspects when you are setting up brokers running on Windows, Linux, z/OS, or UNIX platforms.



Developing applications that use the IBM Integration API

Develop Java applications that use the IBM Integration API (also known as the CMP) to communicate with, deploy to, and manage brokers and their associated resources.

Customizing the z/OS environment

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

About this task

Although you might be installing only one broker initially, you might want to consider how the product will be used in your organization in a few years time. Planning ahead makes developing your IBM Integration Bus configuration easier.

The following rules apply to the configuration:

- A broker requires access to a queue manager.
- A broker cannot share its queue manager with another broker.
- You cannot use WebSphere MQ shared queues to hold data related to IBM Integration Bus as `SYSTEM.BROKER` queues, but you can use shared queues for your message flow queues.

You can find details of the WebSphere MQ queues that are created and used by IBM Integration Bus on z/OS in `mqsicreatebroker` command.

When planning to work in a z/OS environment, you must complete the following tasks:

- Create started task procedures for each broker that you plan to use. These procedures must be defined, in the started task table, with an appropriate user ID.
- Decide on your recovery strategy. As part of your systems architecture, you must have a strategy for restarting systems if they end abnormally. Common solutions are to use automation products like Tivoli NetView® for z/OS or the Automatic Restart Manager (ARM) facility. You can configure IBM Integration Bus to use ARM.
- Plan for corequisite products, including UNIX System Services, Resource Recovery Services (RRS), WebSphere MQ, and Java.

You might also want to include DB2 in your configuration, if your message flows access databases.

- Collect broker statistics on z/OS.

See the following topics for more information:

- “Customizing UNIX System Services on z/OS”
- “Resource Recovery Service planning on z/OS” on page 295
- “Configuring WebSphere MQ on z/OS” on page 295
- “Defining the started tasks to z/OS Workload Manager (WLM)” on page 296
- “Automatic Restart Manager planning” on page 297
- “Mounting file systems” on page 298
- “Checking the permission of the installation directory” on page 300
- “Customizing the version of Java on z/OS” on page 301
- “Checking APF attributes of bipimain on z/OS” on page 301
- “Configuring an integration server address space as non-swappable on z/OS” on page 302

For an overview of how to create IBM Integration Bus brokers, see “Creating a broker on z/OS” on page 313. To verify your configuration, see “IBM Integration Bus and WebSphere MQ setup verification” on page 329.

Related concepts:



Overview of WebSphere MQ data processing on z/OS shared queues
Some messaging transactions depend on the exact sequence of messages from a queue, and for that sequence to be maintained in the event of a failure of the queue manager. In these instances you must serialize the access to those messages.

Related tasks:

“Creating a broker on z/OS” on page 313
Create the broker component and the other resources on which it depends.
“Collecting broker statistics on z/OS” on page 329



Using WebSphere MQ shared queues for input and output (z/OS)
On z/OS systems, you can define WebSphere MQ shared queues as input and output queues for message flows. You might need to serialize access to those messages.

Related reference:



mqsicreatebroker command
Use the **mqsicreatebroker** command to create a broker and its associated resources.

Customizing UNIX System Services on z/OS

IBM Integration Bus requires the setup of some UNIX System Services system parameters.

About this task

This task is part of the larger task of customizing your z/OS environment.

You can use the **SETOMVS** operator command for dynamic changes or the **BPXPRMxx** PARMLIB member for permanent changes. See the *z/OS UNIX System Services* section in the *z/OS V1R9.0 LibraryCenter*.

Use the **DISPLAY OMVS,OPTIONS** command to display your current OMVS options.

Do not include the broker addresses if you use the IEFUSI exit to limit the region size of OMVS address spaces.

Set the UNIX System Services parameters that are shown in the following table.

Description	Parameter	Value
The maximum core dump file size (in bytes) that a process can create. Allow an unlimited size.	MAXCORESIZE	2 147 483 647
The CPU time (in seconds) that a process is allowed to use. Allow an unlimited CPU time. If you do not set MAXCPU TIME to the maximum value, you can set it on the TIME parameter of the broker started task JCL. For more information, see <i>Statements and parameters for BIPXPRMxx</i> in the <i>MVS™ Initialization Tuning and Reference of the z/OS V1R9.0 LibraryCenter</i> .	MAXCPU TIME	2 147 483 647
The address space region size. Set to the size of the biggest address space.	MAXASSIZE	> 1 073 741 824 A minimum value of 393 216 000 bytes is required.
The maximum number of processes for each user.	MAXPROCUSER	Set to a number greater than all brokers and associated integration servers, as well as any queue managers and channel initiators that are running with the same user ID.
The maximum number of file descriptors that a single process can have open concurrently.	MAXFILEPROC	The number of descriptors includes all open files, directories, sockets, and pipes. A minimum value of 65536 is required.
Specifies the maximum number of threads that a single process can have active. Depends on the definitions of message flows.	MAXTHREADS MAXTHREADTASKS	The value of MAXTHREADS and MAXTHREADTASKS depends on your application. To calculate the value that is needed for each message flow: 1. Multiply the number of input nodes by the number of instances (extra threads +1). 2. Sum the values of all message flows, and then add 10 to the sum. 3. Add to the sum the number of threads that are used for each HTTP listener.
Specifies the maximum number of data space pages that can be allocated for memory mapping of z/OS UNIX files. This space is used by the Integration registry.	MAXMMAPAREA	Each integration node (broker) on the image requires 55000 pages. If you have a large number of policies or service definitions, you might need to increase this value. MAXSHAREPAGES controls the maximum number of shared pages to be used for fork, shared memory, memory map files, and ptrace so you might need to adjust this value to accommodate adjustments to MAXMMAPAREA.

Deploying a message flow that starts an integration server in a new address space uses z/OS UNIX System Services Semaphore and SharedMemorySegment resources. Each new address space uses a semaphore and SharedMemorySegment. The SharedMemorySegment is deleted immediately after the new address space starts, but the semaphore remains for the life of the new address space.

Certain z/OS UNIX System Services system parameters can affect the start of a new integration server address space, if you set them incorrectly. These parameters include:

- IPCSEMNIDS
- IPCSHMNIDS
- IPCSHMNSEGS

You must have a minimum of three semaphores for each integration server address space that is started.

You must set IPCSEMNIDS to a value four times the number of potential integration server address spaces on a system.

You must have one SharedMemorySegment for each integration server address space that is started. You must set IPCSEMNIDS to a value that exceeds the number of potential integration server address spaces on a system.

A control address space (BIPSERVICE and BIPBROKER processes) can be attached to many SharedMemorySegments - potentially, one for each integration server address space started for that broker. You must set IPCSHMNSEGS to a value that exceeds the potential number of integration servers for each broker.

Ensuring sufficient space for temporary files

The environment variable TMPDIR is the path name of the directory that is used for temporary files. If it is not set, the z/OS shell uses /tmp.

When IBM Integration Bus components start, sufficient space is required in the directory that is referenced by TMPDIR. In particular, Java requires sufficient space to hold all JAR files that are required by IBM Integration Bus.

If you do not allocate sufficient space, the integration server address spaces abend with a 2C1 code.

Allow at least 50 MB of space in this directory for IBM Integration Bus components. More space might be needed if you deploy large user-defined nodes or other JAR files to the broker component.

Defining IBM Integration Bus files as shared-library programs

The shared library region is a z/OS feature that enables address spaces to share dynamic link library (DLL) files. This feature enables brokers to share DLLs, rather than each broker having to load the DLLs individually. The shared DLLs can be broker DLLs, or DLLs from other products such as Java. The amount of real storage that is used by MVS, and the time it takes for each broker to start, can thus be greatly reduced.

If you plan to deploy to more than one integration server on z/OS, you can reduce the amount of storage that is required by the integration server address spaces. Set the shared-library extended attribute on the following files:

```
/usr/lpp/mqsi/bin/*  
/usr/lpp/mqsi/lil/*  
/usr/lpp/mqsi/lib/*  
/usr/lpp/mqsi/lib/wbimb/*
```

To set the shared-library attribute, use the OMVS **extattr** command with the +1 option. For example:

```
extattr +1 /usr/lpp/mqsi/bin/*
```

To find out whether the shared-library extended attribute is set, use the **ls -E** command. For example, use the command **ls -E bipimain** to generate the following response:

```
-rwxr-x--- a-l- 1 USER  GROUP   139264 Mar 15 10:05 bipimain
```

where l (lowercase L, as in a-l-) shows that the program is enabled to run in a shared address space.

For more information, see the z/OS V1R9.0 LibraryCenter .

The storage that is reserved for the shared library region is allocated in each broker address space when the first address space is started. The amount of storage that is allocated is controlled by the SHRLIBRGNSIZE parameter in z/OS, which is in the BPXPRMxx member of SYS1.PARMLIB. You can tune the amount of storage that is allocated for the shared library region. Investigate how much space you need, bearing in mind that other applications besides IBM Integration Bus might be using the shared library region. Then, adjust the SHRLIBRGNSIZE parameter accordingly.

Use the following MVS command to check that you have enough SHRLIBRGNSIZE to contain all the shared-library programs that are to be used on the system:

```
DISPLAY OMVS,LIMITS
```

If there is unused space in the shared library region, modify SHRLIBRGNSIZE to reduce the amount of storage that is allocated for the shared library region.

Storage is carved out of the high end of private storage of each address space that loads a system shared library object. The amount of storage that is allocated from each of these address spaces is based on the value of the SHRLIBRGNSIZE parameter in the BPXPRMxx PARMLIB member. If this value is too high, the storage that set aside for mapping the shared library region might interfere with the private storage requirements of individual address spaces. Therefore, specify the minimum size that is required to contain all of the shared library programs that are to be used on the system.

Note: z/OS UNIX attempts to map the entire SHRLIBRGNSIZE into the private region, not just the portion that contains programs. If the private region is too small to map the entire SHRLIBRGNSIZE, this shared library region is not used. No message is issued to indicate what happened.

Use the **SETOMVS** operator command to change SHRLIBRGNSIZE dynamically, or the BPXPRMxx PARMLIB member for permanent changes. See the z/OS UNIX *System Services* section in the z/OS V1R9.0 LibraryCenter.

Related concepts:



Integration Registry

The IBM Integration Bus provides an Integration Registry, which can store the following items:

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

Related reference:



Customization tasks and roles on z/OS

System programmers do most of the customization of IBM Integration Bus.

Related information:



z/OS V1R9.0 LibraryCenter

Resource Recovery Service planning on z/OS

About this task

This is part of the larger task of customizing your z/OS environment.

IBM Integration Bus for z/OS uses Resource Recovery Service (RRS) to coordinate changes to WebSphere MQ and DB2 resources. Ensure it is configured and active on your system, because your broker cannot connect to DB2 unless RRS is active.

Refer to the following manuals for detailed information about RRS: *z/OS V1R5.0 MVS Setting Up a Sysplex* and *z/OS V1R5.0 MVS Programming: Resource Recovery SA22-7616*.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

Related reference:



Customization tasks and roles on z/OS

System programmers do most of the customization of IBM Integration Bus.

Configuring WebSphere MQ on z/OS

This is part of the larger task of customizing your z/OS environment.

Procedure

To configure WebSphere MQ on z/OS:

1. Create a queue manager. You must have a separate WebSphere MQ queue manager for each broker.
2. Your queue manager must have a dead-letter queue. If the queue manager does not have a valid dead-letter queue, you must define one.
 - a. Check that this queue exists by using the WebSphere MQ command:
`+cpf DIS QMGR DEADQ`
 - b. Check that the queue exists by using the command:
`+cpf DIS QL(name) STGCLASS`
 - c. Then use the following command to check that the STGCLASS value is valid:
`+cpf DIS STGCLASS(...)`
3. Creating and deleting components on z/OS uses the WebSphere MQ command server.

- a. Ensure the command server component is running. The component is normally started automatically. See the *z/OS System Administration Guide* section of the WebSphere MQ Version 7 product documentation online for more details.
 - b. The command server requires a reply-to queue based on the model queue named SYSTEM.COMMAND.REPLY.MODEL. By default, this model queue is defined as permanent dynamic. However, if you leave the queue defined in this way, each time you run a create or delete component command these reply-to queues remain defined to the queue manager. To avoid these queues persisting, you can set the SYSTEM.COMMAND.REPLY.MODEL queue as temporary dynamic.
4. If you have enabled WebSphere MQ Security, you must define a set of security profiles for the broker to work. For a list of the security profiles required and instructions about how to define them, see WebSphere MQ authorizations.

Results

The WebSphere MQ environment is now ready. All IBM Integration Bus for z/OS system queues are defined during customization.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

Related reference:



Authorizations required for the IBM Integration Bus administrator

Related information:



WebSphere MQ Version 7 product documentation

Defining the started tasks to z/OS Workload Manager (WLM)

This is part of the larger task of customizing your z/OS environment.

About this task

With z/OS workload management (WLM), you define performance goals and assign a business importance to each goal. You define the goals for work in business terms, and the system decides how much resource, such as CPU and storage, should be given to it to meet the goal. Workload Manager constantly monitors the system, and adapts processing to meet the goals.

IBM Integration Bus automatically defines the JOBACCT token for each integration server started, so that the broker systems administrator can use WLM classification to map the JOBACCT token to specific Service and Report classes.

Each integration server can be assigned to a different Service and Report class. Whenever an integration server address space is started, the STEPNAME is assigned to the JOBACCT token.

When an integration server address space is started on z/OS, the STEPNAME is derived from the integration server label; the STEPNAME is defined as the last eight characters. Any characters that are not valid for a STEPNAME are replaced with the @ character.

For example, an integration server address space with the label MyExecutionGroup, has a STEPNAME of IONGROUP and assigns this same value to the JOBACCT.

The STEPNAME/JOBACCT is not guaranteed to be unique across multiple integration server address spaces on the same LPAR. If you require a unique STEPNAME/JOBACCT across multiple integration servers, you must use a suitable integration server naming standard.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

Related reference:



Customization tasks and roles on z/OS

System programmers do most of the customization of IBM Integration Bus.

Automatic Restart Manager planning

This is part of the larger task of customizing your z/OS environment.

IBM Integration Bus for z/OS allows you to register a component to the Automatic Restart Manager (ARM).

When customizing a component, you register it with ARM by editing the following environment variables in the component profile:

Description	Name
Switch that determines whether ARM is used (YES or NO).	MQSI_USE_ARM
ARM element name	MQSI_ARM_ELEMENTNAME
ARM element type	MQSI_ARM_ELEMENTTYPE

By default components do not register to ARM; the initial setting is MQSI_USE_ARM=NO. You can override this default value by setting MQSI_USE_ARM=YES and providing an ARM element name and type.

MQSI_ARM_ELEMENTNAME must be a maximum of eight characters in length, because IBM Integration Bus adds a prefix of SYSWMI. For example, if you supply the value MQP1BRK to ARM_ELEMENTNAME, the element you define in your ARM policy is SYSWMI_MQP1BRK.

To enable automatic restart you must also:

- Set up an ARM couple data set.

- Define the automatic restart actions that you want z/OS to perform in an ARM policy.
- Start the ARM policy.

The following manuals provide detailed information about ARM couple data sets, including samples:

- *z/OS MVS Programming: Sysplex Services Guide*
- *z/OS MVS Programming: Sysplex Services Reference*
- *z/OS MVS Setting up a Sysplex*

You can access these manuals from the z/OS V1R8.0 LibraryCenter

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

Mounting file systems

IBM Integration Bus requires several directories to be defined on the file system at run time.

About this task

`++HOME++` is the location of the environment file (ENVFILE) used to create the runtime environment in which IBM Integration Bus runs.

`++INSTALL++` refers to the IBM Integration Bus installation directories.

`++COMPONENTDIRECTORY++` is the location where all deployed configuration is written to and read from by the IBM Integration Bus runtime libraries.

`++JAVA++` is the location of the Java installation.

`++MQPATH++` is the location of the WebSphere MQ installation.

Because IBM Integration Bus can run in a shared file system sysplex environment, it is important for performance reasons that these directories are mounted locally to the LPAR in which IBM Integration Bus is started. This is particularly important for startup performance; if the IBM Integration Bus installation directories are not mounted locally, startup times can increase significantly.

You can check file system ownership from USS using the command `df -v`. For example, the following output shows that the broker installation file system is owned by MVS1:

```

/usr/lpp/mqsi/V9R0M0:>df -v .
Mounted on      Filesystem                Avail/Total   Files      Status
/usr/lpp/mqsi/V9R0M0 (OMVS.PLEXS.MQSI.V800.WBIMB) 7984/806400  4294966503
  Available
HFS, Read/Write, Device:89, ACLS=Y
File System Owner : MVS1      Automove=Y      Client=N
Filetag : T=off  codeset=0

```

To create a directory in an already mounted file system use the `mkdir` command. For example:

```
mkdir -p /mqsi/brokers/MQP1BRK
```

To mount a new file system, follow the instructions given in the *z/OS UNIX System Services Planning* manual.

From USS, use the following instructions:

```
mkdir -p /mqsi/brokers/MQP1BRK
mount -f MQSI.BROKER.MQP1BRK /mqsi/brokers/MQP1BRK
```

From TSO, use the following instructions:

```

ALLOCATE DATASET('MQSI.BROKER.MQP1BRK') DSNTYPE(HFS) SPACE(5,5) DIR(1) CYL
FREE DATASET('MQSI.BROKER.MQP1BRK')
MOUNT FILESYSTEM('MQSI.BROKER.MQP1BRK') TYPE(HFS)
      MOUNTPOINT('/mqsi/brokers/MQP1BRK')

```

Note that the preceding `ALLOCATE` command is an example; the dataset should be allocated the correct amount of storage as described in “Disk space requirements on z/OS” on page 623

Related concepts:

“Component directory on z/OS” on page 316

The *component directory* is the root directory of the component's runtime environment.

“Installation directory on z/OS” on page 315

On z/OS, the SMP/E installation places all the product executable files into a directory of a file system under UNIX System Services (USS).

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

“Using the file system on z/OS” on page 318

If you have more than one MVS image, consider how you will use the file system. You can share files in a file system with different members of a sysplex. The file system is mounted on one MVS image and requests to the file are routed to the owning system using XCF from systems which do not have it mounted.

Related reference:

“Disk space requirements on z/OS” on page 623

The installation of IBM Integration Bus for z/OS uses approximately 400 MB of disk space; plan on using 500 MB to allow for the component directories, and for new service fixes to be applied.

Checking the permission of the installation directory

This is part of the larger task of customizing your z/OS environment.

About this task

You must ensure that the appropriate user IDs, for example, the IBM Integration Bus Administrator and all component Started Task user IDs, have READ and EXECUTE permission to the IBM Integration Bus installation directory.

Using group access control to set these permissions.

Procedure

1. Display the permissions on the installation directory using the `ls` command.

```
ls -l /usr/lpp/mqsi
```

This command displays lines like the following response:

```
drwxr-xr-x  2 TS0USER MQM      8192 Jun 17 09:54 bin
```

In this example, MQM is the group associated with the directory. Those user IDs requiring permission to the directory must be a member of this group.

This example also shows the permissions defined for the directory. User TS0USER has `rwX` (READ, WRITE, and EXECUTE), group MQM has `rx`, and any other user ID has `rx`.

2. Ensure that the user IDs requiring permission have a group that matches that of the installation directory. Use the following command, where `userid` is the ID you want to check:

```
id <userid>
```

3. If the installation directory does not have a valid group, use the command `chgrp` to set the group of the directory:

```
chgrp -R <group> <pathname>
```

For example:

```
chgrp -R MQSI /usr/lpp/mqsi
```

You must be the owner of the group or have superuser authority to use this command.

4. If the installation directory does not have the correct permissions for the group (READ / EXECUTE), use the command `chmod` to change the permissions:

```
chmod -R g=rx <pathname>
```

For example:

```
chmod -R g=rx /usr/lpp/mqsi
```

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

Related reference:



mqsicreatebroker command

Use the **mqsicreatebroker** command to create a broker and its associated resources.

Customizing the version of Java on z/OS

Check the version of Java in your enterprise, and change it, if necessary.

About this task

This task is part of the larger task of customizing your z/OS environment.

For information about the levels of Java that IBM Integration Bus supports, see IBM Integration Bus Requirements. The 64 bit version of Java is required.

To check the current version of Java in a broker component, complete the following steps.

Procedure

1. Change to the bin directory in which you installed your Java product.

For example, if Java is installed in `/usr/lpp/java`, change to the `/usr/lpp/java/bin` directory, and type `./java -fullversion`. You receive the following response, or a response with similar content:

```
java version "1.6.0"  
Java(TM) 2 Runtime Environment, Standard Edition (build 1.6.0)  
Classic VM (build 1.6.0 J2RE 1.6.0 IBM z/OS Persistent Reusable VM build cm142-20040917  
(JIT enabled: jitc))
```

The response in this example confirms that Java 1.6.0 is the current version, which is the minimum level required for this platform.

2. If you want to set, or change, the current Java version, edit `JAVAHOME` in the broker profile (`BIPBPROF`).

For further details about the broker profile, see “Creating a broker on z/OS” on page 313.

3. Run `BIPGEN` to re-create the broker `ENVFILE` and any integration server specific `ENVFILES`

4. Restart the broker to pick up changes to the `ENVFILE`.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

Checking APF attributes of bipimain on z/OS

This task is part of the larger task of setting up your z/OS environment.

About this task

Use the `extattr` command to display the attributes of the object `bipimain`. For example:

```
extattr /usr/lpp/mqsi/bin/bipimain
```

The attribute APF authorized must be set to YES. If it is not, use `extattr +a bipimain` to set this attribute. For example:

```
extattr +a /usr/lpp/mqsi/bin/bipimain
```

You must have the appropriate authorization to issue this command.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

Configuring an integration server address space as non-swappable on z/OS

About this task

Because broker integration servers run as processes in UNIX System Services they cannot be set as NOSWAP in the PPT.

Instead, you can set the following environment variables in the broker environment file so that some, or all, of the address spaces of the broker integration servers request that they become non-swappable by the system; see “Creating the environment file” on page 325 for further information on adding an environment variable to a broker.

```
MQSI_NOSWAP=yes
```

sets the address spaces of all the integration servers to be non-swappable.

```
MQSI_NOSWAP_egname=yes
```

issues a request to the system, for each integration server labelled egname, that the address space be set as non-swappable.

```
MQSI_NOSWAP_uid=yes
```

issues a request to the system, for each integration server with the UUID labelled uuid, that the address space be set as non-swappable.

In order for the above requests to succeed, the broker's started task ID needs READ access to the BPX.STOR.SWAP facility class through their external security manager, for example, RACF.

When an application makes an address space non-swappable, it can cause additional real storage in the system to be converted to preferred storage. Because preferred storage cannot be configured offline, using this service can reduce the installation's ability to re-configure storage in the future.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

“Creating the environment file” on page 325

This is part of the larger task of creating a broker on z/OS.

Configuring brokers

Create and configure the brokers that you want on the operating system of your choice.

Before you begin

Before you start:

Ensure that the following requirements are met:

- Your user ID has the correct authorizations to perform the task. The authorizations are defined in Security requirements for administrative tasks.
- **Windows** On Windows: you have created a user ID to be used as the service user ID. You specify this ID when you create the broker; it is used to run the broker.

For more information about user ID authorization and creation, refer to “Planning for security” on page 30.

- You have initialized the command environment on distributed systems; see “Setting up a command environment” on page 119.

About this task

To set up a default configuration, or an environment for application development on Linux on x86 or Windows, see Chapter 4, “Configuring brokers for development environments,” on page 273. If you are creating brokers for test and production environments, use the tasks included in this section.

When you have created your physical components, you can administer your test and production environments either by using the IBM Integration Explorer, or programmatically by using the IBM Integration API (the CMP).

This collection of tasks uses specific resource names and user IDs. These names are examples only; you can use your own names. Follow existing naming conventions for WebSphere MQ and other resources.

Procedure

- “Creating a broker” on page 304
- “Verifying brokers” on page 330

What to do next

Next: Continue your broker configuration tasks in the IBM Integration Explorer. For day to day administration of your brokers and associated resources, see Administering brokers and broker resources.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Configuring integration nodes using the IBM Integration Explorer” on page 336
Configure your local and remote integration nodes (brokers) by using the IBM Integration Explorer.



Developing applications that use the IBM Integration API

Develop Java applications that use the IBM Integration API (also known as the CMP) to communicate with, deploy to, and manage brokers and their associated resources.

“Planning for security” on page 30

It is important to be aware of security issues when you are planning to install and use IBM Integration Bus.

Related reference:



Security requirements for administrative tasks

You can configure access permissions to govern which users and groups can manipulate objects in the broker network. Security requirements for administrative tasks depend on the platform that you use.

Creating a broker

You can create brokers on every platform that is supported by IBM Integration Bus. The broker runs as a 64-bit application on all platforms except Linux on x86 and Windows on x86.

Before you begin

Before you start:

Complete the following tasks:

- Ensure that your user ID has the correct authorizations to perform the task. Refer to Security requirements for administrative tasks.
- On distributed systems, you must set up your command-line environment before creating a broker, by running the product profile or console; refer to “Setting up a command environment” on page 119.
- On z/OS, you must create and start the queue manager for this broker before you create the component.

About this task

Create a broker by using the command line on the computer on which you have installed the Integration Bus component. On Windows and Linux on x86, you can alternatively use the IBM Integration Toolkit or IBM Integration Explorer to complete this task.

- You must give the broker a name that is unique on the local computer.
- Broker names are case sensitive on all supported platforms, except Windows.
- You must associate each broker with its own dedicated WebSphere MQ queue manager.

- Brokers can access only local queue managers, so you cannot create a broker on a queue manager that is on a remote system.

The mode in which your broker is working can affect the number of integration servers and message flows that you can deploy, and the types of node that you can use. See “Restrictions that apply in each operation mode” on page 10.

To create a broker, follow the link for the appropriate platform. Alternatively, follow the link to create a broker using the IBM Integration Toolkit or IBM Integration Explorer.

- Linux and UNIX
- Windows
- z/OS
- “Creating an integration node using the IBM Integration Explorer” on page 311

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Operation modes” on page 5

The operation mode that you can use for your broker is determined by the license that you purchase.

Related tasks:

“Configuring a broker as a WebSphere MQ service” on page 614

Use these topics to make changes when your broker is operating as a WebSphere MQ service.

“Using WebSphere MQ trusted applications” on page 306

Configure a broker to run as a WebSphere MQ trusted application.

“Modifying a broker” on page 331

Modify a broker by using the command line on the system where the Integration Bus component is installed.



Viewing broker properties

You can view broker properties by using the `mqsireportbroker` command. You can also use the IBM Integration Explorer to view broker properties.



Deleting a broker

Delete a broker using the command line on the system where the Integration Bus component is installed.



Considering security for a broker

Consider several factors when you are deciding which users can execute broker commands, and which users can control security for other broker resources.

Related reference:



`mqsicreatebroker` command

Use the `mqsicreatebroker` command to create a broker and its associated resources.



Security requirements for administrative tasks

You can configure access permissions to govern which users and groups can manipulate objects in the broker network. Security requirements for administrative tasks depend on the platform that you use.

“Support for 32-bit and 64-bit platforms” on page 625
IBM Integration Bus operates in 32-bit mode or 64-bit mode, on supported operating systems.

Using WebSphere MQ trusted applications

Configure a broker to run as a WebSphere MQ trusted application.

Before you begin

Before you start:

You must complete the following tasks:

- Ensure that your user ID is a member of the mqm group. On UNIX and Linux, specify the user ID mqm as the service user ID when you create the broker. On Windows, use any service user ID that is a member of mqm. Refer to Security requirements for administrative tasks.
- Review the restrictions that WebSphere MQ places on trusted applications that apply to your environment. See the section "Connection to a queue manager using the MQCONN call" in the *Application Programming Guide* section of the WebSphere MQ Version 7 product documentation online.

About this task

You can configure a broker to run as a trusted (fastpath) application on all supported platforms except z/OS, where the option is not applicable. If the broker is configured as a trusted application, it runs in the same process as the WebSphere MQ queue manager agent, and all broker processes benefit from an improvement in the overall system performance.

A broker does not run as a trusted application by default; you either create a trusted application by using the **mqsicreatebroker** command, or modify an existing broker by using the **mqsichangebroker** command.

Configuring a broker as a trusted application does not affect the operation of WebSphere MQ channel agents or listeners. For more information about running these as trusted applications, see the section "Running channels and listeners as trusted applications" in the *Intercommunication* section of the WebSphere MQ Version 7 product documentation online.

Take care when deploying user-defined nodes or parsers. Because a trusted application (the broker) runs in the same operating system process as the queue manager, a user-defined node or parser might compromise the integrity of the queue manager. Consider fully the restrictions that apply to your environment and test user-defined nodes and parsers in a non-trusted environment before deploying them in a trusted broker.

You can either configure a broker to run as a trusted application when you create it, or modify an existing broker.

Procedure

- To create a broker on a command line, run the **mqsicreatebroker** command with the **-t** flag, which specifies that the broker is created as a trusted application.
For example, enter the following command to create a broker called IB9NODE as a trusted application:

```
mqsicreatebroker IB9NODE -q IB9QMGR -i mqm -t
```


See “Creating a broker” on page 304 for more detailed information about how to create a broker for your platform.

- To modify an existing broker:
 1. Run the **mqsistop** command on the command line to stop the broker.
 2. Run the **mqsichangebroker** command with the **-t** flag. For example, enter the following command to modify a broker called IB9NODE to run as a trusted application:

```
mqsichangebroker IB9NODE -t
```

See “Modifying a broker” on page 331 for more detailed information about how to modify a broker for your platform.
 3. Restart the broker by using the **mqsistart** command. The broker restarts with fastpath set.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:



Starting and stopping a broker

Run the appropriate command to start or stop a broker.

“Creating a broker” on page 304

You can create brokers on every platform that is supported by IBM Integration Bus. The broker runs as a 64-bit application on all platforms except Linux on x86 and Windows on x86.

“Modifying a broker” on page 331

Modify a broker by using the command line on the system where the Integration Bus component is installed.

Related reference:



Security requirements for administrative tasks

You can configure access permissions to govern which users and groups can manipulate objects in the broker network. Security requirements for administrative tasks depend on the platform that you use.



mqsicreatebroker command

Use the **mqsicreatebroker** command to create a broker and its associated resources.



mqsichangebroker command

Use the **mqsichangebroker** command to change one or more of the configuration parameters of the broker.

Related information:



WebSphere MQ Version 7 product documentation

Creating a broker on Linux and UNIX systems

On Linux and UNIX systems, create brokers on the command line; on Linux on x86, you can also create brokers in the IBM Integration Toolkit.

Before you begin

Before you start:

- If you want to configure the broker as a WebSphere MQ trusted application, see “Using WebSphere MQ trusted applications” on page 306.
- Read Considering security for a broker.
- Check which operation mode you are licensed to use. If you do not set a mode, the automatic default is advanced mode; see “Operation modes” on page 5.

About this task

When you create a broker, if the WebSphere MQ queue manager does not exist, the queue manager is automatically created. If WebSphere MQ Version 7.1, or later, has been selected for the queue manager, the channel auth security is automatically disabled.

If the specified queue manager already exists when creating the broker, then it is assumed that the user has applied the appropriate security configuration to meet their requirements, and therefore channel auth security is not disabled.

To create a broker:

Procedure

1. Ensure that you are logged in using a user ID that has authority to run the **mqsicreatebroker** command.
2. Run the **mqsiprofile** script to set up the command environment for the broker:
`. install_dir/bin/mqsiprofile`

You must run this script before you can run the IBM Integration Bus commands.

For more information, see “Setting up a command environment” on page 119.

3. The command **mqsicreatebroker** defaults to using the primary installation of WebSphere MQ, if present, for the creation of queue managers. If you want to use a non-primary installation of WebSphere MQ you must configure your environment by using the **setmqenv** script before running the **mqsicreatebroker** command. For more details see: WebSphere MQ Version 7.1 **setmqenv**.

If you are creating a broker on a queue manager that already exists, there is no need to configure your WebSphere MQ environment.

4. Use the **mqsicreatebroker** command to create the broker.

For example, if you want to create a broker called IB9NODE on a queue manager called IB9QMGR, enter the following command:

```
mqsicreatebroker IB9NODE -q IB9QMGR
```

When you create a broker, the web user interface starts automatically on port 4414. You can disable the web user interface or change the port on which it runs by using the **mqsichangeproperties** command. If administration security is not enabled, web users can access the web user interface as a default user with unrestricted access to data and broker resources.

5. To enable function that becomes available in IBM Integration Bus fix packs, use the **-f** parameter on the **mqsichangebroker** command. For more information, see **mqsichangebroker** command.

Results

You have created a broker.

What to do next

Next: Complete the following tasks:

1. Start the broker by using the **mqsistart** command.
2. Connect to the broker from the IBM Integration Toolkit, the IBM Integration Explorer, or an CMP application.

When you have completed these tasks, you can create the resources that you want to associate with the broker; for example message flows. You can create and work with resources by using either the IBM Integration Toolkit or the CMP.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

“Using WebSphere MQ trusted applications” on page 306

Configure a broker to run as a WebSphere MQ trusted application.



Developing applications that use the IBM Integration API

Develop Java applications that use the IBM Integration API (also known as the CMP) to communicate with, deploy to, and manage brokers and their associated resources.

Related reference:



mqsicreatebroker command

Use the **mqsicreatebroker** command to create a broker and its associated resources.



mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.

Creating a broker on Windows

On Windows, you can create brokers on the command line.

Before you begin

Before you start:

- If you want to configure the broker as a WebSphere MQ trusted application, see “Using WebSphere MQ trusted applications” on page 306.
- Read Considering security for a broker.
- Check which operation mode you are licensed to use. If you do not set a mode, the automatic default is advanced mode; see “Operation modes” on page 5.

About this task

When you create a broker, if the WebSphere MQ queue manager does not exist, the queue manager is automatically created. If WebSphere MQ Version 7.1, or later, has been selected for the queue manager, the channel auth security is automatically disabled.

If the specified queue manager already exists when creating the broker, then it is assumed that the user has applied the appropriate security configuration to meet their requirements, and therefore channel auth security is not disabled.

To create a broker by using the command line, complete the following steps:

Procedure

1. Open an IBM Integration Bus command prompt for the runtime installation in which you want to create the broker. For more information about initializing the runtime environment, see “Command environment: Windows systems” on page 120.

On Windows systems, you must open a command console with elevated privileges. To open a command console with elevated privileges, use the **mqsicommandconsole** command. For more information, see **mqsicommandconsole** command.

2. The command **mqsicreatebroker** defaults to using the primary installation of WebSphere MQ, if present, for the creation of queue managers. If you want to use a non-primary installation of WebSphere MQ you must configure your environment by using the **setmqenv** script before running the **mqsicreatebroker** command. For more details see: WebSphere MQ Version 7.1 setmqenv.

If you are creating a broker on a queue manager that already exists, there is no need to configure your WebSphere MQ environment.

3. Use the **mqsicreatebroker** command to create the broker.

For example, if you want to create a broker called IB9NODE on a queue manager called IB9QMGR, enter the following command:

```
mqsicreatebroker IB9NODE -i wbrkuid -a wbrkpw -q IB9QMGR
```

where *wbrkuid* and *wbrkpw* are the user name and password under which the broker runs.

When you create a broker, the web user interface starts automatically on port 4414. You can disable the web user interface or change the port on which it runs by using the **mqsichangeproperties** command. If administration security is not enabled, web users can access the web user interface as a default user with unrestricted access to data and broker resources.

For more information about the command options, see **mqsicreatebroker** command.

Results

You have created a broker.

What to do next

Next: Complete the following tasks:

1. Start the broker by using the **mqsistart** command.
2. Connect to the broker from the IBM Integration Toolkit or an CMP application.

When you have completed these tasks, you can create the resources that you want to associate with the broker; for example message flows. You can create and work with resources by using either the IBM Integration Toolkit or the CMP.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

“Using WebSphere MQ trusted applications” on page 306

Configure a broker to run as a WebSphere MQ trusted application.



Developing applications that use the IBM Integration API

Develop Java applications that use the IBM Integration API (also known as the CMP) to communicate with, deploy to, and manage brokers and their associated resources.

Related reference:



mqsicommandconsole command

Use the **mqsicommandconsole** command to launch an elevated command console from which commands that require elevation on Windows can be run.



mqsicreatebroker command

Use the **mqsicreatebroker** command to create a broker and its associated resources.



mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.

Creating an integration node using the IBM Integration Explorer

On Linux on x86 or Windows, you can create integration nodes (brokers) by using the IBM Integration Explorer.

Before you begin

Before you start:

If you are using the IBM Integration Toolkit, see “Creating a broker for a development environment” on page 279.

- On Windows, you must have administrator access rights to create integration nodes by using the IBM Integration Toolkit or IBM Integration Explorer. When creating an integration node, you might be prompted to agree to the use of administrator rights or be prompted to enter an administrator user ID and password.
- If you want to configure the integration node as a WebSphere MQ trusted application, see “Using WebSphere MQ trusted applications” on page 306.
- Read Considering security for a broker.
- Check which operation mode you are licensed to use. If you do not set a mode, the automatic default is advanced mode; see “Operation modes” on page 5.

About this task

When you create an integration node, if the WebSphere MQ queue manager does not exist, the queue manager is automatically created. If WebSphere MQ Version 7.1, or later, has been selected for the queue manager, the channel auth security is automatically disabled.

If the specified queue manager already exists when creating the integration node, then it is assumed that the user has applied the appropriate security configuration to meet their requirements, and therefore channel auth security is not disabled.

To create an integration node by using the IBM Integration Explorer:

Procedure

1. Right-click the Integration Nodes folder, and click **New > Local Integration Node**. The Create Integration Node wizard is displayed.
2. Enter a name for the integration node, and click **Next**.
3. Enter a name for the queue manager, or select an existing queue manager from the list of available local queue managers. You cannot create an integration node on a queue manager that is already associated with an integration node, or on a queue manager that is on a remote system.
4. Enter values for your user name and password. On Windows only, the default user ID is LocalSystem (for the Windows Local System Account). Local System is a special account and has no password, therefore the password entry field is unavailable.
5. You must create an integration server for the integration node. By default, an integration server called default is created. If you want to create an integration server with a different name, enter the new name in **Create integration server**. You can create additional integration servers using the IBM Integration Toolkit or IBM Integration Explorer after the integration node is created.
6. Optional: On Windows, select whether to start the integration node automatically when Windows starts.
7. Optional: Select **Enable administration security** to activate broker administration security to control which users can complete specific tasks against that integration node and its resources.
8. The wizard starts processing your request. If the action succeeds, the wizard displays messages in the summary panel. If an action fails, the wizard reports the error in a message dialog.
If you know what is causing the error, and can fix it, correct the error and click **Yes**. The wizard reissues the command. If you do not know what is causing the error, or you cannot fix it, click **No**. The wizard backs out any actions that have already completed and returns your system to its initial state.
9. Click **Finish** to close the wizard.

Results

You have created and started an integration node.

Related concepts:



IBM Integration Explorer

The IBM Integration Explorer is a graphical user interface based on the Eclipse platform for administering your integration nodes (brokers).

Related tasks:

“Creating a broker” on page 304

You can create brokers on every platform that is supported by IBM Integration Bus. The broker runs as a 64-bit application on all platforms except Linux on x86 and Windows on x86.

“Creating a default configuration” on page 274

Use the Default Configuration wizard to create and test a basic broker configuration.



Setting up administration security

Control the actions that users can request against a broker and its resources.

“Configuring integration nodes using the IBM Integration Explorer” on page 336

Configure your local and remote integration nodes (brokers) by using the IBM Integration Explorer.



Administering brokers and broker resources

Administering brokers and associated integration node resources includes the tasks that you perform frequently to activate and manage those resources. Choose the method you prefer to administer your brokers and associated resources.

Creating a broker on z/OS

Create the broker component and the other resources on which it depends.

Before you begin

Before you start:

- Read Integration Bus server security.
- Check which operation mode you are licensed to use. The default is advanced mode; see “Operation modes” on page 5 for more information about supported operation modes, and “License requirements” on page 641 for more information about available licenses.
- Read through all the sub topics in the “Customizing the z/OS environment” on page 290 section, and follow the guidance within those topics.
- You must have installed:
 - WebSphere MQ for z/OS, with the optional JMS feature applied; for example, mounted at /usr/lpp/mqm
 - IBM Integration Bus for z/OS, with a broker file system mounted; for example, /usr/lpp/mqsi

About this task

Important: If you want to connect an IBM Integration Toolkit directly to your z/OS system, you *must* install the optional WebSphere MQ Client Attach feature. If you do not have the WebSphere MQ Client Attach feature installed, you can connect the IBM Integration Toolkit through an intermediate queue manager.

Procedure

1. Determine the customization information for your environment. You need to gather information about your environment before you can create a broker on z/OS. The following two topics document the information you need to gather. Complete the information that your enterprise requires.

If necessary, discuss the requirements with your system and WebSphere MQ administrators.

- “Installation information - broker” on page 320

- “Component information - broker” on page 321
2. Set up security for the started task user IDs. Start with Configuring broker user IDs on z/OS.
 3. To create your broker, perform the following tasks in order:
 - a. “Collecting the information required to create a broker” on page 319
 - b. “Creating the broker PDSE” on page 321
 - c. “Creating the broker directory on z/OS” on page 322
 - d. “Customizing the broker component data set” on page 323
 - e. “Customizing the broker JCL” on page 324
 - f. “Creating the environment file” on page 325
 - g. “Creating an integration server-specific environment file” on page 326
 - h. “Creating the Integration Bus component” on page 327
 - i. “Copying the broker started task to the procedures library” on page 328
 - j. Specifying an alternative user ID to run an integration server on z/OS

What to do next

To enable function that becomes available in IBM Integration Bus fix packs, use the **-f** parameter on the **mqsichangebroker** command. For more information, see **mqsichangebroker** command.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

“Modifying a broker on z/OS” on page 334

Use the **mqsichangebroker** command on z/OS to modify your broker.



Deleting a broker on z/OS

Delete the physical broker component.



Starting and stopping a broker on z/OS

Run the appropriate command from SDSF to start or stop a broker.

Related information:

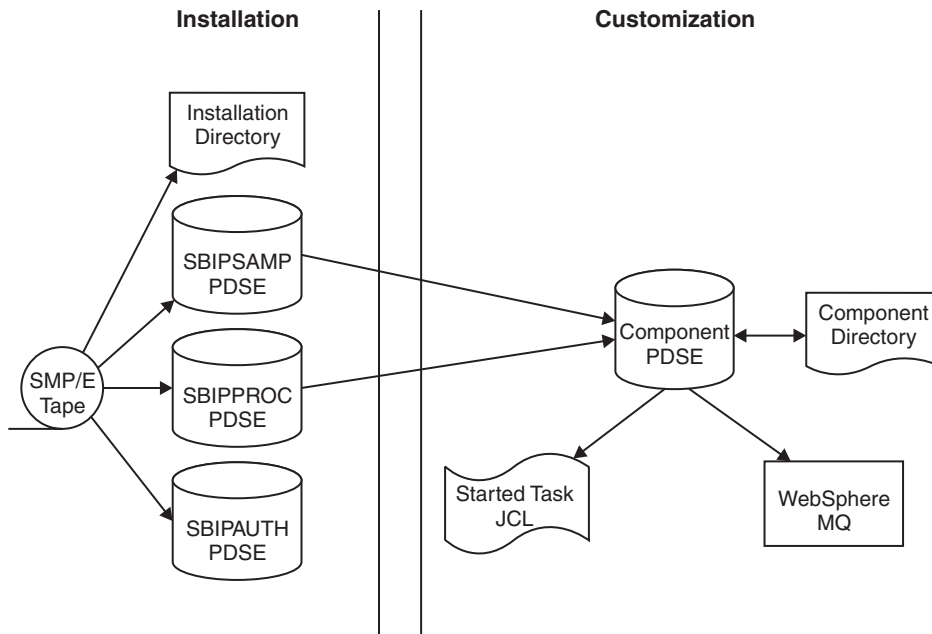


WebSphere MQ Version 7 product documentation

Overview of broker creation on z/OS:

After you have used SMP/E to install IBM Integration Bus for z/OS, the installed executable code is located inside the file system. JCL samples are located in the PDS <hlq>.SBIPPSAMP, the JCL procedures are located in the PDS <hlq>.SBIPPROC, and load module for synchronizing statistics with SMF are located in the PDS <hlq>.SBIPAUTH.

The following diagram gives an overview of the post-installation process.



To perform the customization, update and submit the required JCL. All necessary JCL is supplied to create the runtime environments of your broker. You start the broker using one of the supplied JCL files, which is run as a started task.

For more information, see:

- “Installation directory on z/OS”
- “Components on z/OS” on page 316
- “Component directory on z/OS” on page 316
- “Component PDSE on z/OS” on page 317
- “XPLink on z/OS” on page 317
- “Using the file system on z/OS” on page 318
- “Administration log messages on z/OS” on page 319
- “Security considerations on z/OS” on page 319

Related tasks:

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

Installation directory on z/OS:

On z/OS, the SMP/E installation places all the product executable files into a directory of a file system under UNIX System Services (USS).

For further information on mounting file systems and allocating space, including performance considerations, see “Mounting file systems” on page 298

Related concepts:

“Overview of broker creation on z/OS” on page 314

After you have used SMP/E to install IBM Integration Bus for z/OS, the installed executable code is located inside the file system. JCL samples are located in the PDS <hlq>.SBIPSAAMP, the JCL procedures are located in the PDS <hlq>.SBIPPROC, and load module for synchronizing statistics with SMF are located in the PDS

<h1q>.SBIPAUTH.

“Component directory on z/OS”

The *component directory* is the root directory of the component's runtime environment.

Related tasks:

“Mounting file systems” on page 298

IBM Integration Bus requires several directories to be defined on the file system at run time.

Components on z/OS: A *component* is a set of runtime processes that perform a specific set of functions, and comprises a broker.

A broker that is running has a control address space and one additional address space for each deployed integration server. When the control address space is started, the broker component is started automatically. This behavior can be changed by an optional start parameter in the started task.

The *component name* is the external name of the component and is used, for example, in the IBM Integration Toolkit.

Each component requires a name, which is usually the name of the started task that runs the component. This is typically the queue manager name with a suffix of the facility; for example, MQP1BRK for the broker.

Each component has its own runtime environment in UNIX System Services and needs its own WebSphere MQ queue manager.

Related concepts:

“Overview of broker creation on z/OS” on page 314

After you have used SMP/E to install IBM Integration Bus for z/OS, the installed executable code is located inside the file system. JCL samples are located in the PDS <h1q>.SBIPSAMP, the JCL procedures are located in the PDS <h1q>.SBIPPROC, and load module for synchronizing statistics with SMF are located in the PDS <h1q>.SBIPAUTH.

“Component directory on z/OS”

The *component directory* is the root directory of the component's runtime environment.

Component directory on z/OS:

The *component directory* is the root directory of the component's runtime environment.

The *component directory* is also referred to as ComponentDirectory in some instances within the code. Both the IBM Integration Bus administrator and the component require read and write access to the component directory.

An example directory for the broker component is /mqsi/brokers/MQP1BRK

For further information on mounting file systems and allocating space, including performance considerations, see “Mounting file systems” on page 298

Related concepts:

“Overview of broker creation on z/OS” on page 314

After you have used SMP/E to install IBM Integration Bus for z/OS, the installed executable code is located inside the file system. JCL samples are located in the PDS <h1q>.SBIPSAMP, the JCL procedures are located in the PDS <h1q>.SBIPPROC, and load module for synchronizing statistics with SMF are located in the PDS <h1q>.SBIPAUTH.

“Components on z/OS” on page 316

“Installation directory on z/OS” on page 315

On z/OS, the SMP/E installation places all the product executable files into a directory of a file system under UNIX System Services (USS).

Related tasks:

“Mounting file systems” on page 298

IBM Integration Bus requires several directories to be defined on the file system at run time.

Component PDSE on z/OS:

On z/OS, the *component PDSE* contains jobs customized for a single component. These jobs are used to create and administer the component.

The members specific to a component type are copied from <h1q>.SBIPSAMP and <h1q>.SBIPPROC to the component PDSE. These are then customized for the component.

The broker started-task user ID requires read access to its component PDSE at run time.

You must not share a PDSE across more than one *SYSplex* or *GRSplex*.

Related concepts:

“Overview of broker creation on z/OS” on page 314

After you have used SMP/E to install IBM Integration Bus for z/OS, the installed executable code is located inside the file system. JCL samples are located in the PDS <h1q>.SBIPSAMP, the JCL procedures are located in the PDS <h1q>.SBIPPROC, and load module for synchronizing statistics with SMF are located in the PDS <h1q>.SBIPAUTH.

“Components on z/OS” on page 316

“Component directory on z/OS” on page 316

The *component directory* is the root directory of the component's runtime environment.

“Installation directory on z/OS” on page 315

On z/OS, the SMP/E installation places all the product executable files into a directory of a file system under UNIX System Services (USS).

XPLink on z/OS: XPLink is a z/OS technology used by the C and C++ compilers to reduce the cost of function calling for programs written in these languages.

Many products, including IBM Integration Bus for z/OS, use XPLink technology to improve their performance. To ensure the highest possible performance gains, IBM Integration Bus requires as many as possible of the software components it uses to be XPLink-compliant. These include the broker, Java runtime, ODBC, and z/OS Language Environment®.

The IBM Integration Bus broker has been compiled by IBM to use XPLink technology and has been link-edited within the SMP/E environment to call the

appropriate XPLink routines of the software components it uses. Normally, these XPLink-enabled components are configured during their customization, and the broker needs only to locate the appropriate libraries to become XPLink-enabled.

Related concepts:

“Overview of broker creation on z/OS” on page 314

After you have used SMP/E to install IBM Integration Bus for z/OS, the installed executable code is located inside the file system. JCL samples are located in the PDS <hlq>.SBIPPSAMP, the JCL procedures are located in the PDS <hlq>.SBIPPROC, and load module for synchronizing statistics with SMF are located in the PDS <hlq>.SBIPAUTH.

Related tasks:

“Customizing the version of Java on z/OS” on page 301

Check the version of Java in your enterprise, and change it, if necessary.

Using the file system on z/OS:

If you have more than one MVS image, consider how you will use the file system. You can share files in a file system with different members of a sysplex. The file system is mounted on one MVS image and requests to the file are routed to the owning system using XCF from systems which do not have it mounted.

About this task

This is part of the larger task of customizing your z/OS environment.

Moving a broker from one image to another is straightforward and the files can be shared.

However, there is a performance overhead when using files shared between images in a file system because the data flows through the Coupling Facility (this is true for trace and other diagnostic data).

For further information on mounting file systems and allocating space, including performance considerations, see “Mounting file systems” on page 298

Space requirements:

About this task

For details of the disk space required, see “Disk space requirements on z/OS” on page 623.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

“Mounting file systems” on page 298

IBM Integration Bus requires several directories to be defined on the file system at run time.

Related reference:

“Disk space requirements on z/OS” on page 623

The installation of IBM Integration Bus for z/OS uses approximately 400 MB of disk space; plan on using 500 MB to allow for the component directories, and for new service fixes to be applied.



Customization tasks and roles on z/OS

System programmers do most of the customization of IBM Integration Bus.

Administration log messages on z/OS:

About this task

This is part of the larger task of customizing your z/OS environment.

On z/OS, all address spaces have a job log where BIP messages issued by IBM Integration Bus appear. Additionally, all messages appear on the syslog and important operator messages are filtered to the console using MPF (Message Processing Facility).

To prevent the operator's console receiving unnecessary BIP messages, you must configure MPF to suppress all BIP messages, with the exception of important messages. Note that you do not need to have the USS SYSLOG configured.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

Related reference:



Customization tasks and roles on z/OS

System programmers do most of the customization of IBM Integration Bus.

Security considerations on z/OS:

About this task

This is part of the larger task of customizing your z/OS environment.

The role of the IBM Integration Bus administrator includes customizing and configuring, running utilities, performing problem determination, and collecting diagnostic materials. People involved in these activities need IBM Integration Bus authorities. You must set up some security for IBM Integration Bus to work properly. The information that you need to do this is in Configuring broker user IDs on z/OS.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.



Configuring broker user IDs on z/OS

When you create your brokers on z/OS, you must set up security by configuring broker user IDs with the appropriate permissions.

Related reference:



Customization tasks and roles on z/OS

System programmers do most of the customization of IBM Integration Bus.

Collecting the information required to create a broker:

This is part of the larger task of creating a broker on z/OS.

About this task

You must complete the information in each of the tables, at the following links, before continuing:

- “Installation information - broker”
- “Component information - broker” on page 321

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

Related reference:



Customization planning checklist for z/OS

Use the information contained in the following tables to make a note of the values to use when you customize your system variables on z/OS.

Installation information - broker:

Decide on the values for the list of the JCL variables for your system; an example installation value is provided for each one.

Collect the information shown in the Description column, and complete the values that you require for your particular system. You can see a complete list of the variables that you can customize in z/OS JCL variables .

Description	JCL variable	Example installation value	Your installation value
Fully qualified name of the product's SBIPPROC data set	N/A	<hlq>.SBIPPROC	
Fully qualified name of the product's SBIPSAMP data set	N/A	<hlq>.SBIPSAMP	
File system directory where the product has been installed	++INSTALL++	/usr/lpp/mqsi	
The letter for the language in which you want messages shown.	++LANGLETTER++ ²	E (English)	E
Locale of environment where commands are run by submitting JCL	++LOCALE++	C	
Time zone of environment where commands are run by submitting JCL	++TIMEZONE++	GMT0BST	
Location of Java installation	++JAVA++	/usr/lpp/java/IBM/J1.6	
WebSphere MQ high-level qualifier	++WMQHLQ++	MQM.V700	

Notes:

1. See the WebSphere MQ documentation for a list of supported national languages.

Related tasks:

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

“Collecting the information required to create a broker” on page 319
 This is part of the larger task of creating a broker on z/OS.

Related information:

 [WebSphere MQ Version 7 product documentation](#)

Component information - broker:

Decide on the details of your broker component.

Collect the information shown in the Description column and complete the values you require for your particular system. You can see a complete list of variables you can customize in z/OS JCL variables .

Description	JCL variable	Example component value	Your component value
Home directory of the file system for the broker user ID	++HOME++	/u/mqp1usr/mqp1brk	
Queue Manager associated with the broker	++QUEUEMANAGER++	MQP1	
File system directory where the broker is to exist	++COMPONENTDIRECTORY++	/mqsi/brokers/MQP1BRK	
Broker name	++COMPONENTNAME++	MQP1BRK	
Data set where all JCL relevant to the broker is saved	++COMPONENTDATASET++	TESTDEV.BROKER.MQP1BRK	
Name of the Started Task JCL; can be a maximum of eight characters	++STARTEDTASKNAME++	MQP1BRK	
mqsicreatebroker options	++OPTIONS++	Any additional optional parameters for the mqsicreatebroker command	

Related tasks:

“Creating a broker on z/OS” on page 313
 Create the broker component and the other resources on which it depends.
 “Collecting the information required to create a broker” on page 319
 This is part of the larger task of creating a broker on z/OS.

Creating the broker PDSE:

This is part of the larger task of creating a broker on z/OS.

About this task

Each broker requires a PDSE or a PDS. A PDSE is preferable to a PDS because free space is available without the need to compress the data set.

Create the broker PDSE, for example using option 3.2 on ISPF. The name of the PDSE must be the same as the JCL variable ++COMPONENTDATASET++. Allocate a data set that has the following characteristics:

- Eight directory blocks
- 15 tracks (or 1 cylinder) of 3390 DASD with a record format of fixed blocked 80
- A suitable block size (for example 27920)
- A data set type of library

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

“Creating the broker directory on z/OS”

This is part of the larger task of creating a broker on z/OS.

Creating the broker directory on z/OS:

This is part of the larger task of creating a broker on z/OS.

About this task

Before you start

Before starting this task, you must have completed “Collecting the information required to create a broker” on page 319 and “Creating the broker PDSE” on page 321.

Procedure

1. Use the TSO command OMVS to get into OMVS.
2. Create the broker root directory using the command:

```
mkdir -p <ComponentDirectory>
```

The name of the directory must be the same as the JCL variable ++COMPONENTDIRECTORY++.

3. Display the contents of the directory, which is currently empty, using the command:

```
ls -dl /var/wmqi/MQP1BRK
```
4. Display the permissions on the directory using the command:

```
ls -al /var/wmqi/MQP1BRK
```
5. Ensure that the user ID of the person doing the customization has a group that matches the group of the directory. Use the following command, where `userid` is the ID you want to check:

```
id <userid>
```
6. Check that the directory has a valid group, and that the group has `rxw` permissions. If they do not, use the following command to set the group of the directory:

```
chgrp <group> <pathname>
```

For example:

```
chgrp WMQI /var/wmqi/MQP1BRK
```

You must be the owner of the group, or have superuser authority, to use this command.

7. To give the group `READ`, `WRITE`, and `EXECUTE` access, use the following command:

```
chmod g=rxw <pathname>
```

For example:

```
chmod g=rwx /usr/wmqj/MQP1BRK
```

8. To display the amount of space used and available, use the following command:

```
df -P /var/wmqj/MQP1BRK
```

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

“Creating the broker PDSE” on page 321

This is part of the larger task of creating a broker on z/OS.

“Checking the permission of the installation directory” on page 300

This is part of the larger task of customizing your z/OS environment.

Customizing the broker component data set:

This is part of the larger task of creating a broker on z/OS.

About this task

Before you start

Before starting this task, you must have completed “Collecting the information required to create a broker” on page 319.

Create the broker data set in TSO, identified by ++COMPONENTDATASET++, as instructed in the following steps:

Procedure

1. Copy all the listed files from <hlq>.SBIPSAMP to ++COMPONENTDATASET++.
2. Copy all the listed files from <hlq>.SBIPPROC to ++COMPONENTDATASET++.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

“Collecting the information required to create a broker” on page 319

This is part of the larger task of creating a broker on z/OS.

Related reference:



Contents of the broker PDSE

After you have successfully customized the broker, the broker PDSE members have been set up.

Customizing the broker JCL:

This subtask is part of the larger task of creating a broker on z/OS.

About this task

Before you start

Before starting this task, you must have completed “Customizing the broker component data set” on page 323.

All JCL has a standard header, which includes the following items:

- A brief description of its function.
- A description where further information can be found, relating to the function of the JCL.
- If appropriate, a topic number.
- The section listing the JCL variables themselves.

Each JCL file defines its own STEPLIB.

You can customize the files using an ISPF edit macro that you have to tailor, or you can change each of the PDSE members manually.

BIPEDIT is a REXX program that you can use to help you customize your JCL. After you have customized BIPEDIT, you can run this REXX program against the other JCL files to change their JCL variables.

When you update BIPBPROF (the broker profile), the changes are not accessible until you run BIPGEN to copy the profile to the file system and create the ENVFILE. You must run BIPGEN each time you update BIPBPROF for the changes to take effect. If you have integration server specific profiles, and you change either BIPBPROF or BIPEPROF (renamed to the integration server label), you must also run BIPGEN.

Procedure

1. Customize the renamed BIPEDIT file. Use the information you collected in:
 - “Installation information - broker” on page 320
 - “Component information - broker” on page 321
2. Activate the renamed BIPEDIT file before you customize other JCL files, by running the following TSO command:

```
ALTLIB ACTIVATE APPLICATION(EXEC) DA('COMPONENTDATASET')
```

where 'COMPONENTDATASET' is identical to ++COMPONENTDATASET++.

This command is active for the local ISPF session for which it was issued. If you have split screen sessions, the other sessions are not able to use this command. If you use ISPF option 6 to issue the command, use ISPF option 3.4 to edit the data set, which enables you to use the edit command.

3. Edit each JCL file. Run the renamed BIPEDIT file by typing its name on the command line (for example MQ01EDBK). Instead of editing a member, you might want to View it until you have resolved all problems in your REXX program. Alternatively, you can Cancel the Edit session instead of saving it.

You must set a value for all the variables listed in the JCL; if you do not do so, the JCL does not work correctly.

What to do next

Some JCL files include ++OPTIONS++ for a command; you must replace them with additional optional parameters specific to the command on z/OS, or remove them. Typically, you must replace or remove these options in addition to running BIPEDIT. If you do not require additional options, remove ++OPTIONS++ by using the following command:

```
"c ++OPTIONS++ ' ' all"
```

where ' ' represents two single quotation marks.

Save the edit macro and run this macro against all the members except the edit macro itself.

If the user ID submitting the BIPCRBK command has the appropriate WebSphere MQ authorities, you can ignore the optional **mqsicreatebroker** parameters -1 and -2.

If you expect different administrators to create the WebSphere MQ resources, you can consider using one of these optional parameters; see **mqsicreatebroker** command for further information.

You must be aware that another process might be using the current ENVFILE, therefore you must consider whether updating the current ENVFILE in the file system might have other effects.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

Related reference:



mqsicreatebroker command

Use the **mqsicreatebroker** command to create a broker and its associated resources.

Creating the environment file:

This is part of the larger task of creating a broker on z/OS.

About this task

Before you start

Before starting this task, you must have completed “Customizing the broker JCL” on page 324

Procedure

1. Review the BIPBPROF member. If you define parameters for all users, you can configure BIPBPROF to use these parameters.

Note that the BIPBPROF environment file is the main broker file, and it is used by default for any integration server address spaces, unless a specific integration server environment file exists.

For example, if the time zone option TZ is set as a system-wide parameter for all users, you can remove it from BIPBPROF.

2. Submit member BIPGEN. Review the job output and make sure that the environment file in the output contains the parameters that you expect.

If you change BIPBPROF, or system-wide parameters, you must submit BIPGEN again to pick up the changes.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

“Customizing the broker JCL” on page 324

This subtask is part of the larger task of creating a broker on z/OS.

Creating an integration server-specific environment file:

An integration server-specific environment file can be used instead of the default broker environment file when you restart the integration server.

Before you begin

Before you start this task, complete the steps in “Customizing the broker JCL” on page 324.

About this task

This task forms an optional step of the larger task of creating a broker on z/OS.

By default, a broker has a single environment file (ENVFILE), which is created from BIPBPROF by submitting BIPGEN. The main broker address space and any integration server address spaces use this environment file.

You might need to have an integration server-specific environment file. In this situation, when you restart the integration server, the integration server address space uses the specific environment file and not the default broker environment file.

Procedure

1. Review the BIPBPROF member. If you define parameters that are required by the broker and all integration server address spaces, you can configure BIPBPROF to use these parameters.
2. Copy BIPEPROF and rename the copy of the file to be the same as the integration server label in the component data set, by using the following rule.
The name must equal the last eight characters of the integration server label. Any characters that are not valid for a STEPNAME are replaced with the @ character.

Furthermore, if the first character of a STEPNAME is not an alphabetic character, that character is replaced with an A. (For more information about characters that you can use, see Administration in z/OS.)

For example, an integration server address space with the label MyExecutionGroup, has an integration server-specific profile called IONGROUP.

3. Customize the renamed BIPEPROF file to include any environment variable settings that are specific to the integration server.
4. Update BIPGEN by adding an extra step at the end for the new integration server-specific profile; for example:

```
//BIPEG01 EXEC PROC=BIPEGEN,EG=IONGROUP
```
5. Submit member BIPGEN to re-create the default broker environment file, and to create the new integration server-specific environment file. Review the job output and make sure that the environment file in the output contains the parameters that you expect.
If you change BIPBPROF, or the renamed BIPEPROF file, or system-wide parameters, you must submit BIPGEN again to pick up the changes.

What to do next

Multiple integration server-specific environment files

If you need multiple integration server-specific files, complete the following steps.

1. Create an integration server-specific profile for each file, following the naming rules that are specified in step 2 on page 326.
2. Add extra steps to the end of BIPGEN, giving each step a different name, and specifying the correct eight-character integration server label.
3. Resubmit BIPGEN to create all the environment files.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

“Customizing the broker JCL” on page 324

This subtask is part of the larger task of creating a broker on z/OS.

“Creating the environment file” on page 325

This is part of the larger task of creating a broker on z/OS.

Creating the Integration Bus component:

When you are creating an integration node (broker) on z/OS, one of the tasks is to create the Integration Bus component.

About this task

If the user ID submitting the BIPCRBK command has the appropriate WebSphere MQ authority, you can ignore the optional **mqsicreatebroker** parameters -1 and -2. If it is your intention to have different administrators create the WebSphere MQ resources, you can consider using one of these optional parameters; see **mqsicreatebroker** command for further information.

Procedure

1. Submit job BIPCRBK with option -1. This job creates the files and directories that are placed in the default storage group. You must run this job first; to run this job you need authority to access the broker root directory.
2. Edit BIPCRBK and submit the job with option -2. This job creates the WebSphere MQ queues. If you do not have the requisite authority, ask your WebSphere MQ system administrator to run the job.
3. Ensure that the jobs have run successfully by:
 - Checking the STDOUT stream in the JOBLOG.
 - Viewing STDOUT for any errors and checking for BIP8071I: Successful command completion.

What to do next

If you encounter any problems, delete the broker and re-create it using the following procedure. You must have the appropriate authority to run the jobs.

1. Edit and configure job BIPDLBK.
2. Run job BIPDLBK with the same option, or options, that caused the problems when you ran the BIPCRBK job.
3. Correct the problems and run the BIPCRBK job again.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

Related reference:



mqsicreatebroker command

Use the **mqsicreatebroker** command to create a broker and its associated resources.

Copying the broker started task to the procedures library:

This is part of the larger task of creating a broker on z/OS.

About this task

Before you start

Before starting this task, you must have completed “Creating the Integration Bus component” on page 327.

Procedure

1. Ensure that the user ID for the broker started task is defined and that the broker procedure is associated with the user ID. If you are using a security manager, for example RACF, update the started class for your broker. See Configuring broker user IDs on z/OS and Summary of required access (z/OS) for more information.
2. Copy the Started Task JCL (BIPBRKP) to the procedures library, for example USER.PROCLIB.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:



Configuring broker user IDs on z/OS

When you create your brokers on z/OS, you must set up security by configuring broker user IDs with the appropriate permissions.

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

“Creating the Integration Bus component” on page 327

When you are creating an integration node (broker) on z/OS, one of the tasks is to create the Integration Bus component.



Starting and stopping a broker on z/OS

Run the appropriate command from SDSF to start or stop a broker.

Related reference:



Summary of required access (z/OS)

The professionals in your organization require access to components and resources on z/OS.

Collecting broker statistics on z/OS:

About this task

If you need to write broker statistics to SMF, you need to have the library <HLQ>.SBIPAUTH in your STEPLIB.

This library and all the libraries in the STEPLIB concatenation need to be APF authorized.

You can use the **mqsichangeflowstats** command with o=SMF for this purpose.

You must not share a PDSE across more than one *SYSplex* or *GRSplex*.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

Related reference:



mqsichangeflowstats command

Use the **mqsichangeflowstats** command to control the accumulation of statistics about message flow operation.

IBM Integration Bus and WebSphere MQ setup verification:

Whenever a broker starts, a basic component verification runs automatically.

About this task

This verification checks the following items:

- Basic WebSphere MQ information
- Registry information
- Setup verification

If an error is found the component does not start. All output from the verification step is written to the component JOBLOG.

You must also validate that the entire IBM Integration Bus system that you have installed is running correctly; that is, there are no errors in the system and the system is performing as expected. This check is especially important when you are installing IBM Integration Bus for the first time.

Download and review *IBM Integration Bus SupportPac IP13*.

This SupportPac provides:

- Tools to help test IBM Integration Bus flows running on z/OS. These tools provide:
 - Put and get user messages
 - Measurement of the elapsed time
 - CPU time used to process the messages
 - Recommendations on the environment configuration.
- Guidelines on designing flows for evaluation, to reduce the need for applications outside of the IBM Integration Bus environment.
- IBM Integration Bus flows, which allow you to compare the throughput you achieve in your environment with the results achieved at IBM. Using the tools supplied in *IBM Integration Bus SupportPac IP13* can give suggestions as to why the results might be different.

Access *IBM Integration Bus SupportPac IP13* online.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

Verifying brokers

Use the **mqsilist** command to display the brokers that you have created on your computer.

About this task

If you run the **mqsilist** command with no parameters, a one line summary for every broker that you have created for the current installation is displayed. For example:

```
BIP1281I: Broker 'BrokerA' on queue manager 'QMA' is running.
BIP1281I: Broker 'IB9NODE' on queue manager 'MB7QM' is running.
BIP1281I: Broker 'MBTEST1' on queue manager 'TESTQM' is stopped.
BIP8071I: Successful command completion.
```

You can request further details about all resources by specifying parameter **-d 1** or **-d 2** on the command, You can also specify the **-r** parameter so that the command recursively returns information about integration servers, and the message flows and files that you have deployed to those integration servers.

You can also use this command to display a list of brokers that you have created for all concurrent installations on this computer. For example, you might have installed both Version 6.1 and Version 8.0 for assessment and migration.

mqsilist -a

In earlier versions, only brokers that you have created are displayed. If you want to view information about other components, you must run the corresponding command for that version

Related reference:



mqsilist command

Use the **mqsilist** command to list installed brokers and their associated resources.

Modifying a broker

Modify a broker by using the command line on the system where the Integration Bus component is installed.

Before you begin

Before you start:

You must have completed the following tasks:

- Ensure that your user ID has the correct authorizations to perform the task; see Security requirements for administrative tasks.
- Create a broker.
- On Windows, Linux, and UNIX systems, you must set up your command-line environment before performing this task by running the product profile or console; see “Setting up a command environment” on page 119.

About this task

The parameters you can change on the broker affect the physical broker that was created by using the command line.

You can also modify the broker in the IBM Integration Explorer, where you can change broker properties, such as security settings.

Choose the appropriate task for your platform from the following links:

- “Modifying a broker on Windows, Linux, and UNIX systems” on page 332
- “Modifying a broker on z/OS” on page 334

- “Configuring integration nodes using the IBM Integration Explorer” on page 336

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

Related tasks:

“Creating a broker” on page 304

You can create brokers on every platform that is supported by IBM Integration Bus. The broker runs as a 64-bit application on all platforms except Linux on x86 and Windows on x86.



Deleting a broker

Delete a broker using the command line on the system where the Integration Bus component is installed.



Starting and stopping a broker

Run the appropriate command to start or stop a broker.



Viewing broker properties

You can view broker properties by using the `mqsireportbroker` command. You can also use the IBM Integration Explorer to view broker properties.

Related reference:



Security requirements for administrative tasks

You can configure access permissions to govern which users and groups can manipulate objects in the broker network. Security requirements for administrative tasks depend on the platform that you use.



`mqsichangebroker` command

Use the `mqsichangebroker` command to change one or more of the configuration parameters of the broker.



`mqsireportbroker` command

Use the `mqsireportbroker` command to display broker registry entries.

Modifying a broker on Windows, Linux, and UNIX systems

Use the `mqsichangebroker` command on Windows, Linux, and UNIX to modify your broker.

Before you begin

Before you start:

You must have completed the following task for the appropriate platform:

- “Creating a broker on Linux and UNIX systems” on page 307
- “Creating a broker on Windows” on page 309

About this task

To modify a broker on Windows, Linux, and UNIX:

Procedure

1. Stop the broker by using the `mqsistop` command.

2. Enter the **mqsichangebroker** command, specifying the broker name and one or more parameters that you want to change.

```
mqsichangebroker brokername  
<<-i ServiceUserID> -a ServicePassword>  
<-t | -n> <-l UserLilPath>  
<-g ConfigurationTimeout> <-k ConfigurationDelayTimeout>  
<-v StatisticsMajorInterval> <-P httpListenerPort> <-y ldapPrincipal>  
<-z ldapCredentials> <-c ICUconverterpath> <-x userExitPath>  
<-e activeUserExits> <-o operationMode> <-f functionlevel>
```

where:

brokername

Is the broker name.

- i Is the service user ID that is used to run the broker (applicable on Windows only).
- a Is the password for the service user ID (applicable on Windows only).
- t Indicates that the broker runs as a WebSphere MQ trusted application.
- n Indicates that the broker must cease to run as a WebSphere MQ trusted application.
- l Indicates from where LIL (loadable implementation libraries) files are loaded.
- g Is the maximum time (in seconds) to allow a broker to process a deployed message.
- k Is the maximum time (in seconds) to allow a broker to process a minimum size deployed message.
- v Is the time (in minutes) for the duration of the interval for collecting statistics archive records.
- P Is the port that the broker HTTP listener will use.
The broker starts this listener when a message flow that includes HTTP nodes or Web Services support is started
- y Is the user principal for access to an LDAP directory.
- z Is the user password for access to LDAP.
- c Is a delimited set of directories to search for additional code page converters.
- x Is the path that contains the location of all user exits to be loaded for integration servers in this broker.
- e Is the list of active user exits.
- o Is the operation mode that the broker will use.
- f Indicates the maximum function level of your broker that you want to enable.
- s Indicates whether broker administration security is enabled.

For example:

- To change the user ID that is used to run the broker on Windows, enter the following command:
mqsichangebroker IB9NODE -i wbrkuid -a wbrkpw
- To change the configuration timeout, enter the following command:

```
mqsichangebroker IB9NODE -g 500
```

- To activate broker administration security, enter the following command:

```
mqsichangebroker IB9NODE -s active
```

For further information about these parameters, and more examples, see **mqsichangebroker** command - Linux and UNIX systems and **mqsichangebroker** command - Windows systems.

3. Restart the broker by using the **mqsistart** command. The broker restarts with the new properties.

Results

You cannot change all the parameters with which you created a broker. If you cannot change a property by using the **mqsichangebroker** command, delete the broker and create another broker with the new properties.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

“Operation modes” on page 5

The operation mode that you can use for your broker is determined by the license that you purchase.

Related tasks:



Starting and stopping a broker

Run the appropriate command to start or stop a broker.

“Creating a broker” on page 304

You can create brokers on every platform that is supported by IBM Integration Bus. The broker runs as a 64-bit application on all platforms except Linux on x86 and Windows on x86.



Deleting a broker

Delete a broker using the command line on the system where the Integration Bus component is installed.

Related reference:



mqsichangebroker command

Use the **mqsichangebroker** command to change one or more of the configuration parameters of the broker.



mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.



mqsistop command

Use the **mqsistop** command to stop the specified component.

Modifying a broker on z/OS

Use the **mqsichangebroker** command on z/OS to modify your broker.

Before you begin

Before you start:

You must have completed the following task:

- “Creating a broker on z/OS” on page 313

About this task

To modify a broker:

Procedure

1. Ensure that the broker is running.
2. Stop the broker components by issuing the following command:
`/F BROKERNAME, PC`
3. When the broker has stopped, use the **MVS MODIFY** command with the **changebroker** parameters that you want to change. For example:
`/F <BROKERNAME>,cb g=100,k=200`
4. Restart the broker components by issuing the following command:
`/F BROKERNAME, SC`
The broker now uses the changed parameters.

What to do next

You cannot change all the parameters with which you created a broker. If you cannot modify a parameter that you want to change by using the **changebroker** command, delete the broker and create a new one. By creating another broker you can redefine all the parameters.

You can change the following parameters:

g	ConfigurationChangeTimeout
k	InternalConfigurationTimeout
l	UserLilPath
v	StatisticsMajorInterval
P	HTTPListenerPort
y	LdapPrincipal
z	LdapCredentials
c	ICUConverterPath
x	UserExitPath
e	ActiveUserExits
f	functionlevel

For further information about these parameters, see **mqs i changebroker** command - z/OS.

Related concepts:



The IBM Integration Bus environment

An integration node (broker) is a set of execution processes that hosts one or more

message flows to route, transform, and enrich in flight messages.

Related tasks:

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.



Deleting a broker on z/OS

Delete the physical broker component.

Related reference:



START and STOP commands on z/OS



mqsichangebroker command

Use the **mqsichangebroker** command to change one or more of the configuration parameters of the broker.



mqsistart command

Use the **mqsistart** command to start the specified broker if all initial verification tests complete successfully.



mqsistop command

Use the **mqsistop** command to stop the specified component.

Configuring integration nodes using the IBM Integration Explorer

Configure your local and remote integration nodes (brokers) by using the IBM Integration Explorer.

Before you begin

If you are using systems other than Linux on x86 or Windows, you must first create your integration nodes (brokers) on those systems; see “Configuring brokers” on page 303.

If you are using Windows, you must have administrator access rights to create integration nodes by using the IBM Integration Explorer. When creating an integration node, you might be prompted to agree to the use of administrator rights or be prompted to enter an administrator user ID and password.

About this task

On Linux on x86 or Windows, use the IBM Integration Explorer to create local integration nodes, and to configure, modify, and administer your local and remote integration nodes.

To start the IBM Integration Explorer:

- **Linux** Open a command shell in which the **mqsiprofile** command has not been run and enter the **strmqcfg** command, or run **/usr/bin/strmqcfg**.
- **Windows** Click **Start > All Programs > IBM WebSphere MQ > WebSphere MQ Explorer**, or double-click the shortcut on your desktop labeled 'WebSphere MQ Explorer'. On Windows systems, if user account control is enabled, you must right-click **WebSphere MQ Explorer**, and select **Run as Administrator** to start the application with the appropriate privileges.

In the IBM Integration Explorer you can configure integration node, integration server and message flow properties including security and configurable services. See the following tasks for more information:

- “Configuring integration node properties in the IBM Integration Explorer”
- “Using the IBM Integration Explorer to work with configurable services” on page 345

Related concepts:

 IBM Integration Explorer

The IBM Integration Explorer is a graphical user interface based on the Eclipse platform for administering your integration nodes (brokers).

Related tasks:

“Planning an IBM Integration Bus environment” on page 14


When you start to plan your IBM Integration Bus environment, you must first consider the design of your system infrastructure. A major consideration when you are planning to use IBM Integration Bus is how it will fit into your overall system architecture, and it is important to understand the movement of data around your whole system.

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.

 Administering brokers and broker resources

Administering brokers and associated integration node resources includes the tasks that you perform frequently to activate and manage those resources. Choose the method you prefer to administer your brokers and associated resources.

 Viewing accounting and statistics data in the IBM Integration Explorer

You can use the Broker Statistics and Broker Statistics Graph views in the IBM Integration Explorer to view snapshot accounting and statistics data as it is produced by the integration node (broker).

“Changing IBM Integration Explorer preferences” on page 356

Change preferences in the IBM Integration Explorer.

Configuring integration node properties in the IBM Integration Explorer

Integration nodes (brokers) and associated resource, such as integration servers and message flows, have properties that can be modified. Use the IBM Integration Explorer to configure the properties for your integration nodes, integration servers and message flows. Alternatively you can use the command line or the IBM Integration API.

About this task

See the following tasks for properties on message flows:

- “Configuring description properties in the IBM Integration Explorer” on page 338
- “Configuring DataPower security settings” on page 339

See the following tasks for properties on integration servers:

- “Configuring description properties in the IBM Integration Explorer” on page 338
- “Configuring the flow debug port in the IBM Integration Explorer” on page 342

- “Configuring DataPower security settings” on page 339
- Configuring the embedded global cache by using IBM Integration Explorer

See the following tasks for properties on brokers:

- “Configuring description properties in the IBM Integration Explorer”
- “Changing the operation mode of the integration node using the IBM Integration Explorer” on page 343
- “Changing the integration node security settings in the IBM Integration Explorer” on page 344
- “Configuring DataPower security settings” on page 339
- Configuring the embedded global cache by using IBM Integration Explorer

Related tasks:

“Configuring integration nodes using the IBM Integration Explorer” on page 336
 Configure your local and remote integration nodes (brokers) by using the IBM Integration Explorer.

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.



Administering brokers and broker resources

Administering brokers and associated integration node resources includes the tasks that you perform frequently to activate and manage those resources. Choose the method you prefer to administer your brokers and associated resources.

Configuring description properties in the IBM Integration Explorer

Use the IBM Integration Explorer to configure the Long Description and Short Description on your integration nodes (brokers), integration servers and deployed message flows. Alternatively, you can use the IBM Integration API to configure the Long Description properties.

About this task

The Short Description is stored and used by the IBM Integration Explorer. The Long Description is stored by the integration node, and can be viewed within the IBM Integration Explorer or IBM Integration API. You can also use the Long Description to add keywords to your message flows.

To configure the Long Description and Short Description on your brokers, integration servers, or message flows:

Procedure

1. In the Navigator view, expand the Integration Nodes folder.
2. Right-click the object with which you want to work, and click **Properties**. This object can be an integration node, integration server, or message flow. The Properties window is displayed. The Long Description and Short Description are fields on the **General** tab.
3. Enter the values for the Long Description, Short Description, or both. If you are working with a message flow, you can add keywords into the Long Description field. When you define a keyword, you must follow certain rules to ensure that the information can be parsed. The following example shows some values that you might want to define in the Long Description property:

```
$MQSI Author=John Smith MQSI$  
$MQSI Flow 1 Version=v1.3.2 MQSI$
```

Do not use the following characters within keywords, because they cause unpredictable behavior:

```
^ $ . | \ < > ? + * = & [ ] ( )
```

You can use these characters in the values that are associated with keywords; for example:

- \$MQSI RCSVER=\$id\$ MQSI\$ is acceptable
 - \$MQSI \$name=Fred MQSI\$ is not acceptable
4. Click **Apply** to update the description properties.
 5. Click **OK** to exit the Properties window.

Results

The description properties for the selected object are now updated, and you can view these properties in the **Properties QuickView** for the object in the IBM Integration Explorer.

Related concepts:



IBM Integration Explorer

The IBM Integration Explorer is a graphical user interface based on the Eclipse platform for administering your integration nodes (brokers).



Message flow version and keywords

When you are developing a message flow, you can define the version of the message flow as well as other key information that you want to be associated with it.

Related tasks:

“Configuring integration node properties in the IBM Integration Explorer” on page 337

Integration nodes (brokers) and associated resource, such as integration servers and message flows, have properties that can be modified. Use the IBM Integration Explorer to configure the properties for your integration nodes, integration servers and message flows. Alternatively you can use the command line or the IBM Integration API.

“Configuring integration nodes using the IBM Integration Explorer” on page 336
Configure your local and remote integration nodes (brokers) by using the IBM Integration Explorer.

Related reference:



IBM Integration Explorer views

The IBM Integration Explorer can be used to manage and administer your integration nodes (brokers) and deployed resources.

Configuring DataPower security settings

Use the DataPower Security wizard in the IBM Integration Explorer to configure an external DataPower® appliance to handle the WS-Security Policy for your HTTP, HTTPS input, and SOAP nodes within your message flow. The DataPower box is configured to decrypt incoming messages to your flow and encrypt outgoing messages from your flow without requiring any changes to the message flows or broker configuration.

Before you begin

Before you start:

To use the DataPower Security wizard you must have access to the SYSTEM.DEF.SVRCONN channel on the broker's queue manager. Your clients must send their messages direct to the DataPower appliance on a Client port you specify.

About this task

The following steps are required to configure a DataPower appliance for WS-Security for your message flows:

- Select which HTTP(S)Input and SOAP nodes you want to configure your security for.
- Create a DataPower connection profile or edit an existing profile.
- Use or alter the default Policy Sets to specify your encryption and decryption WS- Security parameters.
- Specify which specific Crypto Keys to use from the DataPower box

On the DataPower appliance the following configuration is created after you run the DataPower Security wizard:

- An XML Firewall with optionally Back (for HTTPSInput Nodes) and Front (Client) SSL connection.
- An XML Firewall Policy consisting of a list of inbound/ request rules and an outbound/ response rule per HTTP Input or SOAP Node.
- Each inbound/ request rule consists of a decryption action with parameters specified from the Policy Set.
- Each outbound/ response rule consists of an encryption action with parameters specified from the Policy Set.

To configure DataPower security for your message flows:

Procedure

1. Right-click on the message flow or integration server with which you want to work, and click **Properties**. You can enable security handling on a single message flow containing HTTP, HTTPS input and SOAP nodes, or you can select an integration server to enable security handling for these nodes in all the message flows in the integration server.
2. In the Properties window, click **DataPower** on the left to open the DataPower tab.
3. Click **Configure Security** to open the Security on DataPower Appliance window. The HTTP, HTTPS input, and SOAP nodes from your message flows are displayed in the Flow Details table.
4. Select a Policy Set Binding from the list of options. If you select the No Policy Set Bindings option, no encryption or decryption nodes are specified in your policy rules. You can use this option as a test for the communication channels before applying a policy set binding. To create a policy set binding, click **Edit Policy Sets**. See Policy Sets and Policy Set Bindings editor, for more information about the Policy Sets and Policy Set Bindings editor.
5. In the DataPower details section, select a User profile from the list of options. Click **Edit Profiles** to create or edit connection profiles. To create a profile:

- a. In the DataPower Connection Profiles window, click **Add**.
- b. Click in the relevant cell in the table to edit the values. You must provide a valid user name, domain, and the host name of your DataPower appliance.
- c. Click **Finish**. The new or edited profile is now available to select in the Security on DataPower Appliance window.
- d. Add a password for the profile in the Password field.

You can also use the DataPower Connection Profiles window to import and export profiles in the IBM Integration Explorer on different machines.

6. You must now decide whether to create a new Policy or merge with an existing Policy. If you attempt to merge with a policy that does not exist, a new one is created. A merge adds request and response rules to your policy, but it does not overwrite any preexisting rules. A merge also does not alter your existing firewall settings.
7. Enter the name or names of your XML Firewalls, and the Client Ports on which your HTTP clients connect to your DataPower box.
8. Optional: Select the nodes to configure in the Flow Details section, and click **Next** to select XML Firewall SSL settings, Decryption, and Encryption rules for your DataPower device.
9. Click **Finish**. An attempt is made to connect to your domain on your DataPower box to retrieve your Crypto Profiles.
10. Click **Yes** to confirm that you want to alter the configuration of your DataPower appliance.

Results

You have configured DataPower security settings for your message flow or integration server.

Related concepts:

“WS-Security” on page 473

Web Services Security (WS-Security) describes enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication. WS-Security mechanisms can be used to accommodate a wide variety of security models and encryption technologies.

“Policy sets” on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

Related tasks:

“Configuring integration node properties in the IBM Integration Explorer” on page 337

Integration nodes (brokers) and associated resource, such as integration servers and message flows, have properties that can be modified. Use the IBM Integration Explorer to configure the properties for your integration nodes, integration servers and message flows. Alternatively you can use the command line or the IBM Integration API.

“Configuring integration nodes using the IBM Integration Explorer” on page 336
Configure your local and remote integration nodes (brokers) by using the IBM Integration Explorer.

Configuring the flow debug port in the IBM Integration Explorer

Before you can debug a message flow, you must configure and enable the flow debug port. The flow debug port is also known as the Java debug port, and the JVM debug port.

About this task

Use the IBM Integration Explorer, or IBM Integration Toolkit to set the flow debug port value to enable the flow debugger to be connected to an integration server. Alternatively, you can use the command console or the IBM Integration API to configure this property.

Using the IBM Integration Toolkit:

Procedure

1. Select the broker with which you want to work in the Integration Nodes view.
2. Right-click the integration server which you want to work, and click **Launch Debugger**. If the flow debug port is already configured, the port number is displayed in the Enable window.
3. To change the flow debug port, click **Configure**.
4. Set a port number for the debug port. The port number must be a number that is not already in use.
5. Click **OK**. The integration server is stopped and restarted, and the flow debug port is configured to use the specified port.
6. If you want to continue to attach the debugger to the integration server for debugging, click **OK**. To exit the Enable window without attaching the debugger, click **Cancel**.

What to do next

To start debugging your message flows, switch to the Debug perspective. Right-click the integration server, and click **Terminate Debugger** to stop debugging on the selected integration server.

Using the IBM Integration Explorer:

Procedure

1. Expand the Integration Nodes folder in the Navigator view.
2. Right-click the integration server which you want to work, and click **Properties**. The Properties window is displayed.
3. Click **Flow Debug Port** on the left to display the Flow Debug Port tab.
4. Set a port number for the debug port. The port number must be a number that is not already in use.
5. Click **Apply**.
6. Click **OK** to exit the Properties view.
7. Right-click the integration server, and click **Flow Debug Port > Enable** to enable debugging on the selected integration server.

What to do next

You can connect the flow debugger to your integration server by using the IBM Integration Toolkit, see the instructions in “Using the IBM Integration Toolkit.” Click **Flow Debug Port > Disable** to disable debugging on the selected integration server.

Related concepts:



IBM Integration Explorer

The IBM Integration Explorer is a graphical user interface based on the Eclipse platform for administering your integration nodes (brokers).



Flow debugger overview

Use the flow debugger in the IBM Integration Toolkit to track messages through your message flows.

Related tasks:

“Configuring integration node properties in the IBM Integration Explorer” on page 337

Integration nodes (brokers) and associated resource, such as integration servers and message flows, have properties that can be modified. Use the IBM Integration Explorer to configure the properties for your integration nodes, integration servers and message flows. Alternatively you can use the command line or the IBM Integration API.



Attaching the flow debugger to an integration server for debugging

Before you can debug your message flow, you must attach the flow debugger to the integration server where your flow is deployed, then start a debugging session.

Related reference:



JVM parameter values

Select the objects and properties associated with the Java Virtual Machine (JVM) that you want to change.

Changing the operation mode of the integration node using the IBM Integration Explorer

You must ensure that your integration nodes (brokers) are running in the operation mode for which you have purchased a license. You can change the operation mode of your integration node in IBM Integration Explorer. Alternatively, you can use the `mqsimode` command to change the operation mode.

About this task

To change the operation mode of your integration node by using the IBM Integration Explorer:

Procedure

1. Switch to the IBM Integration Explorer, and expand the Integration Nodes folder in the Navigator view.
2. Right-click the integration node with which you want to work, and click **Properties**. The Properties window is displayed.
3. Enter the value for the mode of your integration node in the Operation Mode field, in the **General** tab. The options you can set in this field are:
 - advanced
 - standard
 - adapter
 - express
 - scale
4. Click **Apply**.

5. Click **OK** to exit the Properties view.

Results

The operation mode of your integration node is updated. You can see the operation mode of your integration node in the Properties QuickView.

Related concepts:



IBM Integration Explorer

The IBM Integration Explorer is a graphical user interface based on the Eclipse platform for administering your integration nodes (brokers).

“Operation modes” on page 5

The operation mode that you can use for your broker is determined by the license that you purchase.

Related tasks:

“Configuring integration node properties in the IBM Integration Explorer” on page 337

Integration nodes (brokers) and associated resource, such as integration servers and message flows, have properties that can be modified. Use the IBM Integration Explorer to configure the properties for your integration nodes, integration servers and message flows. Alternatively you can use the command line or the IBM Integration API.

Related reference:



mqsi mode command

Use the **mqsi mode** command to configure and retrieve operation mode information.

Changing the integration node security settings in the IBM Integration Explorer

Use the IBM Integration Explorer to configure security settings for your integration node (broker). Alternatively, you can use the command line to change these values.

About this task

You can use the IBM Integration Explorer to configure the following security settings on your broker:

- Cache Sweep Interval
- Cache Timeout
- Security Profiles
- Policy Sets

Procedure

1. In the IBM Integration Explorer, expand the Integration Nodes folder in the Navigator view.
2. Right-click the integration node which you want to work with, and click **Properties**. The Properties window is displayed.
3. Click **Security** on the left to display the Security tab.
4. Enter a value in seconds for the **Cache Sweep Interval** field. The default value for this field is 300.
5. Enter a value in seconds for the **Cache Timeout** field. The default value for this field is 60.

6. Click **Apply** to update the Cache Sweep Interval and Cache Timeout properties.
7. Click **Security Profiles** to configure security profiles on your integration node. See *Creating a security profile*, for more information about the creating Security Profiles.
8. Click **Policy Sets** See Policy Sets and Policy Set Bindings editor, for more information about the Policy Sets and Policy Set Bindings editor.
9. Click **OK** to exit the Properties view.

Results

You have configured security settings for your integration node.

Related concepts:



IBM Integration Explorer

The IBM Integration Explorer is a graphical user interface based on the Eclipse platform for administering your integration nodes (brokers).



Security profiles

A security profile defines the security operations that are to be performed in a message flow at SecurityPEP nodes and security enabled input and output nodes.

“Policy sets” on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

Related tasks:

“Configuring integration node properties in the IBM Integration Explorer” on page 337

Integration nodes (brokers) and associated resource, such as integration servers and message flows, have properties that can be modified. Use the IBM Integration Explorer to configure the properties for your integration nodes, integration servers and message flows. Alternatively you can use the command line or the IBM Integration API.

Related reference:



IBM Integration Explorer views

The IBM Integration Explorer can be used to manage and administer your integration nodes (brokers) and deployed resources.



Parameter values for the securitycache component

Select the objects and properties associated with the securitycache component that you want to change.

Using the IBM Integration Explorer to work with configurable services

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Instead of defining properties on the message flow node or message flow, you can create configurable services so that message flow nodes and message flows can refer to them to find properties at run time. If you use this method, you can

change the values of attributes for a configurable service on the integration node, which then affects the behavior of a message flow node or message flow without the need for redeployment.

For most types of configurable service, your changes will take effect from the next time a message flow that uses the configurable service is called. For some types of configurable service, for which it is explicitly stated in the documentation, you must stop and start the integration server for the change to take effect.

You can create and name new configurable services, based on IBM defined templates. Alternatively, you can modify the existing IBM defined configurable services. If you modify an IBM defined configurable services, their default values are overwritten in the system registry.

Use the IBM Integration Explorer to complete the following tasks for configurable services:

- “Creating a new configurable service” on page 347
- “Viewing configurable services” on page 348
- “Modifying an IBM defined configurable service” on page 349
- “Modifying a configurable service” on page 350
- “Importing a configurable service” on page 351
- “Exporting a configurable service” on page 352
- “Deleting a configurable service” on page 353

Alternatively, you can use the runtime commands to work with configurable services, see Configurable services.

For a full list of configurable services and their properties, see Configurable services properties.

Related concepts:



Configurable services

Configurable services are typically runtime properties. You can use them to define properties that are related to external services on which the broker relies; for example, an SMTP server or a JMS provider.

Related reference:



mqsicreateconfigurablecommand command

Use the **mqsicreateconfigurablecommand** command to create an object name for a broker external resource.



mqsideleteconfigurablecommand command

Use the **mqsideleteconfigurablecommand** command to delete a configurable service, such as a JMS provider, JDBC provider, or FTP server, that you have created by using the **mqsicreateconfigurablecommand** command.



mqsichangepropertiescommand command

Use the **mqsichangepropertiescommand** command to modify broker properties and properties of broker resources.



mqsireportpropertiescommand command

Use the **mqsireportpropertiescommand** command to display properties that relate to a broker, an integration server, or a configurable service.



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

Creating a new configurable service

Use the IBM Integration Explorer to create a new configurable service to define properties for an external service on which the integration node (broker) relies.

About this task

You can add a new configurable service to your integration node based on an IBM defined configurable service template using the IBM Integration Explorer. You must provide a name for your configurable service, which must not duplicate an existing configurable service name.

As an alternative to creating a new configurable service, you can modify an IBM defined configurable service, see “Modifying an IBM defined configurable service” on page 349.

Not all configurable service types are available to select within IBM Integration Explorer. If you want to create a configurable service and the type is not listed, you must first create an instance of that type via the command line. For information about creating configurable services by using the command line, see `mqsicreateconfigurableservice` command.

To add a new configurable service using the IBM Integration Explorer:

Procedure

1. In the Navigator view, expand the integration node on which you want to add a new configurable service.
2. Right-click the Configurable Services folder and click **New > Configurable service**. The Configurable services window is displayed.
3. Enter a name for your configurable service. When naming your JDBCProvider service, consider the following requirements:
 - If you want to use your JDBCProvider service with a JavaCompute node, or with a Java user-defined node, the name of your JDBCProvider service must match the `datasourceName` parameter in the `getJDBCType4Connection()` call from the node.
 - If you want to use your JDBCProvider service with a Mapping node, the name of your JDBCProvider service must match the database name that is used by the database transforms in your Graphical Data Map. For each database transform, the database name is determined by the database definition (.dbm file) in the Data Design project that was used to create the map.
 - If you want to use your JDBCProvider service with a DatabaseRetrieve node, or with a DatabaseRoute node, the name of your JDBCProvider service must match the value of the `Data source name` property of the node.
4. Select the type of configurable service to create.
5. For some configurable services, you can select an IBM defined template to provide default values you can use or update. If appropriate, select the IBM defined template to use for your configurable service.
6. Enter values for the configurable service properties. The properties are populated with default values.

7. Click **OK** to create the new configurable service.

What to do next

For most types of configurable service, your changes will take effect from the next time a message flow that uses the configurable service is called. For some types of configurable service, for which it is explicitly stated in the documentation, you must stop and start the integration server for the change to take effect. For more information, see Configurable services properties.

Related concepts:

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:

“Viewing configurable services”

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view the properties of configurable services defined on your integration node.

“Modifying an IBM defined configurable service” on page 349

You can create a new configurable service for an external service on which the broker relies, by modifying predefined configurable services provided by IBM. Use the IBM Integration Explorer to view and modify existing configurable services.

“Modifying a configurable service” on page 350

Use the IBM Integration Explorer to view and modify existing configurable services.

“Deleting a configurable service” on page 353

Use the IBM Integration Explorer to delete custom configurable services.

Related reference:



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

Viewing configurable services

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view the properties of configurable services defined on your integration node.

About this task

You can view the properties of configurable services that you have defined on the integration node, or you can view the properties of IBM defined configurable services.

To view a configurable service:

Procedure

1. In the Navigator view, expand the integration node on which you want to view the configurable services.

2. Expand the Configurable Services folder. If you want to view IBM defined configurable services, right-click the Configurable Services folder, and click **Show IBM Templates**.
3. Right-click the configurable service that you want to view, and click **Properties**. The Configurable services window is displayed, and shows the properties for the selected configurable service.
4. Click **OK** to close the Configurable services.

Related concepts:

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:

“Creating a new configurable service” on page 347

Use the IBM Integration Explorer to create a new configurable service to define properties for an external service on which the integration node (broker) relies.

“Modifying a configurable service” on page 350

Use the IBM Integration Explorer to view and modify existing configurable services.

“Modifying an IBM defined configurable service”

You can create a new configurable service for an external service on which the broker relies, by modifying predefined configurable services provided by IBM. Use the IBM Integration Explorer to view and modify existing configurable services.

“Deleting a configurable service” on page 353

Use the IBM Integration Explorer to delete custom configurable services.

Related reference:

 Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

Modifying an IBM defined configurable service

You can create a new configurable service for an external service on which the broker relies, by modifying predefined configurable services provided by IBM. Use the IBM Integration Explorer to view and modify existing configurable services.

About this task

You can modify IBM defined configurable services to add a new configurable service to your broker. You cannot change the name of an IBM defined configurable service, and you cannot delete it. If you want to use a custom name for a configurable service, or if you want to create different configurable services of the same type, see “Creating a new configurable service” on page 347.

To modify an IBM defined configurable service:

Procedure

1. In the Navigator view, expand the broker on which you want to modify an IBM defined configurable service.
2. Expand the Configurable Services folder.
3. Right-click the configurable service that you want to modify, and click **Properties**. The Configurable services window is displayed.

4. Enter values for the configurable service properties. The **Basic** tab shows the key properties for the configurable service. Additional properties to configure can be found on the **Advanced** tab. The properties are populated with default values.
5. Click **OK** to update the configurable service.

What to do next

For most types of configurable service, your changes will take effect from the next time a message flow that uses the configurable service is called. For some types of configurable service, for which it is explicitly stated in the documentation, you must stop and start the integration server for the change to take effect. For more information, see Configurable services properties.

Related concepts:

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:

“Viewing configurable services” on page 348

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view the properties of configurable services defined on your integration node.

“Creating a new configurable service” on page 347

Use the IBM Integration Explorer to create a new configurable service to define properties for an external service on which the integration node (broker) relies.

“Modifying a configurable service”

Use the IBM Integration Explorer to view and modify existing configurable services.

“Deleting a configurable service” on page 353

Use the IBM Integration Explorer to delete custom configurable services.

Related reference:



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

Modifying a configurable service

Use the IBM Integration Explorer to view and modify existing configurable services.

About this task

You can modify all the properties of an existing configurable service, except for the name.

To modify a configurable service:

Procedure

1. In the Navigator view, expand the broker on which you want to modify a configurable service.

2. Expand the Configurable Services folder. If you want to modify an IBM defined configurable service that is not displayed, right-click the Configurable Services folder, and click **Show IBM Templates**. The predefined services are displayed in the Configurable Services folder.
3. Right-click the configurable service that you want to modify, and click **Properties**. The Configurable services window is displayed.
4. Enter values for the configurable service properties. The properties are populated with default values.
5. Click **OK** to update the configurable service.

What to do next

For most types of configurable service, your changes will take effect from the next time a message flow that uses the configurable service is called. For some types of configurable service, for which it is explicitly stated in the documentation, you must stop and start the integration server for the change to take effect. For more information, see Configurable services properties.

Related concepts:

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:

“Viewing configurable services” on page 348

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view the properties of configurable services defined on your integration node.

“Creating a new configurable service” on page 347

Use the IBM Integration Explorer to create a new configurable service to define properties for an external service on which the integration node (broker) relies.

“Modifying an IBM defined configurable service” on page 349

You can create a new configurable service for an external service on which the broker relies, by modifying predefined configurable services provided by IBM. Use the IBM Integration Explorer to view and modify existing configurable services.

“Deleting a configurable service” on page 353

Use the IBM Integration Explorer to delete custom configurable services.

Related reference:



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

Importing a configurable service

Use the IBM Integration Explorer to import a configurable service from another integration node (broker).

About this task

The properties of a configurable service can be exported from an integration node, and imported by a new integration node. The properties of the configurable service are stored in a `.configurableservice` file.

If the configurable service is defined on an integration node in the IBM Integration Explorer, you can drag the configurable service from the integration node to a new integration node.

Alternatively, to import a configurable service from a `.configurableservice` file:

Procedure

1. In the Navigator view, expand the integration node to which you want to import a configurable service.
2. Right-click the Configurable Services folder, and click **Import** `*.configurableservice`.
3. Click the configurable service files that you want to import, and click **OK**.

What to do next

You must restart the message flows that used the configurable service for the change to be recognized by the message flows. You can stop and start the integration node to ensure that the configurable service is updated for all the resources running on the integration node.

Related concepts:

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:

“Viewing configurable services” on page 348

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view the properties of configurable services defined on your integration node.

“Creating a new configurable service” on page 347

Use the IBM Integration Explorer to create a new configurable service to define properties for an external service on which the integration node (broker) relies.

“Modifying an IBM defined configurable service” on page 349

You can create a new configurable service for an external service on which the broker relies, by modifying predefined configurable services provided by IBM. Use the IBM Integration Explorer to view and modify existing configurable services.

“Modifying a configurable service” on page 350

Use the IBM Integration Explorer to view and modify existing configurable services.

“Exporting a configurable service”

Use the IBM Integration Explorer to export a configurable service to use on another integration node (broker).

Related reference:



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

Exporting a configurable service

Use the IBM Integration Explorer to export a configurable service to use on another integration node (broker).

About this task

The properties of a configurable service can be exported from an integration node, and imported by a new integration node. The properties of the configurable service are stored in a `.configurableservice` file.

If the configurable service is defined on an integration node in the IBM Integration Explorer, you can drag the configurable service from the integration node to a new integration node.

Alternatively, to export a configurable service to a `.configurableservice` file:

Procedure

1. In the Navigator view, expand the integration node to which you want to import a configurable service.
2. Expand the Configurable Services folder.
3. Right-click the configurable service that you want to export, and then click **Export *.configurableservice**.
4. Enter a name for the `.configurableservice` file, and click **OK**.

Related concepts:

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:

“Viewing configurable services” on page 348

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view the properties of configurable services defined on your integration node.

“Creating a new configurable service” on page 347

Use the IBM Integration Explorer to create a new configurable service to define properties for an external service on which the integration node (broker) relies.

“Modifying an IBM defined configurable service” on page 349

You can create a new configurable service for an external service on which the broker relies, by modifying predefined configurable services provided by IBM. Use the IBM Integration Explorer to view and modify existing configurable services.

“Modifying a configurable service” on page 350

Use the IBM Integration Explorer to view and modify existing configurable services.

“Importing a configurable service” on page 351

Use the IBM Integration Explorer to import a configurable service from another integration node (broker).

Related reference:



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

Deleting a configurable service

Use the IBM Integration Explorer to delete custom configurable services.

About this task

You can delete custom configurable services that you have defined on an integration node (broker), but you cannot delete the IBM defined configurable services on an integration node, even if you have previously modified the IBM defined configurable service.

To delete a configurable service:

Procedure

1. In the Navigator view, expand the integration node on which you want to delete a configurable service.
2. Expand the Configurable Services folder.
3. Right-click the configurable service that you want to delete, and click **Delete**.
4. Click **OK** to confirm you want to delete the configurable service.

What to do next

For most types of configurable service, your changes will take effect from the next time a message flow that uses the configurable service is called. For some types of configurable service, for which it is explicitly stated in the documentation, you must stop and start the integration server for the change to take effect.

Related concepts:

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:

“Viewing configurable services” on page 348

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view the properties of configurable services defined on your integration node.

“Creating a new configurable service” on page 347

Use the IBM Integration Explorer to create a new configurable service to define properties for an external service on which the integration node (broker) relies.

“Modifying an IBM defined configurable service” on page 349

You can create a new configurable service for an external service on which the broker relies, by modifying predefined configurable services provided by IBM. Use the IBM Integration Explorer to view and modify existing configurable services.

“Modifying a configurable service” on page 350

Use the IBM Integration Explorer to view and modify existing configurable services.

Related reference:



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

Working with UserDefined configurable services

Use the IBM Integration Explorer to create, view, and modify a UserDefined configurable service.

About this task

For further information on using the IBM Integration Explorer with configurable services, see:

- “Creating a new configurable service” on page 347
- “Viewing configurable services” on page 348
- “Modifying a configurable service” on page 350

To add a new configurable service:

Procedure

1. Switch to the WebSphere MQ Explorer perspective.
2. In the Domains view, expand the integration node (broker) on which you want to add a new configurable service.
3. Right-click the Configurable Services folder and click **New > Configurable service**. The Configurable Service window is displayed.
4. Enter a name for your configurable service.
5. Select UserDefined as the type of configurable service to create.
6. Enter the values you require for the configurable service properties, using the add and delete buttons that become enabled for UserDefined configurable services. There are no predefined values for a UserDefined configurable service.
7. Click **Finish** to create the new configurable service.

Related concepts:

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:

“Viewing configurable services” on page 348

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view the properties of configurable services defined on your integration node.

“Creating a new configurable service” on page 347

Use the IBM Integration Explorer to create a new configurable service to define properties for an external service on which the integration node (broker) relies.

“Modifying a configurable service” on page 350

Use the IBM Integration Explorer to view and modify existing configurable services.

“Deleting a configurable service” on page 353

Use the IBM Integration Explorer to delete custom configurable services.



Working with properties of a configurable service of type UserDefined at run time in a JavaCompute node

Use the CMP in a JavaCompute node to query, set, create, and delete properties dynamically at run time in configurable services that you have defined with type UserDefined.

Related reference:



Configurable services properties

The supplied configurable services, and the configurable services that you create,

are defined by their names and properties. You can use the supplied services.

Changing IBM Integration Explorer preferences

Change preferences in the IBM Integration Explorer.

About this task

You can change the following preferences to tailor how you work in the IBM Integration Explorer:

- Display resources in the Navigator view.
- Warn before deleting Administration log events.
- Integration node connection wait times.
- DataPower configuration properties.
- Service trace properties
- SSL parameters.

To change IBM Integration Explorer preferences:

Procedure

1. Start the IBM Integration Explorer.
2. Click **Window > Preferences**.
3. Expand the **IBM Integration** category in the left pane.
4. Make your selections.
5. Click **OK**.

Related concepts:



IBM Integration Toolkit

The IBM Integration Toolkit is an integrated development environment and graphical user interface based on the Eclipse platform.



IBM Integration Explorer

The IBM Integration Explorer is a graphical user interface based on the Eclipse platform for administering your integration nodes (brokers).

Related tasks:

“Configuring the Toolkit” on page 280

You can configure various settings in the Toolkit to suit your requirements and your working environment.



Changing trace settings from the IBM Integration Explorer

Collect trace information, in addition to user and service trace, by selecting options in the IBM Integration Explorer.

Changing the operation mode of your broker

Change the operation mode in which your broker is working by using the `mqs imode` command.

About this task

You must change your broker configuration to ensure that your brokers are running in the operation mode for which you have purchased a license. You can change the mode to standard, adapter, express, scale, or advanced.

When you view the broker in the IBM Integration Explorer, the current mode of each broker is displayed. If the resources deployed to a broker exceed the permitted amounts, the display indicates these violations. You can change the broker mode in the IBM Integration Explorer; for more information, see “Changing the operation mode of the integration node using the IBM Integration Explorer” on page 343. Alternatively, you can use the **mqs imode** command to make any required changes.

Procedure

1. Run the **mqs imode** command with the **-o** parameter to change the mode of the broker, or without the **-o** parameter to view the current setting; for more information, see **mqs imode** command.
2. Check for error messages. If you attempt to reconfigure the broker to a mode that is not sufficient for the deployed resources, the **mqs imode** command issues a warning indicating that changing the mode is not allowed. Resolve any violations, if required; for more information, see Resolving problems that occur during deployment of message flows.
3. (Optional) Run the **mqs imode** command again to confirm that there are no violations.

Example

For more information, see Example: Changing the operation mode of your broker.

What to do next

Related concepts:

“Operation modes” on page 5

The operation mode that you can use for your broker is determined by the license that you purchase.

Related tasks:

“Checking the operation mode of your broker” on page 359

Run the **mqs imode** command to report the mode that is being used by your broker.

“Moving from Developer Edition” on page 358

You can move from IBM Integration Bus for Developers (Developer Edition) to an alternative edition, by purchasing the required license and then changing the operation mode of your existing brokers to the mode that conforms to your license.

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

Related reference:

“Restrictions that apply in each operation mode” on page 10

The operation mode in which your broker is working defines how many integration servers you can use, and which nodes are available.

mqs imode command

Use the **mqs imode** command to configure and retrieve operation mode information.

mqs ichangebroker command

Use the **mqs ichangebroker** command to change one or more of the configuration parameters of the broker.

“License requirements” on page 641

Use the reference information in this section to understand license requirements.

Moving from Developer Edition

You can move from IBM Integration Bus for Developers (Developer Edition) to an alternative edition, by purchasing the required license and then changing the operation mode of your existing brokers to the mode that conforms to your license.

Before you begin

Before you start: Contact your IBM representative to upgrade your license.

About this task

When you have installed IBM Integration Bus for Developers, all brokers are created in developer mode by default. If you want to move to an alternative edition, you must change the operation mode for each of these brokers by following the steps described below. If you want to create brokers with a default mode of operation other than developer, you must uninstall the IBM Integration Bus for Developers product and install the full IBM Integration Bus product.

Follow these steps to move your existing brokers from Developer Edition to an alternative edition:

Procedure

1. Change the operation mode of each broker by using either the **mqsimode** command or the IBM Integration Explorer:
 - Use the **mqsimode** command by following these steps:
 - a. Open a command prompt:
 - **Linux** On Linux, run the **mqsiprofile** command to initialize the command environment.
 - **Windows** On Windows, click **Start > All Programs > IBM Integration Bus 9.0.0.0 > IBM Integration Console 9.0** to open a command console.
 - b. Use the **mqsimode** command to change the mode of each broker. Specify the mode that you require for each broker by setting the **-o** parameter (*Mode_Type*) to advanced, standard, express, scale, or adapter. For example, to change a local broker to run in advanced mode, run the following command:

```
mqsimode IB9NODE -o advanced
```

where IB9NODE is the name of your broker.
To change a remote broker, specify a configuration file that defines the broker (*brokername.broker*), or one or more connection parameters that identify the broker **-i**, **-p**, **-q**.
 - Use the IBM Integration Explorer by following these steps:
 - a. Start the IBM Integration Explorer.
 - b. In the navigator pane, right-click the required node (broker) and then select **Properties**. In the Properties pane, the current operation mode of the broker is displayed.
 - c. Select the required operation mode from the list and click **OK**.

2. Restart all your brokers; for more information, see Starting and stopping a broker.

Related concepts:

“Operation modes” on page 5

The operation mode that you can use for your broker is determined by the license that you purchase.

Related tasks:

“Checking the operation mode of your broker”

Run the `mqsimode` command to report the mode that is being used by your broker.

“Changing the operation mode of your broker” on page 356

Change the operation mode in which your broker is working by using the `mqsimode` command.

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

Related reference:

“Restrictions that apply in each operation mode” on page 10

The operation mode in which your broker is working defines how many integration servers you can use, and which nodes are available.



mqsimode command

Use the `mqsimode` command to configure and retrieve operation mode information.

Checking the operation mode of your broker

Run the `mqsimode` command to report the mode that is being used by your broker.

About this task

Use the connection parameters to identify the broker that you want to check.

Example

To report the mode that is being used by your broker, run the following command:

```
mqsimode -i host -p port -q qmgr
```

where *host* is the IP address or host name, *port* is the port number, and *qmgr* is the name of the queue manager that is associated with your broker. Alternatively, if you created the broker on the computer on which you run this command, you can specify the broker name instead. For example:

```
mqsimode broker_name
```

where *broker_name* is the name of your broker.

Here is a sample response message for this command:

```
BIP1044: Connecting to the broker's queue manager...  
BIP1807: Discovering mode information from broker 'IB9NODE'...  
BIP1802: Broker 'IB9NODE' is in 'standard' mode.  
BIP8071: Successful command completion.
```

Running this command also reports all mode violations, see “Restrictions that apply in each operation mode” on page 10. For example:

BIP1821W: WARNING: Broker 'IB9NODE' is in 'express' mode but has '2' execution groups, which exceeds the allowed maximum for this mode.

Related concepts:

“Operation modes” on page 5

The operation mode that you can use for your broker is determined by the license that you purchase.

Related tasks:

“Changing the operation mode of your broker” on page 356

Change the operation mode in which your broker is working by using the **mqsimode** command.

Related reference:



mqsimode command

Use the **mqsimode** command to configure and retrieve operation mode information.

Advanced configuration

Change your broker configuration to use one or more of the advanced options available.

About this task

- If you are developing message flows that include resources such as the WebSphere Adapters nodes, IMS nodes, or the CICSRequest node, you must set up your broker environment to support these applications. Details of the setup that you require is in “Configuring for applications” on page 361.
- If you are working with web services message flows and want to ensure reliable messaging, you can configure WS-RM using policy sets. See “Web Services Reliable Messaging” on page 469.
- You can use WS-Security with your web services message flows to provide quality of protection through message integrity, message confidentiality, and single message authentication. See “WS-Security” on page 473.
- If you have been working with IBM Integration Bus on z/OS, and want to move your configuration to a distributed platform, see “Moving from IBM Integration Bus on a distributed system to z/OS” on page 535.
- IBM Integration Bus supports a number of locales on each platform; details of what is supported, and how to use code converters, are provided in “Changing locales” on page 536.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.

“Changing the operation mode of your broker” on page 356

Change the operation mode in which your broker is working by using the **mqsimode** command.

Configuring for applications

If you are developing message flows that access external resources such as databases, Enterprise Information Systems, IMS, CICS® Transaction Server for z/OS, or email servers, you must set up your broker environment to support these applications.

About this task

- “Configuring databases”
- “Configuring global coordination of transactions (two-phase commit)” on page 397
- “Configuring properties to connect to external resources” on page 416
- “Configuring internal resources required by flows” on page 455

Related tasks:

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.



Connecting to Enterprise Information Systems

Use WebSphere Adapters to communicate with Enterprise Information Systems (EIS) such as SAP, Siebel, PeopleSoft, and JD Edwards.



Working with IMS

You can use the IMSRequest node to connect to IMS, a message-based transaction manager and hierarchical-database manager for z/OS.



Working with CICS Transaction Server for z/OS

Use the CICSRequest node to connect to CICS Transaction Server for z/OS applications.



Processing email messages

You can configure the EmailOutput node to deliver an email from a message flow to an email server that supports Simple Mail Transfer Protocol (SMTP). You can also configure the EmailInput node to retrieve an email from an email server that supports Post Office Protocol 3 (POP3) or Internet Message Access Protocol (IMAP).

Configuring databases

Configure databases to hold application or business data that you can access from your message flows.

About this task

Databases that hold application or business data are read from and written to by nodes within the message flows that you deploy to one or more brokers in your domain.

In some situations, and for some applications, you might need to ensure the integrity of the data that you hold in databases across multiple systems and resource managers by coordinating table updates and the writing to one database with the deletion of data in another. To achieve these goals, you must configure your databases, your brokers, and your message flows to be globally coordinated.

For more information about the requirement for, and set up of, databases, and the restrictions that apply, see Databases overview.

The process of making databases available has the following phases:

1. Optional: Create and configure databases. If your message flows interact with databases, you must create those databases and configure ODBC and JDBC connections for them. Broker can then connect to the databases on behalf of the message flows.
2. Optional: If your databases contain critical information, coordinate their updates through a transaction manager.

On distributed systems, the WebSphere MQ queue manager is the transaction manager that interacts with the resource managers (the database providers). On z/OS, RRS provides equivalent coordination.

To complete these phases:

Procedure

1. If you want to access databases from your deployed message flows, create and configure the databases and the connections to them:
 - a. Create the databases.
 - b. Authorize access to the databases.
 - c. Optional: If you want your databases to participate in globally coordinated transactions, configure the databases for global coordination.
2. On distributed platforms, create and configure connections to the databases that you have created:
 - a. If your message flows use an ODBC connection to a database, enable an ODBC connection for that database. Repeat this step for each database that you want to access in this way.

Note that you can use the `mqsicvp` command as an ODBC test tool; see “Enabling ODBC connections to the databases” on page 369 for further information.
 - b. If your message flows use a JDBC connection to a database, enable a JDBC connection for that database. Repeat this step for each database that you want to access in this way.
3. On z/OS, connect to the database.
4. Optional: If you want your databases to participate in globally coordinated transactions, configure the environment for global coordination.

Related tasks:

“Configuring brokers” on page 303

Create and configure the brokers that you want on the operating system of your choice.

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

Securing database connections:

Set up security for a database connection, whether it is required by the database provider or optional.

Before you begin

- If you are using a JDBC provider, you must first set up the JDBC provider. See [Set up your JDBC provider definition](#).
- If you are using an ODBC connection, you must first connect to the database. See [“Enabling ODBC connections to the databases”](#) on page 369.

About this task

Some databases require that all access is associated with a known user ID, for others this association is optional. For example, DB2 requires a data source login name and password on all connections.

Use the **mqsisetdbparms** command to specify a user ID and password that the integration node can use to access each database. The values for user ID and password are platform specific:

1. **Windows** On Windows: The integration node service ID and password that you specified on the **mqsicreatebroker** command.
2. **Linux** **UNIX** On Linux and UNIX: The user ID `mqsidUser` and password `*****` (these values are fixed).
3. **z/OS** On z/OS: The user ID `"` and password `"`.

If you migrated your integration node from a previous release, the integration node accessed a database for its own use. You might have defined the user ID and password used to access that database by specifying a database connection user ID and password with the **-u** and **-p** parameters on the **mqsicreatebroker** command. Alternatively, you might have used the integration node service user ID and its password (specified with the **-i** and **-a** parameters on the same command). When you migrate the integration node, these parameters are migrated and stored, and are used by the migrated integration node for access to databases that do not have specific ID access defined. If you want to check what user ID you set, or test that the password for a specific user ID is what you expect, you can use the **mqsireportdbparms** command.

Steps for setting up security are specific to the type of database connection that you are using. Choose your database connection type to see the steps:

- [“ODBC connections”](#)
- [“JDBC connections”](#) on page 364

ODBC connections:

About this task

If your ODBC data source requires you to define secure access, or if you want to implement security where this is optional, complete the following steps:

Procedure

1. Identify the user IDs that you want to associate with the database connection, or create a user ID with a password, following the appropriate instructions for your operating system and database.
2. Define the user IDs and passwords that the integration node can use to access a particular data source.
3. Run the **mqsisetdbparms** command to create user IDs and passwords that can be used to access the data source from an integration node. Use the following format:

```
mqsisetdbparmsbroker_name -n data_source_name -u
database_userID -p database_userID_password
```

Results

You have secured access to your ODBC data source.

JDBC connections:

About this task

If your JDBC database requires you to define secure access, or if you want to implement security where this is optional, complete the following steps:

Procedure

1. Identify the user ID that you want to associate with the database connection, or create a user ID with a password, following the appropriate instructions for your operating system and database.
2. Define a default user ID and password that the integration node can use if you do not define specific values for a particular database.

- a. Run the **mqsisetdbparms** command to associate the user ID and password with the security identity that is associated with the database. Use the following command format:

```
mqsisetdbparms broker_name -n security_identity -u userID -p password
```

- b. Specify the database connection type in the **-n** parameter. For example, if you want user ID `myuserid` with a password of `secretpw` to access a database on integration node `INODE1`, run the following command:

```
mqsisetdbparms INODE1 -n jdbc::mySecurityIdentity -u myuserid -p secretpw
```

In the example, the `mySecurityIdentity` is prefixed with `jdbc::` to indicate the type of the connection for which the user ID and password are defined.

- c. Optional: If you want to use the same user ID and password for more than one database, you can specify a resource name on this command, as shown in the following example:

```
mqsisetdbparms INODE1 -n jdbc::JDBC -u defaultuid -p defaultpw
```

3. Update the corresponding **securityIdentity** property for the configurable service to associate the connection with the security identity that you have defined. Use the following command format:

```
mqsichangeproperties broker_name -c JDBCProviders -o service_name -n securityIdentity -v security_identity
```

For example, if you are using the supplied JDBCProvider definition for Oracle:

```
mqsichangeproperties INODE1 -c JDBCProviders -o Oracle -n securityIdentity -v mySecurityIdentity
```

Results

You have secured access to your JDBC databases. If you need to define user credentials that can be shared across a business area or account, you can reuse the same security identity that you defined in the previous steps in different JDBCProviders configurable services.

Related concepts:



Databases overview

Brokers can manipulate business data that is stored in databases. If you have databases, you must configure them before you can access them from your message flow.

Related tasks:

“Setting up a JDBC provider for type 4 connections” on page 386

Use the **mqsicreateconfigurableservice** or the **mqsichangeproperties** command to configure a JDBC provider service.

“Configuring a JDBC type 4 connection for globally coordinated transactions” on page 391

If you want the database that you access through a JDBC type 4 connection to participate in globally coordinated transactions, set up the appropriate environment.

“Authorizing access to JDBC type 4 JDBCProvider resources on Windows” on page 394

Authorize the broker and queue manager to access shared resources that are associated with the JDBCProvider. This task is required only if you want the database updates to be included in globally coordinated transactions on Windows systems.



Interacting with databases by using the JavaCompute node

Access databases from Java code included in the JavaCompute node.



Extending the capability of a Java message processing or output node

Within a message processing or output node, you can add extended functions to your Java node.

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.



mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.



mqsicreateconfigurableservice command

Use the **mqsicreateconfigurableservice** command to create an object name for a broker external resource.



mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.



mqsisetdbparms command

Use the **mqsisetdbparms** command to associate a specific user ID and password (or SSH identity file) with one or more resources that are accessed by the broker.



mqsireportdbparms command

Use the **mqsireportdbparms** command to list all parameters that are set for a specific broker.

Configuring databases for global coordination of transactions:

If your message flow interacts with a database, and you want to globally coordinate the updates made to the database with other actions within the message flow, configure your databases for global coordination.

Before you begin

Before you start: Create your database and authorize access to it.

About this task

If you restart a database while the broker is still running, you must also restart the broker. The broker cannot detect that the database has stopped, and WebSphere MQ therefore retains its old connections to the database. When the database starts again, the broker tries, and fails, to use these connections.

To configure databases for coordinated message flows, follow the instructions relevant to your database manager:

- DB2
- Oracle
- Sybase

Configuring DB2 for global coordination of transactions:

Before you begin

You must complete these steps for databases that you connect to with an ODBC or a JDBC connection.

You must have database administrator (DBA) privileges to perform the following tasks.

About this task

To configure DB2 database instances for global coordination of transactions:

Procedure

1. **Windows** **Linux** Windows and Linux on x86 systems only: for each 32-bit instance that is involved in the global coordination, run the following commands to set the Transaction Process Monitor name (TP_MON_NAME) to MQ:

```
db2 update dbm cfg using TP_MON_NAME MQ
db2stop
db2start
```

UNIX **Linux** On Linux and UNIX systems (except for Linux on x86), do not set this variable for 32-bit or 64-bit instances.

2. Ensure that you have adequate connection resources and find out from the broker administrator whether the broker uses TCP/IP or shared memory to connect to databases.

To use TCP/IP connections, see the example in the section about message SQL1224N in Resolving problems when using databases.

To enable extended shared memory:

- a. On the DB2 server, run the following commands:

```
export EXTSHM=ON
db2set DB2ENVLIST=EXTSHM
db2stop
db2start
```

- b. Ensure that shared memory support is enabled in the broker environment. For more information, see “Configuring global coordination with DB2” on page 400.
3. If you are connecting a broker on a distributed platform to a DB2 instance on z/OS, you must configure DB2 Connect™ to enable support for global coordination. Ensure that you have already configured a DB2 alias to represent the database by using DB2 Connect.

Perform the following tasks on the system that hosts the broker:

- a. Turn on the Connection Concentrator by configuring the DB2 database manager configuration parameters so that the value of the **MAX_CONNECTIONS** parameter is greater than the value of the **MAX_COORDAGENTS** parameter:

```
db2 update dbm cfg using MAX_CONNECTIONS max_connections_value
```

where *max_connections_value* is greater than the existing value of the **MAX_COORDAGENTS** parameter.

- b. Define the SPM name as the name of the system that hosts the broker:

```
db2 update dbm cfg using SPM_NAME host_name
```

where *host_name* is the TCP/IP name of the system that hosts the broker.

- c. Stop, then restart DB2 on the system that hosts the broker to apply the changes:

```
db2stop  
db2start
```

DB2 Connect is now configured to enable global coordination of message flows that are deployed to the broker (on a distributed platform) and that access DB2 on z/OS.

Results

The DB2 database instances are now configured for global coordination.

What to do next

Next: See “Configuring ODBC connections for globally coordinated transactions” on page 399.

Configuring Oracle for global coordination of transactions:

Before you begin

You must complete these steps for databases that you connect to with an ODBC connection only.

You must have database administrator (DBA) privileges to perform the following tasks.

About this task

To configure Oracle databases for global coordination of transactions:

Procedure

1. Ensure that the JAVA_XA package is present on the Oracle database by using, for example, the following Oracle SQLPLUS command:

```
describe JAVA_XA;
```

For more information, see the Oracle product documentation.

2. Ensure that the user ID that the broker uses to access the database has the necessary Oracle privileges to access the DBA_PENDING_TRANSACTIONS view. You can grant the required access by using, for example, the following Oracle SQLPLUS command:

```
grant select on DBA_PENDING_TRANSACTIONS to userid;
```

If more than one user ID is involved (for example, if IBM Integration Bus and WebSphere MQ run under different user IDs), you also need the privilege FORCE ANY TRANSACTION.

Results

The Oracle databases are now configured for global coordination.

What to do next

Next: See “Configuring ODBC connections for globally coordinated transactions” on page 399.

Configuring Sybase for global coordination of transactions:

Before you begin

You must complete these steps for databases that you connect to with an ODBC connection only.

You must have database administrator (DBA) privileges to perform the following tasks.

Procedure

To configure Sybase databases for global coordination of transactions, ensure that the user ID that the broker uses to access the database has been granted the Sybase role of dtm_tm_role.

Results

The Sybase databases are now configured for global coordination.

What to do next

Next: See “Configuring ODBC connections for globally coordinated transactions” on page 399.

Related concepts:



The transactional model

The transactional model describes the way in which you can use transactions in message flows to accomplish certain tasks and results.

Related tasks:



Configuring transactionality for message flows

A message flow runs in a single transaction, which is started when data is received by an input node, and can be committed or rolled back when all processing has completed.

“Configuring ODBC connections for globally coordinated transactions” on page 399

Configure the definition of your ODBC databases to the transaction manager (the queue manager).

“Enabling ODBC connections to the databases”

Set up the resources and environment that the broker requires for Open Database Connectivity (ODBC) connections to databases on distributed systems.

“Enabling JDBC connections to the databases” on page 384

Configure connections to a database through a JDBCProvider configurable service.

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

Related information:

 [WebSphere MQ Version 7 product documentation](#)

Enabling ODBC connections to the databases:

Set up the resources and environment that the broker requires for Open Database Connectivity (ODBC) connections to databases on distributed systems.

About this task

You can configure both ODBC and Java Database Connectivity (JDBC) connections for access to databases:

- To set up ODBC connections to databases, follow the instructions in this section. Optionally, after configuring the ODBC connection parameters, run the `mqsicvp` command to verify that the broker can connect to the data source, and to provide useful information about the data source and its interface. On Linux and UNIX systems, this command also checks that the ODBC environment is set up correctly.
- To set up JDBC connections to databases, see “Enabling JDBC connections to the databases” on page 384.
- On z/OS systems, see Data sources on z/OS for information about enabling connections to databases. You do not have to follow the tasks that are described in this section.
- On Linux and UNIX systems, unixODBC is the supported driver manager and is installed by default as part of the IBM Integration ODBC Database Extender SupportPac. The SupportPac is automatically installed in silent mode when the installation of the IBM Integration Bus component completes. For more information, see “Installing the IBM Integration ODBC Database Extender” on page 76. DataDirect database drivers continue to be used by the unixODBC driver manager when interfacing with Oracle, Sybase, and SQLServer databases.
- On Linux and UNIX systems, delete the ODBCINI64 environment variable if it exists; it is not required by IBM Integration Bus Version 9.0. For more information, see Database connections. The sample `odbc.ini` and `odbcinst.ini` files that are supplied, and the information that is contained in these configuration topics, include all the connection parameters that are supported for connections to your databases. Any additional parameters that are provided by your chosen database drivers are not tested or supported in a broker environment; consider your requirements carefully before specifying other parameters in your tailored ODBC `.ini` files.

To enable connections on distributed systems:

Procedure

Define the ODBC DSNs according to your platform:

On Windows:

Follow the instructions in “Connecting to a database from Windows systems” on page 371.

On Linux and UNIX systems:

For all supported databases, follow the instructions in “Connecting to a database from Linux and UNIX systems by using the IBM Integration ODBC Database Extender” on page 375.

You have now configured the ODBC DSNs for your databases.

Results

You have now enabled the broker to make connections to your databases.

What to do next

Next: If you have been using the instructions in Working with databases, the next task is “Configuring ODBC connections for globally coordinated transactions” on page 399 (optional).

Related concepts:



Database connections

Databases contain business data that is written and accessed by deployed message flows. You must create connections from the broker to the database by using ODBC or JDBC.

Related tasks:



Working with databases

Create and configure databases to use with your message flows.

“Enabling JDBC connections to the databases” on page 384

Configure connections to a database through a JDBCProvider configurable service.

“Connecting to a database from Linux and UNIX systems by using the IBM Integration ODBC Database Extender” on page 375

IBM Integration ODBC Database Extender encapsulates the unixODBC driver manager. You must set up and configure the broker to use it.

Related reference:

“Support for 32-bit and 64-bit platforms” on page 625

IBM Integration Bus operates in 32-bit mode or 64-bit mode, on supported operating systems.

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.



Sample IBM Integration Bus ODBC configuration files

A copy of the sample ODBC definition files that are supplied with IBM Integration Bus.

mqsicvp command

Use the **mqsicvp** command to perform verification tests on a broker, or to verify ODBC connections.

Connecting to a database from Windows systems:

To enable a broker to connect to a database, define the ODBC data source name (DSN) for the database.

Before you begin

Before you start: Check that you have set up your environment so that the broker can connect to the database. Most database managers set up the required environment when you install, but others supply a database profile that you must run. For information about environments and running database profiles, see *Setting up a command environment: Windows platforms*.

For 32-bit Data Flow Engines (DFEs), you must use the 32-bit ODBC driver manager to make the 32-bit ODBC definitions. Similarly, for 64-bit DFEs, you must use the 64-bit ODBC driver manager to make the 64-bit ODBC definitions.

About this task

Configure an ODBC data source by using the ODBC Data Source Administrator:

1. Click **Start > Control Panel > Administrative Tools > Data Sources (ODBC)**.

If you are using 32-bit DFEs on Windows 64-bit systems, complete the following steps:

- a. Make a copy of **Data Sources (ODBC)**.
 - b. Right-click the **Data Sources (ODBC)** shortcut and select properties.
 - c. In the 64-bit ODBC dialog, by default the target points to %SystemRoot%\system32\odbcad32.exe. Copy the shortcut, modify the target to point to %SystemRoot%\SysWow64\odbcad32.exe, and use the revised copy of the shortcut instead.
2. Click the **System DSN** tab and click **Add**.
 3. Complete the steps in the following sections for the databases that you are working with.

If you need more information about a particular database product, see the product-specific documentation.

DB2 UDB

Define a data source for DB2 UDB:

1. Select the driver **IBM DB2 ODBC DRIVER**.
2. Enter the data source name (DSN) and description.
3. Select the correct database alias from the list.
4. Click **Finish** to save your definition.
5. Click **OK** to close the ODBC Data Source Administrator.
6. Optional: If you need to use Global Coordination with your database from IBM Integration Bus on Windows systems, the next task is to set up the 32-bit environment that is needed by WebSphere MQ, see “Setting your environment to support 32-bit access to databases” on page 383.

You must register the data source as a system data source.

If you prefer, you can use the Configuration Assistant instead of the ODBC Data Source Administrator:

1. Open the DB2 Configuration Assistant.
2. Right-click the database and select **Change Database**.
3. Select **Data Source**.
4. Select **Register this database for ODBC**. Select the system data source option.
5. Click **Finish**.
6. The Test Connection dialog opens automatically and you can test the various connections.

Informix Dynamic Server

Define a data source for Informix Dynamic Server:

1. Select the driver **IBM INFORMIX ODBC DRIVER**.
2. On the **Connection** tab, specify:
 - The Informix server name.
 - The server host name.
 - The Informix network service name (as defined in the services file).
 - The network protocol (for example, `olsoc tcp`).
 - The Informix data source name.
 - The user identifier to access the data source within.
 - The password for that user identifier.
3. Click **Apply**.
4. Click **Test Connection** to check your supplied values.
5. Click **OK** to close the ODBC Data Source Administrator.

Microsoft SQL Server

Define a data source for Microsoft SQL Server:

1. Select the driver for the version of SQL Server that you are using:
 - **SQL Native Client for SQL Server 2005**.
 - **SQL Native Client 10.0 for SQL Server 2008**.
2. Specify a name and description.
3. Select the correct server from the list.
4. Optional: To specify the authentication mode that is used by the server:
 - a. Click **Next**.
 - b. Select the authentication mode.
 - c. Click **Back** to move back to the first panel.
5. Click **Finish** to save your definition.
6. Click **OK** to close the ODBC Data Source Administrator.

Oracle Define a data source for Oracle:

1.
 - If you are using IBM Integration Bus for Windows 32-bit , select the driver **IBM Integration Bus (9.0.0.0) DataDirect Technologies 7.0 32-BIT Oracle Wire Protocol**.
 - If you are using IBM Integration Bus for Windows 64-bit , select the driver **IBM Integration Bus (9.0.0.0) DataDirect Technologies 7.0 64-BIT Oracle Wire Protocol**.

The ODBC Oracle Driver Setup dialog box opens.

2. On the **General** tab:
 - a. Enter the DSN name, description, and host name of the machine where Oracle is running, the port number on which Oracle is listening, and the Oracle Service Name that you want to connect to.
3. On the **Advanced** tab:
 - a. Select **Enable SQLDescribeParam**.
 - b. Select **Procedure Returns Results**. The resultant ODBC definition in the Windows registry has a string value that is called **ProcedureRetResults** with the value 1.
 - c. Select **Login Timeout** and set the value to 0.
4. Click **OK** to close the ODBC Data Source Administrator.
5. Click **Start > Run**.
6. Type REGEDIT in the **Open** field and click **OK**.
7. In the Registry Editor, navigate to the correct location.
 - If you are using IBM Integration Bus for Windows 32-bit on Windows 32-bit editions: HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI
 - If you are using IBM Integration Bus for Windows 32-bit on Windows 64-bit editions: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI
 - If you are using IBM Integration Bus for Windows 64-bit on Windows 64-bit editions: HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI
8. Expand that location, and right-click your DSN entry. Select **New > String Value**.
9. Specify **WorkArounds** for the string name.
 - a. Right-click **WorkArounds**.
 - b. Select **Modify**.
 - c. Type the data value 536870912.
10. Close the Registry Editor.

Oracle using Secure Socket Layer (SSL) authentication

Complete the steps for Oracle above.

Then, complete these additional steps:

1. Reopen the ODBC Oracle Driver Setup dialog box, see Step 1 for Oracle above.
2. On the **Security** tab:
 - a. In the **Authentication** section, set the **Authentication Method** to **Encrypt Password**.
 - b. In the **Encryption Section**, set the **Encryption Method** to **SSL Auto**.
 - c. Select the check box if you want to **Validate the Server certificate**.
 - d. Enter a fully qualified path for your **Trust Store**.
 - e. Enter your **Trust Store Password**.
 - f. Enter a fully qualified path for your **Key Store**.
 - g. Enter your **Key Store Password**.
 - h. Enter your **SSL Key Password**.
3. Click **OK** to close the ODBC Data Source Administrator.

Sybase Adaptive Server Enterprise

Define a data source for Sybase Adaptive Server Enterprise:

1.
 - If you are using IBM Integration Bus for Windows 32-bit , select the driver IBM Integration Bus (9.0.0.0) DataDirect Technologies 7.0 32-BIT Sybase Wire Protocol.
 - If you are using IBM Integration Bus for Windows 64-bit , select the driver IBM Integration Bus (9.0.0.0) DataDirect Technologies 7.0 64-BIT Sybase Wire Protocol.
2. Enter the DSN name, description, and network address of the server, where the network address is made up of **MyHostMachineName,MyHostMachinePortNumber**.
3. On the **Advanced** tab:
 - Select **Enable Describe Parameter**.
 - Select **Login Timeout** and set the value to 0.
4. On the **Performance** tab:
 - Ensure that the **Prepare Method** setting is 1 - Partial.
5. Click **Start > Run**.
6. Type REGEDIT in the **Open** field and click **OK**.
7. In the Registry Editor, navigate to the correct location:
 - If you are using IBM Integration Bus for Windows 32-bit on Windows 32-bit editions: HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI
 - If you are using IBM Integration Bus for Windows 32-bit on Windows 64-bit editions: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI
 - If you are using IBM Integration Bus for Windows 64-bit on Windows 64-bit editions: HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI
8. Expand that location, and right-click your DSN entry. Select **New > String Value**. Specify **SelectUserName** for the string, and set the value to 1.
9. Right-click your DSN again, and select **New > String Value**. Specify **EnableSPColumnTypes** for the string, and set the value to 2.
10. Right-click your DSN again, and select **New > String Value**. Specify **TimestampTruncationBehavior** for the string, and set the value to 1.
11. Right-click your DSN again, and select **New > String Value**. Specify **XACConnOptBehavior** for the string, and set the value to 3.
12. Close the Registry Editor.

solidDB

Define a data source for solidDB:

1. Select the driver IBM solidDB - (Unicode) DRIVER.
2. Enter the description.
3. Enter the communication port in the network location field, for example, tcp 2315.
4. Click **Finish** to save your definition.
5. Click **OK** to close the ODBC Data Source Administrator.

The solidDB is only supported on Windows 32-bit operating systems.

Results

You have now configured your ODBC data source names on Windows.

Related tasks:

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

Connecting to a database from Linux and UNIX systems by using the IBM Integration ODBC Database Extender:

IBM Integration ODBC Database Extender encapsulates the unixODBC driver manager. You must set up and configure the broker to use it.

Before you begin

The following information applies to all supported databases.

Before you start:

- Read the information about the unixODBC Project.
- Create the database.
- Ensure that the broker is authorized to access the database.
- Check that you have set up your environment so that the broker can access the database. You might have to run a database profile that is supplied by the database vendor. For more information, see *Setting up a command environment: Linux and UNIX systems*.

Procedure

1. Copy the `odbc.ini` sample file that is supplied in the `install_dir/ODBC/unixodbc/` directory to a location of your choice. Each broker service user ID on the system can therefore use its own DSN definitions.

See the sample file contents in *Sample IBM Integration Bus ODBC configuration files*.

Note: To prevent problems with the backup and restore procedures, we recommend that the copy of the sample file is placed into the `/var/mqsi` directory rather than the home directory for your user ID.

2. Ensure that the `odbc.ini` file is owned by the `mqbrkrs` group, and has 664 permissions.
3. Set the `ODBCINI` environment variable to point to your `odbc.ini` file, specifying a full path and file name. Make sure that you point to the copy, do not point to the `odbc.ini` file in the installation directories.
4. Copy the `odbcinst.ini` sample file that is supplied in the `install_dir/ODBC/unixodbc/` directory to a location of your choice. See **Note:** in step 1.

See the sample file contents in *Sample IBM Integration Bus ODBC configuration files*.

5. Ensure that the `odbcinst.ini` file has file ownership of `mqm:mqbrkrs` and has the same permissions as the supplied sample file.
6. Set the `ODBCSYSINI` environment variable to point to the directory that contains the `odbcinst.ini` file, specifying a full path name. Make sure that you point to the directory containing the copy, do not point to the directory containing the `odbcinst.ini` file in the installation directories.
7. If you are connecting to DB2 solidDB, or Informix databases, set the library search path environment variable to show the location of the libraries for the database manager that you are using.

For more information about the library search path, ask your database administrator (DBA), or see the documentation for your database manager.

The library search path environment variable depends on your platform:

- **Linux** **Solaris** On Linux and Solaris, set `LD_LIBRARY_PATH`.
- **HP-UX** On HP-UX, set `SHLIB_PATH`.
- **AIX** On AIX, set `LIBPATH`.

Updates to the library search path are not required for other supported databases.

8. If you are using a DB2 database instance that is installed on AIX, a single process can make a maximum of 10 connections that use shared memory to a DB2 database. Use TCP/IP mode to connect to the database instance; see DB2 error message `SQL1224N` is issued when you connect to DB2.
9. Edit the final stanza in the `odbc.ini` file, the `[ODBC]` stanza, to specify the location of the installed DataDirect ODBC Drivers.

To ensure that you edit the correct `odbc.ini` file, you can open the file in the `vi` text editor by using the following command:

```
vi $ODBCINI
```

- a. In `InstallDir`, add the IBM Integration Bus installation location to complete the fully qualified path to the ODBC directory. If you do not specify this value correctly, the ODBC definition does not work.
- b. Accept the default values shown in the sample `odbc.ini` file for all the other entries in the stanza.

For example, on AIX:

```
#####
##### Mandatory information stanza #####
#####
```

```
[ODBC]
InstallDir=/usr/opt/IBM/mqsi/9.0.0.0/ODBC/V7.0
UseCursorLib=0
IANAAppCodePage=4
UNICODE=UTF-8
```

10. Edit the first stanza in the `odbc.ini` file, the `[ODBC Data Sources]` stanza, to list the DSN of each database.

For example, on AIX:

```
#####
##### List of data sources stanza #####
#####
[ODBC Data Sources]
DB2DB=IBM DB2 ODBC Driver
ORACLEDB=DataDirect 7.0 ODBC Oracle Wire Protocol
ORACLERACDB=DataDirect 7.0 ODBC Oracle RAC Wire Protocol
ORACLESSLDB=DataDirect 7.0 ODBC Oracle SSL Wire Protocol
```



```
SYBASEDB=DataDirect 7.0 ODBC Sybase Wire Protocol
SYBASEDBUTF8=DataDirect 7.0 ODBC Sybase UTF8 Wire Protocol
SQLSERVERDB=DataDirect 7.0 ODBC SQL Server Wire Protocol
INFORMIXDB=IBM Informix ODBC Driver
SOLIDDB_DB=IBM Solid DB ODBC Driver
```

List all your DSNs in your `odbc.ini` file, regardless of the database manager. You can define multiple DSNs to resolve to the same database; however, if you are using global coordination of transactions with an Oracle database, do not use this option because it might cause data integrity problems.

11. For each database that you listed in the [ODBC Data Sources] stanza, within the `odbc.ini` file, create a data source stanza in the `odbc.ini` file. The entries in the stanza depend on the database manager. Slight differences also occur between operating systems, for example the file paths to the drivers.

For a DB2 database instance:

For Linux on x86:

- a. In **Driver**, add the full path of your DB2 installation.
- b. In **Description**, type a meaningful description of the database. This field is for information only and does not affect the connection.
- c. In **Database**, type the DB2 alias. The data source name must be the same as the database alias name. If you are using a remote DB2 database, you must set up your client/server connection to resolve this alias to the correct database.

If the requirement is to have multiple stanzas that refer to the same DB2 database, aliases must be created in DB2 by using the DB2 CATALOG command. These aliases can then have their own stanza in the ODBCINI file.

The ODBCINI file cannot be used to set up aliases for DB2.

For more information, see the DB2 documentation.

For example:

```
# DB2 stanza
[MYDB2DB]
Driver=/opt/IBM/db2/V9.7/lib32/libdb2.so
Description=IBM DB2 ODBC Database
Database=MYDB2DB
```

For all other platforms:

- a. In **Driver**, add the full path of your DB2 installation.
- b. In **Description**, type a meaningful description of the database. This field is for information only and does not affect the connection.
- c. In **Database**, type the DB2 alias. The data source name must be the same as the database alias name. If you are using a remote DB2 database, you must set up your client/server connection to resolve this alias to the correct database. For more information, see the DB2 documentation.

If the requirement is to have multiple stanzas that refer to the same DB2 database, aliases must be created in DB2 by using the DB2 CATALOG command. These aliases can then have their own stanza in the ODBCINI file.

The ODBCINI file cannot be used to set up aliases for DB2.

For example, on AIX:

```

;# DB2 stanza
[MYDB2DB]
DRIVER=/opt/IBM/db2/V9.7/lib64/db2o.o
Description=IBM DB2 ODBC Database
Database=MYDB2DB

```

For an Oracle database:

For all platforms:

- a. In **Driver**, add the IBM Integration Bus installation location to complete the fully qualified path to the driver shown in the sample `odbc.ini` file.
- b. In **Description**, type a meaningful description of the database. This field is for information only and does not affect the connection.
- c. In **HostName**, type the name or IP address of the machine that is hosting your Oracle system.
- d. In **PortNumber**, type the number of the port on which your Oracle server is listening on the machine you specified in **HostName**.
- e. In **ServiceName**, type the Oracle service name that you want to connect to on the system you specified in **HostName**.
- f. Accept the default values shown in the sample `odbc.ini` file for all the other entries in the stanza.

For example, on AIX:

```

;# Oracle stanza
[MYORACLEDB]
Driver=/usr/opt/IBM/mqsi/9.0.0.0/ODBC/V7.0/lib/UKora26.so
Description=DataDirect ODBC 7.0 Oracle Wire Protocol
HostName=my-machine.hursley.ibm.com
PortNumber=1521
ServiceName=my-oracle-service
CatalogOptions=0
EnableStaticCursorsForLongData=0
ApplicationUsingThreads=1
EnableDescribeParam=1
OptimizePrepare=1
WorkArounds=536870912
ProcedureRetResults=1
ColumnSizeAsCharacter=1
LoginTimeout=0

```

For an Oracle database that uses Real Application Clusters:

For all platforms:

- a. In **Driver**, add the IBM Integration Bus installation location to complete the fully qualified path to the driver shown in the sample `odbc.ini` file.
- b. In **Description**, type a meaningful description of the database. This field is for information only and does not affect the connection.
- c. In **HostName**, type the name or IP address of the machine that is hosting your primary (preferred) Oracle instance.
- d. In **PortNumber**, type the number of the port on which your Oracle server is listening on the machine you specified in **HostName**.
- e. In **ServiceName**, type the Oracle Real Application Cluster service name that you want to connect to on the system you specified in **HostName**.
- f. In **AlternateServers**, provide a list of alternative locations for this service for situations when the primary location, which is defined

in **HostName**, is unavailable. Each location specification consists of three parts, which are separated by colons. Enter these values as one continuous string; the text in this example has been split to improve readability.

```
HostName=<Alternative host name>
:PortNumber=<Oracle listener port on alternative server>
:ServiceName=<Service name on the alternative server>
```

If you want to specify more than one AlternateServer, separate each additional location specification with a comma. Whenever a new database connection is required, for example after an Oracle instance failover, the primary location will be tried first. However, if the primary location is unavailable, the driver will try the list of alternative locations in turn.

- g. Accept the default values shown in the sample `odbc.ini` file for all the other entries in the stanza.

For example, on AIX:

```

# Oracle Real Application Clusters stanza
[MYORACLESLDB]
Driver=/usr/opt/IBM/mqsi/9.0.0/ODBC/V7.0/lib/UKora26.so
Description=DataDirect 7.0 ODBC Oracle Wire Protocol
HostName=my-machine.hursley.ibm.com
PortNumber=2484
ServiceName=my-oracle-rac-service
#This stanza uses alternate server definition. Add extra ones using ; to separate each definition.
AlternateServers=HostName=first-backup-machine.hursley.ibm.com:PortNumber=1521;ServiceName=my-oracle-rac-first-backup-service;HostName=second-backup-machine.hursley.ibm.com:PortNumber=1521;ServiceName=my-oracle-rac-second-backup-service
CatalogOptions=0
EnableStaticCursorsForLongData=0
SQLCursorTimeout=1
EnableUnicodeSupport=1
UnicodeSupport=1
Workaround=53821002
Precedence=Default=1
ColumnDisplayCharacter=1
sqlunicode=0

```

For an Oracle database that uses Secure Socket Layer (SSL):

For all platforms:

- a. In **Driver**, add the IBM Integration Bus installation location to complete the fully qualified path to the driver shown in the sample `odbc.ini` file.
- b. In **Description**, type a meaningful description of the database. This field is for information only and does not affect the connection.
- c. In **HostName**, type the name or IP address of the machine that is hosting your primary (preferred) Oracle instance.
- d. In **PortNumber**, type the number of the port on which your Oracle server is listening for SSL connections on the machine you specified in **HostName**.
- e. In **ServiceName**, type the Oracle SSL service name that you want to connect to on the system you specified in **HostName**.
- f. In **KeyPassword**, type your SSL key password.
- g. In **KeyStore**, type the fully qualified name of your SSL key store.
- h. In **KeyStorePassword**, type your SSL key store password.
- i. In **TrustStore**, type the fully qualified name of your SSL trust store.
- j. In **TrustStorePassword**, type your SSL trust store password.
- k. Accept the default values shown in the sample `odbc.ini` file for all the other entries in the stanza.

For example, on AIX:

```

;# Oracle using SSL stanza
[MYORACLESSLDB]
Driver=/usr/opt/IBM/mqsi/9.0.0/ODBC/V7.0/lib/UKora26.so
Description=DataDirect 7.0 ODBC Oracle Wire Protocol
HostName=my-machine.hursley.ibm.com
PortNumber=2484
ServiceName=my-oracle-ssl-service
CatalogOptions=0
EnableStaticCursorsForLongData=0

```

```

ApplicationUsingThreads=1
EnableDescribeParam=1
OptimizePrepare=1
WorkArrounds=536870912
ProcedureRetResults=1
ColumnSizeAsCharacter=1
LoginTimeout=0
AuthenticationMethod=1
KeyPassword=my-password
KeyStore=/Development/ssl/my-store.p12
KeyStorePassword=my-password
TrustStore=/Development/ssl/my-store.p12
TrustStorePassword=my-password
ValidateServerCertificate=1

```

For a Sybase database:

For all platforms except Linux on System z:

- a. In **Driver**, add the IBM Integration Bus installation location to complete the fully qualified path to the driver shown in the sample `odbc.ini` file.
- b. In **Description**, type a meaningful description of the database. This field is for information only and does not affect the connection.
- c. In **Database**, type the name of the database to which you want to connect by default. If you do not specify a value, the default value is the database that is defined by your system administrator for each user.
- d. In **NetworkAddress**, type the network address of your Sybase ASE server (this address is required for local and remote databases). Specify an IP address or server name as follows:
 <Your Sybase server name or IP address>,<Your Sybase port number>
 For example: Sybaseserver,5000. You can also specify the IP address directly, for example 199.226.224.34,5000. You can find the port number in the Sybase interfaces file that is named `interfaces`.
- e. Accept the default values shown in the sample `odbc.ini` file for all the other entries in the stanza.

For example, on AIX:

```

;# Sybase Stanza
[MYSYBASEDB]
Driver=/usr/opt/IBM/mqsi/9.0.0.0/ODBC/V7.0/lib/UKase26.so
Description=DataDirect 7.0 ODBC Sybase Wire Protocol
Database=SYBASEDB1
ApplicationUsingThreads=1
EnableDescribeParam=1
OptimizePrepare=1
SelectMethod=0
NetworkAddress=my-machine.hursley.ibm.com:4100
SelectUserName=1
ColumnSizeAsCharacter=1
EnableSPColumnTypes=2
LoginTimeout=0
TimestampTruncationBehavior=1
XAConnOptBehavior=3

```

If you want to use a UNICODE UTF8 Sybase data source, add the following line to the end of your Sybase stanza:

Charset=UTF8

For remote access to an SQL Server database

For all platforms except Linux on POWER:

- a. In **Driver**, add the IBM Integration Bus installation location to complete the fully qualified path to the driver shown in the sample `odbc.ini`.
- b. In **Description**, type a meaningful description of the database. This field is for information only and does not affect the connection.
- c. In **Address**, either:
 - 1) Specify an IP address or server name and a port number to locate the SQLServer database you want to connect to, as follows:
<Your SQLServer machine name or IP address>,<Your SQLServer port number>

or

 - 2) Specify an IP address or server name and an instance name to locate the SQLServer database you want to connect to, as follows:
<Your SQLServer machine name or IP address>\<Your SQLServer instance name>

If your instance name is blank, specify <Your SQLServer machine name or IP address>
- d. In **Database**, type the name of the database to which you want to connect by default. If you do not specify a value, the default value is the database that is defined by your system administrator for each user.
- e. Accept the default values shown in the sample `odbc.ini` file for all the other entries in the stanza.

For example, on AIX:

```
;&#x21; UNIX to SQLServer stanza
[MYSQLSERVERDB]
Driver=/usr/opt/IBM/mqsi/9.0.0.0/ODBC/V7.0/lib/UKmsss26.so
Description=DataDirect 7.0 ODBC SQL Server Wire Protocol
Address=my-machine.hursley.ibm.com:1433
AnsiNPW=Yes
Database=MYSQLSERVERDB
QuotedId=No
ColumnSizeAsCharacter=1
LoginTimeout=0
```

For an Informix database

For Linux on x86

- a. In **Driver**, add the full path of your Informix Client library.
- b. In **Description**, type a meaningful description of the database. This field is for information only and does not affect the connection.
- c. In **ServerName**, type the name of the Informix IDS server.
- d. In **Database**, type the name of the database to which you want to connect by default. If you do not specify a value, the default value is the database that is defined by your system administrator for each user.

For example:

```

;# Informix Stanza
[MYINFORMIXDB]
Driver=/Informix/client/lib/cli/iclit09b.so
Description=IBM Informix ODBC Database
ServerName=my-machine
Database=MYDB

```

For all other platforms:

- a. In **Driver**, add the full path of your Informix Client library.
- b. In **Description**, type a meaningful description of the database. This field is for information only and does not affect the connection.
- c. In **ServerName**, type the name of the Informix IDS server.
- d. In **Database**, type the name of the database to which you want to connect by default. If you do not specify a value, the default value is the database that is defined by your system administrator for each user.

For example, on AIX:

```

;# Informix Stanza
[MYINFORMIXDB]
Driver=libinfWrapper.so
Description=IBM Informix ODBC Database
ServerName=my-machine
Database=MYDB

```

For a solidDB database:

For all platforms except Linux on POWER and Linux on System z:

Client side

odbc.ini file

- a. In **Driver**, add the full path of your solidDB Client library.
- b. In **Description**, type a meaningful description of the database. This field is for information only and does not affect the connection.
- c. In **Database**, type the name of the database to which you want to connect by default. If you do not specify a value, the default value is the database that is defined by your system administrator for each user.

For example, on AIX:

```

;# SolidDB Stanza
[SOLID_DB]
Driver=/opt/solidDB/bin/soca5x6465.so
Description=IBM Solid DB ODBC database
Database=SOLIDDB_DB

```

Note: all additional information is ignored.

solid.ini file

- a. This configuration file is located in the directory that is referenced by the environment variable SOLIDDIR.
- b. The solid.ini mapping is from the data source name (as defined in ODBCINI) to the solidDB connection string.
- c. The connection string takes the form <logical name of the driver> = <physical solidDB connect string>.
- d. The Physical connection string specifies the:
 - Protocol
 - Machine name or IP address

- Port number to use

For example, on AIX:

```
[Data Sources]
SOLIDDB_DB=tcp my_aix_system 1964
```

Server side

solid.ini file

- This configuration file is located in the installation directory for solidDB.
- Set the Data Source as for the Client side `solid.ini` file.
- Set **Listen** to where the listener for the server is located.
- Set **CharPadding=yes** and **NumericPadding=yes** to turn on padding.

For example, on AIX:

```
[Data Sources]
SOLIDDB_DB=tcp my_aix_system 1964
```

```
[COM]
Listen=tcPIP 1964
```

```
[SQL]
CharPadding=yes
NumericPadding=yes
```

- Ensure that you have edited all necessary parts of all of the relevant `.ini` files:
 - The [ODBC Data Source] stanza at the top of the `odbc.ini` file.
 - A stanza for each data source in the `odbc.ini` file.
 - The [ODBC] stanza at the end of the `odbcinst.ini` file.
 - Additionally, for solidDB, both the client and server-side `solid.ini` files.

If you do not configure all parts correctly, the ODBC DSNs do not work and the broker is unable to connect to the database.

Results

You have now configured database connections on Linux and UNIX. You can check that the ODBC environment is configured correctly by running the `mqsicvp` command. For more information, see `mqsicvp` command.

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.



Sample IBM Integration Bus ODBC configuration files

A copy of the sample ODBC definition files that are supplied with IBM Integration Bus.

Related information:

WebSphere MQ Version 7 product documentation

Setting your environment to support 32-bit access to databases:

When you have configured your ODBC data source names (DSNs), you must also configure the environment so that you can issue console commands, and the

brokers that you start can access the required database libraries. For example, if you have a DB2 database, you must add the DB2 client libraries to your library search path.

About this task Windows

1.

On Windows platforms, the environment is typically set up for you when you install the database product, and no further action is required. However, some database managers provide a database profile that you must run to enable the connection from the broker; for further information, see *Setting up a command environment: Windows platforms*.

2. Optional: If you intend to use Global Coordination (XA) with your database then the next task is “Configuring ODBC connections for globally coordinated transactions” on page 399.

Linux

UNIX

1.

On Linux and UNIX systems, run a profile for each database you want to access. For example, on DB2 you must run `db2profile`; other database vendors have similar profiles. For further information, see *Setting up a command environment: Linux and UNIX systems*.

2. Optional: If you have been following the instructions in *Working with databases* and you intend to use Global Coordination (XA) with your database, then the next task is “Configuring ODBC connections for globally coordinated transactions” on page 399.

Related tasks:

“Connecting to a database from Linux and UNIX systems by using the IBM Integration ODBC Database Extender” on page 375

IBM Integration ODBC Database Extender encapsulates the unixODBC driver manager. You must set up and configure the broker to use it.

Related reference:

“Support for 32-bit and 64-bit platforms” on page 625

IBM Integration Bus operates in 32-bit mode or 64-bit mode, on supported operating systems.

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.



Sample IBM Integration Bus ODBC configuration files

A copy of the sample ODBC definition files that are supplied with IBM Integration Bus.

Enabling JDBC connections to the databases:

Configure connections to a database through a JDBCProvider configurable service.

About this task

Use a JDBC connection from Java programs that are associated with a JavaCompute node or a user-defined node that is written in Java.

You must also set up JDBC connections if your message flows include graphical data maps with one or more database transforms to be run from a Mapping node, or if they include DatabaseRetrieve or DatabaseRoute nodes.

If you configure a JDBC type 4 connection from an application running on a Linux, UNIX, or Windows system, you can configure your broker and queue manager to include interactions with the databases in globally-coordinated transactions. On z/OS, JDBC connections can be broker-coordinated only.

The information provided in this section is independent of whether your operating systems, brokers, integration servers, queue managers, and databases operate in 32-bit or 64-bit mode, except where stated.

When you write Java classes for a JavaCompute node or a user-defined node, your code must comply with the following restrictions:

- Do not include any code that performs a COMMIT or a ROLLBACK function.
- Do not close the connection to the database. The broker manages all connections, and closes a connection if it is idle for approximately one minute, or if the message flow completes.

To configure JDBC type 4 connections:

Procedure

1. Set up your JDBC provider definition.
2. Optional: Set up security.
3. Optional: Configure for global-coordination of transactions.
4. Optional: If your broker is running on a Windows system, authorize access to JDBCProvider resources.

What to do next

Next: If you have been following the instructions in Working with databases or Mapping database content, the next task is “Setting up a JDBC provider for type 4 connections” on page 386.


When you have completed configuration of the databases, add or modify Java code in your JavaCompute or user-defined nodes to access the database that is identified in the JDBCProvider configurable service.


Related concepts:

 Databases overview

Brokers can manipulate business data that is stored in databases. If you have databases, you must configure them before you can access them from your message flow.

Related tasks:

 Interacting with databases by using the JavaCompute node
Access databases from Java code included in the JavaCompute node.

 Extending the capability of a Java message processing or output node
Within a message processing or output node, you can add extended functions to your Java node.

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

Setting up a JDBC provider for type 4 connections:

Use the **mqscreateconfigurable** service or the **mqschange** properties command to configure a JDBC provider service.

Before you begin

Before you start:

- Create a broker
- Create the database by following your database documentation.

About this task

When you include a DatabaseRetrieve, DatabaseRoute, JavaCompute, Mapping, or Java user-defined node in a message flow, and interact with a database in that node, the broker must establish a connection with the database to fulfill the operations that are performed by the node. You must define a JDBCProvider configurable service to provide the broker with the information that it needs to complete the connection.

Important: When naming your JDBCProvider service, consider the following requirements:

- If you want to use your JDBCProvider service with a JavaCompute node, or with a Java user-defined node, the name of your JDBCProvider service must match the *datasourceName* parameter in the `getJDBCType4Connection()` API call from the node.
- If you want to use your JDBCProvider service with a Mapping node, the name of your JDBCProvider service must match the database name that is used by the database transforms in your Graphical Data Map. For each database transform, the database name is determined by the database definition (.dbm file) in the Data Design project that was used to create the map.
- If you want to use your JDBCProvider service with a DatabaseRetrieve node, or with a DatabaseRoute node, the name of your JDBCProvider service must match the value of the *Data source name* property of the node.

A JDBCProvider configurable service supports connections to one database only; you must create a service for each database that your nodes or Java applications connect to.

To set up a JDBC provider for type 4 connections by using the IBM Integration Explorer, see “Creating a new configurable service” on page 347.

To set up a JDBC provider for type 4 connections by using the **mqscreateconfigurable** service or **mqschange** properties commands, complete the following steps:

Procedure

1. Identify the type of database for which you require a JDBCProvider service.

Supported JDBC drivers and databases are shown in “Supported databases” on page 628; support for globally coordinated (XA) transactions is restricted on some platforms and for some databases.

2. Run the **mqsireportproperties** command to view the list of available JDBCProvider services. Substitute the name of your broker in place of *broker_name*.

```
mqsireportproperties broker_name -c JDBCProviders -a -o AllReportableEntityNames
```

The command response lists all the JDBCProvider configurable services that are defined. If you have not created your own definitions, the following list of default supplied services is shown:

- DB2
- Informix
- Informix_With_Date_Format
- Microsoft_SQL_Server
- Oracle
- Sybase_JConnect6_05

If you are connecting to an Informix database:

- Use Informix_With_Date_Format for compatibility with client applications that are dependent on the date format connection attribute that was used by earlier versions of Informix servers.
 - Use Informix for client applications that are not dependent on the date format attribute.
3. View the contents of the relevant JDBCProvider service definition. For example, run the following command to display the supplied Oracle definition:

```
mqsireportproperties broker_name -c JDBCProviders -o Oracle -r
```

The command response lists all the properties for the Oracle definition. If you have not changed this definition, the properties are set to initial values, some of which you must change to create a viable definition. For example, the property **databaseName** is set to `default_Database_Name`, and you must change it to identify the specific database that you want to connect to.


A JDBCProvider service has the following properties:

- **connectionUrlFormat**. A pattern that represents the connection URL definition, which is specific to a particular database type. For example, the pattern for DB2 is defined with the following content:

```
jdbc:db2://[serverName]:[portNumber]/[databaseName]:user=[user];password=[password];
```

The pattern is used and completed by the broker at run time when it connects to the database. The values in brackets, for example `[serverName]`, are substituted by the broker into the pattern by using the values that you have specified on the **mqsicreateconfigurableservice**, **mqsichangeproperties**, or **mqsisetdbparms** commands.

The following values and order of preference are used by the broker to substitute the user ID and password in the pattern:

- a. First, on all platforms: The user ID and password that you have set for the specific database, by using the **mqsisetdbparms** and specifying the database in the **-n** parameter.
- b. Second, on all platforms: The user ID and password that you have set for all other databases, by using the **mqsisetdbparms** and specifying `jdbc::JDBC` in the **-n** parameter.
- c. Third, the values are platform-specific:
 - 1)  On Windows: The broker service ID and password that you specified on the **mqsicreatebroker** command.

- 2) **Linux** **UNIX** On Linux and UNIX: The user ID `mqsUser` and password `*****` (these values are fixed).
- 3) **z/OS** On z/OS: The user ID "" and password "".

If you are using one of the supplied JDBCProvider services, do not use the **mqschangeproperties** command to change the pattern itself; changes made to the pattern might cause unpredictable results.

If you use the **mqscreateconfigurable-service** command to define your own JDBCProvider service, use the **mqsreportproperties** command to check that the content of the **connectionUrlFormat** string exactly matches the default supplied provider for the database type that you are using.

In addition, if you are working on z/OS, and are using the JCL files BIPCRCS and BIPCHPR to define your JDBCProvider service, ensure that your 3270 emulator is configured to use the same code page that the broker is running in. If the code pages do not match, the **connectionUrlFormat** string pattern that you define might not be recognized correctly by the broker.

- **connectionUrlFormat Attr1-5.** If the defined URL pattern for a database contains non-standard JDBC data source properties, such as a server identifier, specify these properties in addition to the standard attributes by using one of five general-purpose connection URL attributes. For example:
 - If **connectionURLFormat** = `jdbc:oracle:thin:[user]/[password]@[serverName]:[portNumber]:[connectionUrlFormatAttr1]`, **connectionUrlFormatAttr1** must contain an Oracle server identifier, which you must supply by defining the value for the property **connectionUrlFormatAttr1** on the **mqscreateconfigurable-service** or **mqschangeproperties** command. The broker can then substitute all the required values into the required pattern.
 - If **connectionURLFormat** = `jdbc:informix-sqli://[serverName]:[portNumber]/[databaseName]:informixserver=[connectionUrlFormatAttr1];user=[user];password=[password]`, **connectionUrlFormatAttr1** must contain the name of the Informix instance on the server (typically specified by the INFORMIXSERVER environment variable). This value is case-sensitive.
- **databaseName.** The name of the database to which the data source entry enables connections; for example, `employees`.
- **databaseSchemaNames.** Optionally override the name of the database schema used in SQL statements created by message flow nodes. This property is used only by the Mapping node, and only when calling a graphical data map that contains a database transform. For more information, see JDBCProviders configurable service.
- **databaseType.** The database type; for example, `DB2`.
- **databaseVersion.** The database version; for example, `9.1`.
- **description.** An optional property to describe the data source definition.
- **environmentParms.** For DB2 and Informix only. An optional property specifying a list of data source properties of the form `name=value` each separated by a semicolon.
- **jarsURL.** The local directory path, on the system on which the broker is running, where the JAR file that contains the type 4 driver class is located. In addition, a storage area network disk can be used for the directory path, but a mapped network drive to a remote computer cannot be used.

- **maxConnectionPoolSize.** Optionally set this property to create a JDBC connection pool. For more information, see Using a JDBC connection pool to manage database resources used by an integration server.
- **portNumber.** The port number on which the database server is listening; for example, 50000.
- **securityIdentity.** A unique security key to perform a second broker registry lookup to find an entry under the broker security identities, which store the encrypted password for the user on the associated host system; for example, mysecurityIdentity.

Create a security identity by using the **mqsisetdbparms** command, as described in “Securing database connections” on page 362. The value of **securityIdentity** (for example, mysecurityIdentity) must match the value that you specify following the prefix **jdbc::** for the parameter **-n** on that command.

The security identity provides a user ID and password value pair, which are used to access the specified data source defined for a particular JDBCProvider service entry. This property is ignored if the connection URL does not contain both a user ID and password pair, which require property values to be substituted for such inserts.

The default values, which you can set by specifying a **ResourceName** of **jdbc::JDBC** on the **mqsisetdbparms** command, are used under the following conditions:

- If the **securityIdentity** is blank, or if you have not changed it from the default value **default_User@default_Server**, but the identity is required for the connection URL pattern.
- If you have entered a valid unique security identity key, but it cannot be found under the DSN key.

- **serverName.** The name of the server; for example, host1.
 - **type4DatasourceClassName.** The name of the JDBC data source class name that is used to establish a type 4 connection to a remote database, and to coordinate transaction support. For example, specify **com.ibm.db2.jcc.DB2XADataSource** for DB2, or specify **oracle.jdbc.xa.client.OracleXADataSource** for Oracle. You must specify the XA class name when using the **getJDBCType4Connection()** API call for coordinated transactions. If the database server does not support XA transactions, or you do not want to use the XA protocol, this property is optional and you must set the **jdbcProviderXASupport** property to false.
 - **type4DriverClassName.** The name of the JDBC type 4 driver class name that is used to establish a connection. For example, specify **com.ibm.db2.jcc.DB2Driver** for DB2, or specify **oracle.jdbc.OracleDriver** for Oracle.
 - **jdbcProviderXASupport.** An optional property that controls whether the broker connects to a database server using the XA Protocol. By default this property is set to true. If the database server is not enabled for XA Support, or coordinated transactions are not required, set the value to false. In which case the type 4 driver specified using the **type4DriverClassName** property is used, instead of the type 4 datasource specified in the **type4DatasourceClassName** property.
4. If you want to use the provided definition, run the **mqsichangeproperties** command to replace default values with the values specific to your database and environment. If you are in any doubt about the required values, consult your database administrator, or check the documentation that is provided with your chosen database. Some values depend on how and where you have

installed the database product; for example, the property **jarsURL** identifies the location of the JAR files supplied and installed by the database provider.

5. If you want to create a new configurable service, perhaps because you want to retain the supplied service as a template for future definitions, run the **mqsicreateconfigurable-service** command to create the definition.

```
mqsicreateconfigurable-service broker_name -c JDBCProviders -o provider_name  
-n list of properties -v list of values
```

Enter the command on a single line; the example is split to enhance readability.

Specify all the properties that are required by the database provider that you have chosen. To specify a list of properties and values, separate the items after each flag with a comma. For example, `-n databaseName,databaseType -v EmployeeDB,DB2`. If you do not specify all the properties on the **mqsicreateconfigurable-service** command, you can update them later with the **mqsichangeproperties** command.

6. When you have set up or modified your JDBCProvider service, you must reload any integration servers which currently use, or intend to use, the JDBCProvider service.

What to do next

Next: If required, set up security for the JDBC connection, set up the environment to include the JDBCProvider service in globally coordinated transactions, or both.

Related concepts:



Databases overview

Brokers can manipulate business data that is stored in databases. If you have databases, you must configure them before you can access them from your message flow.

Related tasks:

“Securing database connections” on page 362

Set up security for a database connection, whether it is required by the database provider or optional.



Using a JDBC connection pool to manage database resources used by an integration server

Use broker JDBC provider resources to configure the use of thread pools independently of message flow and input node thread pools.

“Configuring a JDBC type 4 connection for globally coordinated transactions” on page 391

If you want the database that you access through a JDBC type 4 connection to participate in globally coordinated transactions, set up the appropriate environment.

“Authorizing access to JDBC type 4 JDBCProvider resources on Windows” on page 394

Authorize the broker and queue manager to access shared resources that are associated with the JDBCProvider. This task is required only if you want the database updates to be included in globally coordinated transactions on Windows systems.



Interacting with databases by using the JavaCompute node

Access databases from Java code included in the JavaCompute node.



Extending the capability of a Java message processing or output node

Within a message processing or output node, you can add extended functions to

your Java node.

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

 **mqsichangeproperties** command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.

 **mqsicreateconfigurable-service** command

Use the **mqsicreateconfigurable-service** command to create an object name for a broker external resource.

 **mqsireportproperties** command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.

 **mqsisetdbparms** command

Use the **mqsisetdbparms** command to associate a specific user ID and password (or SSH identity file) with one or more resources that are accessed by the broker.

 JDBCProviders configurable service

Select the objects and properties that you want to change for the JDBCProviders configurable service.

Configuring a JDBC type 4 connection for globally coordinated transactions:

If you want the database that you access through a JDBC type 4 connection to participate in globally coordinated transactions, set up the appropriate environment.

Before you begin

Before you start: Set up your JDBC provider definition.

About this task

Updates that you make to a database across a JDBC type 4 connection can be coordinated with other actions taken within the message flow, if you set up the resources to support coordination.

Complete the following steps:

Procedure

1. Check that the definition of your JDBCProvider service is appropriate for coordinated transactions.

For example, to set up the required JDBC classes:

- For DB2, set **type4DataSourceClassName** to `com.ibm.db2.jcc.DB2XADataSource` and **type4DriverClassName** to `com.ibm.db2.jcc.DB2Driver`
- For Oracle, set **type4DataSourceClassName** to `oracle.jdbc.xa.client.OracleXADataSource` and **type4DriverClassName** to `oracle.jdbc.OracleDriver`

Consult your database administrator or the documentation provided by your database supplier, to confirm that all the JDBCProvider service properties are set appropriately. For example, a database supplier might require secure access if it is participating in coordinated transactions.

2. Define the switch file and the database properties:

- a. **Linux** **UNIX** On Linux and UNIX systems, open the `qm.ini` file for the broker queue manager with a text editor. Add the following stanza for each database:

```
XAResourceManager:  
  Name=Database_Name  
  SwitchFile=JDBC_Switch  
  XAOpenString=JDBC_DataSource  
  ThreadOfControl=THREAD
```

`Database_Name` is the database name (DSN) of the database defined to the JDBCProvider configurable service (for example, specified by `-n databaseName -v Database_Name` on the `mqsichangeproperties` command).

`JDBC_Switch` is a fixed generic name that represents the switch file for XA coordination. Use this value, or another single fixed value, in each stanza; the specific switch file that the queue manager uses is defined by the symbolic links you create in the next step.

`JDBC_DataSource` is the identifier of the JDBCProvider configurable service (the value that you specified for the `-o` parameter on the `mqsichangeproperties` command).

Define a stanza for each database (DSN) that you connect to from this broker. You must create separate definitions even if the DSNs resolve to the same physical database. Therefore, you must have a stanza for each JDBCProvider configurable service that you have defined, because each service can define the properties for a single database.

- b. **Windows** On Windows on x86 systems, open WebSphere MQ Explorer and select the queue manager for your broker, for example `BROKERQM`.

Open the **XA resource manager** page, and modify the attributes to create the definition of the database. The attributes are the same as those shown for Linux and UNIX; **Name**, **SwitchFile**, **XAOpenString**, and **ThreadofControl**. Leave the additional attribute, **XACloseString**, blank.

Enter the fully qualified file name in **SwitchFile**; `install_dir\bin\JDBC_Switch.dll`.

- c. **Windows** On Windows on x86-64 systems, open WebSphere MQ Explorer and select the queue manager for your broker, for example `BROKERQM`.

Open the **XA resource manager** page, and modify the attributes to create the definition of the database. The attributes are the same as those shown for Linux and UNIX; **Name**, **SwitchFile**, **XAOpenString**, and **ThreadofControl**. Leave the additional attribute, **XACloseString**, blank.

Enter `JDBC_Switch` in **SwitchFile**.

3. Set up queue manager access to the switch file:

- a. **Linux** **UNIX** On Linux and UNIX systems, create a symbolic link to the switch files that are supplied in your `install_dir/lib` directory.

`install_dir` is the directory to which you installed the Integration Bus component. The default location for this directory is `/opt/ibm/mqsi/v.r.m.f` on Linux, or `/opt/IBM/mqsi/v.r.m.f` on UNIX systems. The default directory includes the version, release, modification, and fix of the product, in the format `v.r.m.f` (version.release.modification.fix).

Set up links in the `/var/mqm/exits` directory, or the `/var/mqm/exits64` directory, or both. The file names for each platform are shown in the following table.

Platform	32-bit file	64-bit file
AIX		libJDBCSwitch.so
HP-Itanium		libJDBCSwitch.so
Linux on POWER		libJDBCSwitch.so
Linux on System z		libJDBCSwitch.so
Linux on x86	libJDBCSwitch.so	
Linux on x86-64		libJDBCSwitch.so
Solaris on SPARC		libJDBCSwitch.so
Solaris on x86-64		libJDBCSwitch.so

Specify the same name of the switch file, `JDBCSwitch` or your own value, in both the `/exits` and `/exits64` directories. For example, on AIX:

```
ln -s install_dir/lib/libJDBCSwitch.so /var/mqm/exits/JDBCSwitch
```

and

```
ln -s install_dir/lib/libJDBCSwitch.so /var/mqm/exits64/JDBCSwitch
```

- b. **Windows** On Windows, for the 32-bit version of IBM Integration Bus, copy the `JDBCSwitch.dll` file from the `install_dir\bin` directory to the `\exits` subdirectory in the WebSphere MQ installation directory.
 - c. **Windows** On Windows, for the 64-bit version of IBM Integration Bus, copy the `JDBCSwitch32.dll` file from the `install_dir\bin` directory to the `\exits` subdirectory in the WebSphere MQ installation directory, and rename the file to `JDBCSwitch.dll`. Then, copy the `JDBCSwitch.dll` file from the `install_dir\bin` directory to the `\exits64` subdirectory in the WebSphere MQ installation directory.
4. Configure the message flow that includes one or more nodes that access databases that are to participate in a globally coordinated transaction.
 - a. Open an IBM Integration Toolkit session.
 - b. Switch to the Integration Development perspective.
 - c. Add the message flow that includes the node or nodes that connect to the database that is to participate in a globally coordinated transaction to a new or existing BAR file.
 - d. Build the BAR file.
 - e. Click the **Configure** tab, select the message flow that you have added, and select the **Coordinated Transaction** check box.

What to do next

Next: If your broker is running on Windows, authorize the broker and its queue manager to access resources associated with the `JDBCProvider` configurable service.

If you have been following the instructions in *Working with databases*, the next task is “Configuring ODBC connections for globally coordinated transactions” on page 399 (optional).

Related concepts:

Databases overview

Brokers can manipulate business data that is stored in databases. If you have databases, you must configure them before you can access them from your message flow.

Related tasks:

“Setting up a JDBC provider for type 4 connections” on page 386

Use the **mqsicreateconfigurable-service** or the **mqsichangeproperties** command to configure a JDBC provider service.

“Authorizing access to JDBC type 4 JDBCProvider resources on Windows”

Authorize the broker and queue manager to access shared resources that are associated with the JDBCProvider. This task is required only if you want the database updates to be included in globally coordinated transactions on Windows systems.

Configuring transactionality for message flows

A message flow runs in a single transaction, which is started when data is received by an input node, and can be committed or rolled back when all processing has completed.

Interacting with databases by using the JavaCompute node

Access databases from Java code included in the JavaCompute node.

Extending the capability of a Java message processing or output node

Within a message processing or output node, you can add extended functions to your Java node.

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.

Authorizing access to JDBC type 4 JDBCProvider resources on Windows:

Authorize the broker and queue manager to access shared resources that are associated with the JDBCProvider. This task is required only if you want the database updates to be included in globally coordinated transactions on Windows systems.

Before you begin

Before you start: Set up your JDBC provider definition.

About this task

When the queue manager coordinates transactions, both queue manager and broker access shared memory to control a connection to the databases with which the message flow interacts. Therefore, they require the same access control of the shared memory. One method to achieve this control is to use the same ID for the broker service ID and the queue manager administrative ID.

Complete the following steps on the Windows system on which the broker is running:

Procedure

- If you defined the broker queue manager when you created the broker by running the **mqsicreatebroker** command, the two components share the same administrative ID, defined as the broker service ID, and you do not have to take further action.
- If you specified an existing queue manager when you created the broker, check that its administrative ID is the same ID as the one used for the service ID of the broker. If the ID is not the same, change the queue manager ID to be the same as the broker service ID:
 1. Click **Start > Run** and enter `dcomcnfg`. The Component Services window opens.
 2. In the left pane, expand **Component Services > Computers > My Computer** and click **DCOM Config**.
 3. In the right pane, right-click the WebSphere MQ service labeled **IBM MQSeries Services**, and click **Properties**.
 4. Click the **Identity** tab.
 5. Select **This user** and enter the user ID and password for the broker service ID to associate that ID with the queue manager.
 6. Click **OK** to confirm the change.

What to do next

Next: If you have been following the instructions in *Working with databases*, the next task is “Configuring ODBC connections for globally coordinated transactions” on page 399 (optional).

Related concepts:



Databases overview

Brokers can manipulate business data that is stored in databases. If you have databases, you must configure them before you can access them from your message flow.

Related tasks:

“Setting up a JDBC provider for type 4 connections” on page 386

Use the **mqsicreateconfigurable-service** or the **mqsichangeproperties** command to configure a JDBC provider service.

“Securing database connections” on page 362

Set up security for a database connection, whether it is required by the database provider or optional.

“Configuring a JDBC type 4 connection for globally coordinated transactions” on page 391

If you want the database that you access through a JDBC type 4 connection to participate in globally coordinated transactions, set up the appropriate environment.



Interacting with databases by using the JavaCompute node

Access databases from Java code included in the JavaCompute node.



Extending the capability of a Java message processing or output node

Within a message processing or output node, you can add extended functions to your Java node.

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

Connecting to a database on z/OS:

Complete these tasks to connect to your databases on z/OS.

Before you begin

Before you start: Create the broker and the database.

Procedure

1. Obtain the following DB2 values for the DB2 subsystem that you want to connect your broker to. Use these values to customize the BIPEDIT, BIPDSNAO, and BIPBRKP files.

JCL variable	Description	Example value
++DB2CONVERSION++	The DB2 converter	SINGLE
++DB2SUBSYSTEM++	The DB2 subsystem ID which the component connects	DFK4
++DB2LOCATION++	The DB2 location value of the DB2 subsystem to which the component connects	DSN910PK
++DB2CURRENTSQLID++	The default DB2 user ID for connecting	MQP1BRK
++DB2DSNACLIPLAN++	The DB2 plan name.	DSNACLI
++DB2HLQ++	The DB2 high-level-qualifier	SYS2.DB2.V910
++DB2RUNLIB++	The DB2 run library value	DSN910PK.RUNLIB.LOAD

2. Copy the supplied sample BIPDSNAO file into the component data set, and customize its content based on the values obtained in step 1. The stanza for your database will look like the following content:

```
[COMMON]
APPLTRACE=0
APPLTRACEfilename="DD:APPLTRC"
TRACESTAMP=3
CONNECTTYPE=2
DIAGTRACE=0
DIAGTRACE_NO_WRAP=0
MAXCONN=0
MULTICONTEXT=0
MVSDEFAULTSSID=DFK4

; SUBSYSTEM
[DFK4]
MVSATTACHTYPE=RRSAF
PLANNAME=DSNACLI

; DATASOURCES
[DSN910PK]
CURRENTSQLID=MQP1BRK
```

3. Update the broker profile BIPBPROF in the component data set.
 - a. Check that the environment variable DSNAOINI points to your copy of the BIPDSNAO file in the component data set. By default, it is set to the correct value; update this variable if you use a different name.

- b. Check that the environment variable MQSI_DB2_CONVERSION is set correctly.
- 4. Submit BIPGEN to re-create the broker ENVFILE and any integration server specific profiles.
- 5. Update the EGENV and EGNOENV steps in the broker started task JCL to include the following DB2 data sets:

```
//STEPLIB DD DISP=SHR,DSN=&DB2HLQ..SDSNEXIT
//          DD DISP=SHR,DSN=&DB2HLQ..SDSNLOAD
//          DD DISP=SHR,DSN=&DB2HLQ..SDSNLOD2
//          DD DISP=SHR,DSN=++DB2RUNLIB++
```

Check that the PROC variable DB2HLQ='++DB2HLQ++' is not commented out.

- 6. Start the broker by using the **start (/S)** command.

What to do next

Next: You can set up your databases to participate in coordinated transactions by using the Resource Recovery Service (RRS). For more information, see “Resource Recovery Service planning on z/OS” on page 295.

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

“Database locations” on page 631

The broker can access databases that are set up on the local computer or on a remote server, subject to restrictions.

mqsisstart command

Use the **mqsisstart** command to start the specified broker if all initial verification tests complete successfully.

Configuring global coordination of transactions (two-phase commit)

Globally coordinate message flow transactions with a transaction manager to ensure the data integrity of transactions. On distributed systems, the WebSphere MQ queue manager that is associated with the broker performs the transaction manager role.

Before you begin

Before you start:

Read Message flow transactions to understand how the broker handles transactions. Depending on the external resource managers that the broker accesses during the processing of its deployed message flows, you must complete the appropriate resource-dependent set of tasks to ensure that all resources are configured correctly. For example, you might have to create and configure databases.

You, or your message flow developer, must also ensure that the message flows deployed to the broker are set up to support coordination. The tasks involved in configuring the message flows correctly are described in Configuring transactionality for message flows.

About this task

You can access the following external resources in a message flow transaction:

- WebSphere MQ queues and messages
- Databases
- JMS providers

On distributed platforms, the default behavior of the broker is to manage all message flow transactions by using a one-phase commit approach. In many contexts this approach is sufficient, but if your business requires assured data integrity and consistency (for example, for audit reasons, or for financial transactions), you can configure the broker and its WebSphere MQ queue manager to manage the message flow transactions in a two-stage commit approach, by using the XA protocol standard.

You configure the WebSphere MQ queue manager by updating its `qm.ini` file, to add definitions of the additional resource managers with which you want WebSphere MQ to coordinate updates. Follow the instructions provided for the resource managers that are relevant in your environment:

- ODBC connections to databases
- JDBC connections to databases
- JMS providers

z/OS On z/OS, all transactions are globally coordinated by Resource Recovery Service (RRS), therefore the instructions in this topic do not apply. However, you must ensure that RRS is available; see “Resource Recovery Service planning on z/OS” on page 295.

Results

When you have completed these steps, your message flows are processed by using global coordination, which is managed by the queue manager.

You must complete all the steps correctly; if you do not, global coordination will not work.

Example

For an example of how you can use WebSphere MQ to globally coordinate transactions, look at the following sample:

- Error Handler

You can view information about samples only when you use the information center that is integrated with the IBM Integration Toolkit or the online information center. You can run samples only when you use the information center that is integrated with the IBM Integration Toolkit.

Related concepts:



The transactional model

The transactional model describes the way in which you can use transactions in message flows to accomplish certain tasks and results.

Related tasks:



Configuring transactionality for message flows

A message flow runs in a single transaction, which is started when data is received by an input node, and can be committed or rolled back when all processing has completed.

“Configuring databases for global coordination of transactions” on page 365

If your message flow interacts with a database, and you want to globally coordinate the updates made to the database with other actions within the message flow, configure your databases for global coordination.

“Resource Recovery Service planning on z/OS” on page 295

“Enabling ODBC connections to the databases” on page 369

Set up the resources and environment that the broker requires for Open Database Connectivity (ODBC) connections to databases on distributed systems.

“Enabling JDBC connections to the databases” on page 384

Configure connections to a database through a JDBCProvider configurable service.

“Creating a broker” on page 304

You can create brokers on every platform that is supported by IBM Integration Bus. The broker runs as a 64-bit application on all platforms except Linux on x86 and Windows on x86.

Configuring ODBC connections for globally coordinated transactions:

Configure the definition of your ODBC databases to the transaction manager (the queue manager).

Before you begin

Before you start:

You must create and configure the databases by following the instructions in Working with databases.

You, or your message flow developer, must also ensure that the message flows deployed to the broker are set up to support coordination. The tasks that are involved in configuring the message flows correctly are described in Configuring transactionality for message flows.

Procedure

1. Ensure that your databases are configured for global coordination. For the databases that your message flows connect to, complete the tasks that are described in “Configuring databases for global coordination of transactions” on page 365.
2. Configure the broker environment so that the broker queue manager coordinates transactions with database resource managers. The steps to configure the broker environment depend on the database manager that you are using.
 - “Configuring global coordination with DB2” on page 400
 - “Configuring global coordination with Oracle” on page 405
 - “Configuring global coordination with Sybase” on page 410

Results

When you have completed these steps, your message flows are processed by using global coordination, which is managed by the queue manager.

You must complete all the steps correctly; if you do not, global coordination will not work.

Related concepts:



The transactional model

The transactional model describes the way in which you can use transactions in message flows to accomplish certain tasks and results.

Related tasks:



Configuring transactionality for message flows

A message flow runs in a single transaction, which is started when data is received by an input node, and can be committed or rolled back when all processing has completed.

“Configuring databases for global coordination of transactions” on page 365

If your message flow interacts with a database, and you want to globally coordinate the updates made to the database with other actions within the message flow, configure your databases for global coordination.

“Resource Recovery Service planning on z/OS” on page 295

“Enabling ODBC connections to the databases” on page 369

Set up the resources and environment that the broker requires for Open Database Connectivity (ODBC) connections to databases on distributed systems.

“Enabling JDBC connections to the databases” on page 384

Configure connections to a database through a JDBCProvider configurable service.

“Creating a broker” on page 304

You can create brokers on every platform that is supported by IBM Integration Bus. The broker runs as a 64-bit application on all platforms except Linux on x86 and Windows on x86.

Configuring global coordination with DB2:

Configure your broker environment to globally coordinate message flow transactions with updates in DB2 databases under the control of your broker queue manager.

Before you begin

Ensure that the databases are configured for global coordination of transactions, see “Configuring databases for global coordination of transactions” on page 365.

Procedure

Follow the instructions appropriate to your platform:

- **Linux** **UNIX** “Linux and UNIX”
- **Windows** “Windows 32-bit” on page 402
- **Windows** “Windows 64-bit” on page 403

Linux and UNIX: **Linux** **UNIX**

Procedure

1. **Linux** On Linux on x86: decide whether the broker needs to connect to databases by using TCP/IP or shared memory.

For more information about TCP/IP connections, see the example in the section about message SQL1224N in Resolving problems when using databases.

To enable shared memory:

- a. Stop the broker by running the following command, where *broker* is the name of your broker:
`mqsistop broker`
- b. Run the following command to ensure that the broker is run in an environment with the extended memory variable exported:
`export EXTSHM=ON`
- c. Restart the broker by running the following command, where *broker* is the name of your broker:
`mqsistart broker`
- d. On the DB2 server, ensure that shared memory support is turned on. For more information, see “Configuring databases for global coordination of transactions” on page 365.

2. Linux UNIX Run the `mqsimanagexalinks` command. For more information, see `mqsimanagexalinks` command.

3. Linux UNIX Configure the broker queue manager with XA resource manager information for each database that is involved in the transaction that the queue manager will globally coordinate.

- a. Open the queue manager's `qm.ini` file in a text editor. The `qm.ini` file is located at `/var/mqm/qmgrs/queue_manager_name/qm.ini`, where *queue_manager_name* is the name of the broker that is associated with the queue manager.

- b. At the end of the `qm.ini` file, paste the following stanza:

```
XAResourceManager:  
Name=DB2  
SwitchFile=db2swit  
XAOpenString=db=MyDataSource,uid=MyUserId,pwd=MyPassword,toc=t  
XACloseString=  
ThreadOfControl=THREAD
```

The switch file is supplied by IBM Integration Bus.

- c. On the `XAOpenString` line, replace the following values with values that are appropriate for your configuration:

- *MyDataSource* is the name of the data source to which you want to connect.
- *MyUserId* must be the user name that the broker uses to connect to the database.

You can define the user name that the broker uses in a number of ways; make sure that you specify the correct name in this file. The broker determines the user name by checking the following conditions in the order listed:

- 1) A specific user name and password for this data source name (DSN), that you have defined by running the `mqsisetdbparms` command.
- 2) A default user name and password for all DSNs, that you have defined by running the `mqsisetdbparms` command.
- 3) A default user name and password for all DSNs, that you have defined by specifying the `-u` parameter on the `mqsicreatebroker` command.

This parameter is valid only for brokers that you have migrated from previous releases.

- 4) The broker service user name, which you define with the `-i` parameter on the `mqsicreatebroker` command
 - *MyPassword* is the password that is associated with the user name.
- d. Accept the default values for all the other lines in the stanza. For example:

```
XAResourceManager:
Name=DB2
SwitchFile=db2swit
XAOpenString=db=MYDB,uid=wbrkuid,pwd=wbrkpw,toc=t
XACloseString=
ThreadOfControl=THREAD
```

4. Linux UNIX Stop then restart the queue manager to apply the changes, because `qm.ini` is read only while the queue manager is running.

To stop and restart the queue manager, enter the following commands, where *queue_manager_name* is the name of the queue manager:

```
endmqm queue_manager_name
strmqm queue_manager_name
```

When the queue manager restarts, check the queue manager log for all warnings that are associated with the restart. The log files are located in `/var/mqm/qmgrs/queue_manager_name/errors`, where *queue_manager_name* is the name of the queue manager that you restarted. When the queue manager restarts successfully, the changes that you made to `qm.ini` are applied.

Results

DB2 is now configured for global coordination with the broker queue manager coordinating transactions.

What to do next

Next: you can deploy globally coordinated message flows to the broker.

Windows 32-bit: Windows

Procedure

1. Windows Configure the broker queue manager with XA resource manager information for each database that is involved in the transaction that the queue manager will globally coordinate.
 - a. From the **Start** menu, open WebSphere MQ Explorer.
 - b. Open the queue manager Properties dialog box, then open **XA resource managers**.
 - c. Click the **Add** button to create a resource manager.
 - 1) In the **Name** field, enter a name to refer to a resource manager.
 - 2) In the **SwitchFile** field, enter the full path to the switch file, as shown in the following example where *install_dir* is the location in which the broker is installed:


```
install_dir\sampl\xatm\db2swit.dll
```
 - 3) In the **XAOpenString** field, paste the following string:


```
db=MyDataSource,uid=MyUserId,pwd=MyPassword,toc=t
```
 - 4) In the **XAOpenString** field, replace the values with values that are appropriate for your configuration:
 - a) *MyDataSource* is the name of the data source to which you want to connect.

- b) *MyUserId* must be the user name that the broker uses to connect to the database.

You can define the user name that the broker uses in a number of ways; make sure that you specify the correct name in this file. The broker determines the user name by checking the following conditions in the order listed:

- i. A specific user name and password for this data source name (DSN), that you have defined by running the **mqsisetdbparms** command.
- ii. A default user name and password for all DSNs, that you have defined by running the **mqsisetdbparms** command.
- iii. A default user name and password for all DSNs, that you have defined by specifying the **-u** parameter on the **mqsicreatebroker** command.

This parameter is valid only for brokers that you have migrated from previous releases.

- iv. The broker service user name, which you define with the **-i** parameter on the **mqsicreatebroker** command

- c) *MyPassword* is the password that is associated with the user name.

For example:

```
db=MYDB,uid=wbrkuid,pwd=wbrkpw,toc=t
```

- 5) Accept the default values for all the other fields on the page.

2. **Windows** Stop then restart the queue manager to apply the changes.

To stop and restart the queue manager, enter the following commands, where *queue_manager_name* is the name of the queue manager:

```
endmqm queue_manager_name  
strmqm -si queue_manager_name
```

When the queue manager restarts, check the queue manager log for all warnings that are associated with the restart. The log files are located in *install_dir*\WebSphere MQ\Qmgrs\IB9QMGR\errors, where *install_dir* is the location in which the broker is installed. When the queue manager restarts successfully, the changes that you made are applied.

Results

DB2 is now configured for global coordination with the broker queue manager coordinating transactions.

What to do next

Next: you can deploy globally coordinated message flows to the broker.

Windows 64-bit: **Windows**

Procedure

1. **Windows** Configure the broker queue manager with XA resource manager information for each database that is involved in the transaction that the queue manager will globally coordinate.
 - a. From the **Start** menu, open WebSphere MQ Explorer.
 - b. Open the queue manager Properties dialog box, then open **XA resource managers**.
 - c. Click the **Add** button to create a resource manager.

- 1) In the **Name** field, enter a name to refer to a resource manager.
- 2) In the **SwitchFile** field, enter *db2swit*
- 3) Copy the provided DB2 switch files from the broker install location to the queue managers exits and *exits64* directories (by default under C:\Program Files (x86)\IBM\WebSphere MQ).

Copy the file:

```
install_dir\sample\xatm\db2swit32.dll
```

into the queue managers exits directory, and rename it to *db2swit.dll*.

Copy the file

```
install_dir\sample\xatm\db2swit.dll
```

into the queue managers *exits64* directory.

- 4) In the **XAOpenString** field, paste the following string:
db=MyDataSource,uid=MyUserId,pwd=MyPassword,toc=t
- 5) In the **XAOpenString** field, replace the values with values that are appropriate for your configuration:
 - a) *MyDataSource* is the name of the data source to which you want to connect.
 - b) *MyUserId* must be the user name that the broker uses to connect to the database.

You can define the user name that the broker uses in a number of ways; make sure that you specify the correct name in this file. The broker determines the user name by checking the following conditions in the order listed:

- i. A specific user name and password for this data source name (DSN), that you have defined by running the **mqsisetdbparms** command.
- ii. A default user name and password for all DSNs, that you have defined by running the **mqsisetdbparms** command.
- iii. A default user name and password for all DSNs, that you have defined by specifying the **-u** parameter on the **mqsicreatebroker** command.

This parameter is valid only for brokers that you have migrated from previous releases.

- iv. The broker service user name, which you define with the **-i** parameter on the **mqsicreatebroker** command

- c) *MyPassword* is the password that is associated with the user name.

For example:

```
db=MYDB,uid=wbrkuid,pwd=wbrkpw,toc=t
```

- 6) Accept the default values for all the other fields on the page.

2. Windows Stop then restart the queue manager to apply the changes.

To stop and restart the queue manager, enter the following commands, where *queue_manager_name* is the name of the queue manager:

```
endmqm queue_manager_name
strmqm -si queue_manager_name
```

When the queue manager restarts, check the queue manager log for all warnings that are associated with the restart. The log files are located in

`install_dir\WebSphere MQ\Qmgrs\IB9QMGR\errors`, where `install_dir` is the location in which the broker is installed. When the queue manager restarts successfully, the changes that you made are applied.

Results

DB2 is now configured for global coordination with the broker queue manager coordinating transactions.

What to do next

Next: you can deploy globally coordinated message flows to the broker.

Related tasks:



Configuring transactionality for message flows

A message flow runs in a single transaction, which is started when data is received by an input node, and can be committed or rolled back when all processing has completed.

“Configuring ODBC connections for globally coordinated transactions” on page 399

Configure the definition of your ODBC databases to the transaction manager (the queue manager).

Related reference:



`mqsimanagexalinks` command

Use the `mqsimanagexalinks` command to set up the links required by WebSphere MQ to include supported databases in XA transactions.



`mqsicreatebroker` command

Use the `mqsicreatebroker` command to create a broker and its associated resources.

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

Related information:



WebSphere MQ Version 7 product documentation

Configuring global coordination with Oracle:

Configure your broker environment to globally coordinate message flow transactions with updates in Oracle databases under the control of a WebSphere MQ queue manager.

Before you begin

Before you start:

- Ensure that you have configured the databases for global coordination of transactions.

About this task

Configure your broker environment for global coordination by using a 64-bit queue manager as the transaction manager with the DataDirect drivers:

Procedure

1. **Linux** **UNIX** On Linux and UNIX, run the `mqsimanagexalinks` command.
2. **Windows** On Windows 64-bit, for the 64-bit version of IBM Integration Bus, run the `mqsimanagexalinks` command and add the directory specified as the queue manager's `ExitsDefaultPath` to the system PATH.
3. Configure the broker queue manager with XA resource manager information for each database that is involved in the transaction that the queue manager will globally coordinate.

Linux **UNIX** On Linux and UNIX:

- a. Open the queue managers `qm.ini` file in a text editor. The `qm.ini` file is located at `/var/mqm/qmgrs/queue_manager_name/qm.ini`. Where `queue_manager_name` is the name of the broker that is associated with the queue manager.
- b. Add the following stanza to the end of the `qm.ini` file:

```
XAResourceManager:  
Name=OracleXA  
SwitchFile=UKoradtc26.so  
XAOpenString=ORACLE_XA  
+HostName=MyHostName  
+PortNumber=MyPortNumber  
+ServiceName=MyServiceName  
+ACC=P/MyUserId/MyPassword  
+sestm=100+threads=TRUE  
+DataSource=MyDataSourceName  
+K=2+  
XACloseString=  
ThreadOfControl=THREAD
```

- c. On the **XAOpenString** line, replace the following values with values that are appropriate for your configuration:
 - *MyHostName* is the name of the TCP/IP host that hosts the Oracle database listener. When you are using Oracle Real Application Clusters with multiple listeners for the given Service Name, if the Oracle listener identified by the values for *MyHostName* and *MyPortNumber* in the **XAOpenString** is unavailable, the alternative Oracle listeners that you might have defined in the **AlternateServers** list in your `odbc.ini` file are also tried.
 - *MyPortNumber* is the TCP/IP port on which the Oracle database listener is listening.
 - *MyUserId* must be the user name that the broker uses to connect to the database.

You can define the user name that the broker uses in a number of ways; make sure that you specify the correct name in this file. The broker determines the user name by checking the following conditions in the order listed:

- 1) A specific user name and password for this data source name (DSN), that you have defined by running the `mqsisetdbparms` command.
- 2) A default user name and password for all DSNs, that you have defined by running the `mqsisetdbparms` command.
- 3) A default user name and password for all DSNs, that you have defined by specifying the `-u` parameter on the `mqsicreatebroker` command.

This parameter is valid only for brokers that you have migrated from previous releases.

- 4) The broker service user name, which you define with the **-i** parameter on the **mqsicreatebroker** command
 - *MyPassword* is the password that is associated with the user name.
 - *MyDataSourceName* is the ODBC data source name for the database, as defined in your `odbc.ini` file.
 - *MyServiceName* is the value set for the Service Name in the stanza for *MyDataSourceName* in your `odbc.ini` file.
- d. Accept the default values for all the other lines in the stanza. For example:
 - On AIX:

```
XAResourceManager:  
Name=OracleXA  
SwitchFile=UKoradtc26.so  
XAOpenString=ORACLE_XA  
  +HostName=diaz.hursley.ibm.com  
  +PortNumber=1521  
  +ServiceName=accounts_service  
  +ACC=P/wbrkuid/wbrkpw  
  +sestm=100+threads=TRUE  
  +DataSource=MYDB+K=2+  
XACloseString=  
ThreadOfControl=THREAD
```

- e. If you are using Global coordination to Oracle:
 - Set the ODBCINI environment variable to be visible in the environment from which you start your queue manager. The ODBCINI variable must reference the same file as the one being used by your broker.
 - An optional additional property, *CT0=Value*, is available for the **XAOpenString**. *CT0* is the value set for a Connection TimeOut, indicating the number of seconds that the Oracle XA switch file will wait for a response from the Oracle database to an XA request. For example, the timeout can be used to prevent long delays in the broker failing over to an alternative Oracle Real Application Clusters node, when the active Oracle instance fails abruptly leaving socket connections hanging. The *Value* should be set to a value larger than the Oracle session timeout *sestm* value set in the **XAOpenString**. If this property is not used, or is set to zero, there will be no timeout (this is the default behaviour).

Windows On Windows 32-bit and Windows 64-bit

- a. From the **Start** menu, open WebSphere MQ Explorer.
- b. Open the queue manager Properties dialog box, then open **XA resource managers**.
- c. In the **SwitchFile** field, enter the name of the switch file, as shown in the following example where *install_dir* is the location in which the broker is installed:
 - If you are using the 32-bit version of IBM Integration Bus:
`install_dir\bin\UKora26.dll`
 - If you are using the 64-bit version of IBM Integration Bus:
`UKora26.dll`
- d. In the **XAOpenString** field, paste the following string:

```
ORACLE_XA+  
  +HostName=MyHostName  
  +PortNumber=MyPortNumber
```

```
+ServiceName=MyServiceName
+ACC=P/MyUserId/MyPassword
+sestm=100+threads=TRUE
+DataSource=MyDataSourceName
+K=2+
```

- e. In the **XAOpenString** field, replace the values with values that are appropriate for your configuration:
- *MyHostName* is the name of the TCP/IP host that hosts the Oracle database listener. When you are using Oracle Real Application Clusters with multiple listeners for the given Service Name, if the Oracle listener identified by the values for *MyHostName* and *MyPortNumber* in the **XAOpenString** is unavailable, the alternative Oracle listeners that you might have defined in the **AlternateServers** list in your `odbc.ini` file are also tried.
 - *MyPortNumber* is the TCP/IP port on which the Oracle database listener is listening.
 - *MyUserId* must be the user name that the broker uses to connect to the database.

You can define the user name that the broker uses in a number of ways; make sure that you specify the correct name in this file. The broker determines the user name by checking the following conditions in the order listed:

- 1) A specific user name and password for this data source name (DSN), that you have defined by running the **mqsisetdbparms** command.
 - 2) A default user name and password for all DSNs, that you have defined by running the **mqsisetdbparms** command.
 - 3) A default user name and password for all DSNs, that you have defined by specifying the **-u** parameter on the **mqsicreatebroker** command.
This parameter is valid only for brokers that you have migrated from previous releases.
 - 4) The broker service user name, which you define with the **-i** parameter on the **mqsicreatebroker** command
- *MyPassword* is the password that is associated with the user name.
 - *MyDataSourceName* is the ODBC data source name for the database, as defined in your `odbc.ini` file.
 - *MyServiceName* is the value set for the Service Name in the ODBC definition for the data source *MyDataSourceName*.

For example:

```
ORACLE_XA+
+HostName=diaz.hursley.ibm.com
+PortNumber=1521
+ServiceName=accounts_service
+ACC=P/wbrkuid/wbrkpw
+sestm=100+threads=TRUE
+DataSource=MYDB+K=2+
```

- f. Accept the default values for all the other fields on the page.
4. **AIX** On AIX, if you want to enable Oracle data sources for use in global coordination from a queue manager and broker to perform dynamic XA registration, set the following environment variable:
DDTEK_XA_DYNAMIC_REGISTRATION=1

5. **Linux** **UNIX** Stop then restart the queue manager to apply the changes, because `qm.ini` is read only while the queue manager is running.

To stop and restart the queue manager, enter the following commands, where `queue_manager_name` is the name of the queue manager:

```
endmqm queue_manager_name
strmqm queue_manager_name
```

When the queue manager restarts, check the queue manager log for all warnings that are associated with the restart. The log files are located in `/var/mqm/qmgrs/queue_manager_name/errors`, where `queue_manager_name` is the name of the queue manager that you restarted. When the queue manager restarts successfully, the changes that you made to `qm.ini` are applied.

6. **Windows** Stop then restart the queue manager to apply the changes.

To stop and restart the queue manager, enter the following commands, where `queue_manager_name` is the name of the queue manager:

```
endmqm queue_manager_name
strmqm -si queue_manager_name
```

When the queue manager restarts, check the queue manager log for all warnings that are associated with the restart. The log files are located in `/var/mqm/qmgrs/queue_manager_name/errors`, where `queue_manager_name` is the name of the queue manager that you restarted. When the queue manager restarts successfully, the changes that you made are applied.

Results

Oracle is now configured for global coordination with the broker queue manager coordinating transactions.

What to do next

Next: You can deploy globally coordinated message flows to the broker.

Related tasks:

 **Configuring transactionality for message flows**

A message flow runs in a single transaction, which is started when data is received by an input node, and can be committed or rolled back when all processing has completed.

“Configuring ODBC connections for globally coordinated transactions” on page 399

Configure the definition of your ODBC databases to the transaction manager (the queue manager).

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

 **mqsicreatebroker** command

Use the **mqsicreatebroker** command to create a broker and its associated resources.

 **mqsimanagexalinks** command

Use the **mqsimanagexalinks** command to set up the links required by WebSphere MQ to include supported databases in XA transactions.



mqsisetdbparms command

Use the **mqsisetdbparms** command to associate a specific user ID and password (or SSH identity file) with one or more resources that are accessed by the broker.

Related information:



WebSphere MQ Version 7 product documentation

Configuring global coordination with Sybase:

Configure your broker environment to globally coordinate message flow transactions with updates in Sybase databases under the control of a queue manager.

Before you begin

Before you start:

- Ensure that the databases are configured for global coordination of transactions.

About this task

To configure your broker environment for global coordination using a WebSphere MQ queue manager as the transaction manager with the DataDirect drivers:

Procedure

1. Linux UNIX On Linux and UNIX, run the **mqsimanagexalinks** command.
2. Configure the broker queue manager with XA resource manager information for each database that is involved in the transaction that the queue manager will globally coordinate.

Linux

UNIX

On Linux and UNIX:

- a. Open the queue manager's `qm.ini` file in a text editor. The `qm.ini` file is located at `/var/mqm/qmgrs/queue_manager_name/qm.ini`, where `queue_manager_name` is the name of the broker that is associated with the queue manager.
- b. At the end of the `qm.ini` file, paste the following stanza:


```
XAResourceManager:
  Name=SYBASEXA
  SwitchFile=UKasedtc26.so
  XAOpenString=-NSYBASEDB -MyServerName,MyPortNumber -Uid -Ppwd -K2
  XACloseString=
  ThreadOfControl=THREAD
```
- c. On the **XAOpenString** line, replace the following values with values that are appropriate for your configuration:
 - *MyServerName* is the name of the TCP/IP host that hosts the Sybase ASE server.
 - *MyPortNumber* is the TCP/IP port on which the Sybase ASE server is listening.
 - *uid* must be the user name that the broker uses to connect to the database.

You can define the user name that the broker uses in a number of ways; make sure that you specify the correct name in this file. The broker determines the user name by checking the following conditions in the order listed:

- 1) A specific user name and password for this data source name (DSN), that you have defined by running the **mqsisetdbparms** command.
- 2) A default user name and password for all DSNs, that you have defined by running the **mqsisetdbparms** command.
- 3) A default user name and password for all DSNs, that you have defined by specifying the **-u** parameter on the **mqsicreatebroker** command.

This parameter is valid only for brokers that you have migrated from previous releases.

- 4) The broker service user name, which you define with the **-i** parameter on the **mqsicreatebroker** command
 - *pwd* is the password that is associated with the user name.
- d. Accept the default values for all the other lines in the stanza. For example:
 - On AIX:

```
XAResourceManager:
  Name=SYBASEXA
  SwitchFile=UKasedtc26.so
  XAOpenString=-NSYBASEDB -Adiaz,1521 -Uwbrkuid -Pwbrkpw -K2
  XACloseString=
  ThreadOfControl=THREAD
```

Windows On Windows 32-bit:

- a. From the **Start** menu, open WebSphere MQ Explorer.
- b. Open the Properties dialog box for the queue manager, then open **XA resource managers**.
- c. In the **SwitchFile** field, enter the full path to the switch file, as shown in the following example where *install_dir* is the location in which the broker is installed:


```
install_dir\bin\ukase26.dll
```
- d. In the **XAOpenString** field, paste the following string:


```
-NSYBASEDB -AMyServerName,MyPortNumber -WWinsock -Uuid -Ppwd -K2
```
- e. In the **XAOpenString** field, replace the values with values that are appropriate for your configuration:
 - *install_dir* is the location in which the broker is installed.
 - *MyServerName* is the name of the TCP/IP host that hosts the Sybase ASE server.
 - *MyPortNumber* is the TCP/IP port on which the Sybase ASE server is listening.
 - *uid* must be the user name that the broker uses to connect to the database.

You can define the user name that the broker uses in a number of ways; make sure that you specify the correct name in this file. The broker determines the user name by checking the following conditions in the order listed:

- 1) A specific user name and password for this data source name (DSN), that you have defined by running the **mqsisetdbparms** command.
- 2) A default user name and password for all DSNs, that you have defined by running the **mqsisetdbparms** command.
- 3) A default user name and password for all DSNs, that you have defined by specifying the **-u** parameter on the **mqsicreatebroker** command.

This parameter is valid only for brokers that you have migrated from previous releases.

- 4) The broker service user name, which you define with the **-i** parameter on the **mqsicreatebroker** command

- *pwd* is the password that is associated with the user name.

For example:

```
-NSYBASEDB -Adiaz,1521 -WWinsock -Uwbrkuid -Pwbrkpw -K2
```

- f. Accept the default values for all the other fields on the page.

3. Stop then restart the queue manager to apply the changes, because *qm.ini* is read only while the queue manager is running.

To stop and restart the queue manager, enter the following commands, where *queue_manager_name* is the name of the queue manager:

```
endmqm queue_manager_name  
strmqm queue_manager_name
```

When the queue manager restarts, check the queue manager log for all warnings that are associated with the restart. The log files are located in */var/mqm/qmgrs/queue_manager_name/errors*, where *queue_manager_name* is the name of the queue manager that you restarted. When the queue manager restarts successfully, the changes that you made to *qm.ini* are applied.

Results

Sybase is now configured for global coordination with your queue manager coordinating transactions.

What to do next

Next: you can deploy globally coordinated message flows to the broker.

Related tasks:



Configuring transactionality for message flows

A message flow runs in a single transaction, which is started when data is received by an input node, and can be committed or rolled back when all processing has completed.

“Configuring ODBC connections for globally coordinated transactions” on page 399

Configure the definition of your ODBC databases to the transaction manager (the queue manager).

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.



mqsicreatebroker command - Linux and UNIX systems

Use the **mqsicreatebroker** command to create a broker on a Linux or UNIX systems.



mqsicreatebroker command - Windows systems

Use the **mqsicreatebroker** command to create a broker on a Windows system.



mqsimanagexalinks command

Use the **mqsimanagexalinks** command to set up the links required by WebSphere

MQ to include supported databases in XA transactions.



mqsisetdbparms command

Use the **mqsisetdbparms** command to associate a specific user ID and password (or SSH identity file) with one or more resources that are accessed by the broker.

Related information:



WebSphere MQ Version 7 product documentation

Configuring a JDBC type 4 connection for globally coordinated transactions:

If you want the database that you access through a JDBC type 4 connection to participate in globally coordinated transactions, set up the appropriate environment.

Before you begin

Before you start: Set up your JDBC provider definition.

About this task

Updates that you make to a database across a JDBC type 4 connection can be coordinated with other actions taken within the message flow, if you set up the resources to support coordination.

Complete the following steps:

Procedure



1. Check that the definition of your JDBCProvider service is appropriate for coordinated transactions.

For example, to set up the required JDBC classes:

- For DB2, set **type4DataSourceClassName** to `com.ibm.db2.jcc.DB2XADataSource` and **type4DriverClassName** to `com.ibm.db2.jcc.DB2Driver`
- For Oracle, set **type4DataSourceClassName** to `oracle.jdbc.xa.client.OracleXADataSource` and **type4DriverClassName** to `oracle.jdbc.OracleDriver`

Consult your database administrator or the documentation provided by your database supplier, to confirm that all the JDBCProvider service properties are set appropriately. For example, a database supplier might require secure access if it is participating in coordinated transactions.

2. Define the switch file and the database properties:

- a.   On Linux and UNIX systems, open the `qm.ini` file for the broker queue manager with a text editor. Add the following stanza for each database:

```
XAResourceManager:  
  Name=Database_Name  
  SwitchFile=JDBCSwitch  
  XAOpenString=JDBC_DataSource  
  ThreadOfControl=THREAD
```

Database_Name is the database name (DSN) of the database defined to the JDBCProvider configurable service (for example, specified by `-n databaseName -v Database_Name` on the **mqsichangeproperties** command).

JDBCSwitch is a fixed generic name that represents the switch file for XA coordination. Use this value, or another single fixed value, in each stanza;

the specific switch file that the queue manager uses is defined by the symbolic links you create in the next step.

JDBC_DataSource is the identifier of the JDBCProvider configurable service (the value that you specified for the **-o** parameter on the **mqsichangeproperties** command).

Define a stanza for each database (DSN) that you connect to from this broker. You must create separate definitions even if the DSNs resolve to the same physical database. Therefore, you must have a stanza for each JDBCProvider configurable service that you have defined, because each service can define the properties for a single database.

- b. **Windows** On Windows on x86 systems, open WebSphere MQ Explorer and select the queue manager for your broker, for example BROKERQM.

Open the **XA resource manager** page, and modify the attributes to create the definition of the database. The attributes are the same as those shown for Linux and UNIX; **Name**, **SwitchFile**, **XAOpenString**, and **ThreadofControl**. Leave the additional attribute, **XACloseString**, blank.

Enter the fully qualified file name in **SwitchFile**; *install_dir*\bin\JDBCSwitch.dll.

- c. **Windows** On Windows on x86-64 systems, open WebSphere MQ Explorer and select the queue manager for your broker, for example BROKERQM.

Open the **XA resource manager** page, and modify the attributes to create the definition of the database. The attributes are the same as those shown for Linux and UNIX; **Name**, **SwitchFile**, **XAOpenString**, and **ThreadofControl**. Leave the additional attribute, **XACloseString**, blank.

Enter *JDBCswitch* in **SwitchFile**.

- 3. Set up queue manager access to the switch file:

- a. **Linux** **UNIX** On Linux and UNIX systems, create a symbolic link to the switch files that are supplied in your *install_dir/lib* directory.

install_dir is the directory to which you installed the Integration Bus component. The default location for this directory is */opt/ibm/mqsi/v.r.m.f* on Linux, or */opt/IBM/mqsi/v.r.m.f* on UNIX systems. The default directory includes the version, release, modification, and fix of the product, in the format v.r.m.f (version.release.modification.fix).

Set up links in the */var/mqm/exits* directory, or the */var/mqm/exits64* directory, or both. The file names for each platform are shown in the following table.

Platform	32-bit file	64-bit file
AIX		libJDBCSwitch.so
HP-Itanium		libJDBCSwitch.so
Linux on POWER		libJDBCSwitch.so
Linux on System z		libJDBCSwitch.so
Linux on x86	libJDBCSwitch.so	
Linux on x86-64		libJDBCSwitch.so
Solaris on SPARC		libJDBCSwitch.so
Solaris on x86-64		libJDBCSwitch.so

Specify the same name of the switch file, *JDBCswitch* or your own value, in both the */exits* and */exits64* directories. For example, on AIX:

```
ln -s install_dir/lib/libJDBCSwitch.so /var/mqm/exits/JDBCSwitch
```

and

```
ln -s install_dir/lib/libJDBCSwitch.so /var/mqm/exits64/JDBCSwitch
```

- b. **Windows** On Windows, for the 32-bit version of IBM Integration Bus, copy the JDBCSwitch.dll file from the *install_dir*\bin directory to the \exits subdirectory in the WebSphere MQ installation directory.
 - c. **Windows** On Windows, for the 64-bit version of IBM Integration Bus, copy the JDBCSwitch32.dll file from the *install_dir*\bin directory to the \exits subdirectory in the WebSphere MQ installation directory, and rename the file to JDBCSwitch.dll. Then, copy the JDBCSwitch.dll file from the *install_dir*\bin directory to the \exits64 subdirectory in the WebSphere MQ installation directory.
4. Configure the message flow that includes one or more nodes that access databases that are to participate in a globally coordinated transaction.
 - a. Open an IBM Integration Toolkit session.
 - b. Switch to the Integration Development perspective.
 - c. Add the message flow that includes the node or nodes that connect to the database that is to participate in a globally coordinated transaction to a new or existing BAR file.
 - d. Build the BAR file.
 - e. Click the **Configure** tab, select the message flow that you have added, and select the **Coordinated Transaction** check box.

What to do next

Next: If your broker is running on Windows, authorize the broker and its queue manager to access resources associated with the JDBCProvider configurable service.

If you have been following the instructions in Working with databases, the next task is “Configuring ODBC connections for globally coordinated transactions” on page 399 (optional).

Related concepts:



Databases overview

Brokers can manipulate business data that is stored in databases. If you have databases, you must configure them before you can access them from your message flow.

Related tasks:

“Setting up a JDBC provider for type 4 connections” on page 386

Use the **mqsicreateconfigurableservice** or the **mqsichangeproperties** command to configure a JDBC provider service.

“Authorizing access to JDBC type 4 JDBCProvider resources on Windows” on page 394

Authorize the broker and queue manager to access shared resources that are associated with the JDBCProvider. This task is required only if you want the database updates to be included in globally coordinated transactions on Windows systems.



Configuring transactionality for message flows

A message flow runs in a single transaction, which is started when data is received by an input node, and can be committed or rolled back when all processing has

completed.



Interacting with databases by using the JavaCompute node
Access databases from Java code included in the JavaCompute node.



Extending the capability of a Java message processing or output node
Within a message processing or output node, you can add extended functions to your Java node.

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.



mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.

Configuring properties to connect to external resources

You can use configurable services to prepare the environment for external resources, or change connection details for external resources.

About this task

The following topics describe how to prepare your environment to connect to external resources.

- “Configuring for Enterprise Information Systems”
- “Configuring for IMS” on page 430
- Configuring for CORBA
- “Configuring for CICS Transaction Server for z/OS” on page 436
- “Configuring for JMS” on page 447
- Configuring for WS-Security
- Configuring for email
- Configuring for TCP/IP

Configuring for Enterprise Information Systems:

You can use configurable services to enable the WebSphere Adapters nodes in the broker runtime environment to connect with Enterprise Information Systems such as SAP, Siebel, and PeopleSoft.

About this task

The following topics describe how to prepare the environment to connect to Enterprise Information Systems, and how to change the connection details for adapters without the need to redeploy them.

- “Preparing the environment for WebSphere Adapters nodes” on page 417
- “Changing connection details for SAP adapters” on page 418
- “Changing connection details for Siebel adapters” on page 420
- “Changing connection details for PeopleSoft adapters” on page 422
- “Changing connection details for JD Edwards adapters” on page 424
- “Configuring EIS connections to expire after a specified time” on page 426

Preparing the environment for WebSphere Adapters nodes:

Before you can use the WebSphere Adapters nodes, you must set up the broker runtime environment so that you can access the Enterprise Information System (EIS).

Before you begin

Before you start:

Read WebSphere Adapters nodes.

About this task

To enable the WebSphere Adapters nodes in the broker runtime environment, configure the broker with the location of the EIS provider JAR files and native libraries. (On Windows, the location of the JAR files cannot be a mapped network drive on a remote Windows computer; the directory must be local or on a Storage Area Network (SAN) disk.)

Procedure

1. The WebSphere Adapters nodes require libraries from the EIS vendors. For more information on how to obtain and use these libraries, see *Developing message flows that use WebSphere Adapters*.
2. Use the following commands to make the JAR files and shared libraries available to the WebSphere Adapters nodes.

- To set up the dependencies, use the following command.

```
mqschangeproperties broker name -c EISProviders -o EIS type -n jarsURL -v jar directory  
mqschangeproperties broker name -c EISProviders -o EIS type -n nativeLibs -v bin directory
```

For example:

```
mqschangeproperties brk6 -c EISProviders -o SAP -n jarsURL -v c:\sapjco\jars  
mqschangeproperties brk6 -c EISProviders -o SAP -n nativeLibs -v c:\sapjco\bin
```

After you have run the **mqschangeproperties** command, restart the broker so that the changes take effect.

- To report the dependencies, use the following command.

```
mqsireportproperties broker name -c EISProviders -o EIS type -r
```

For example:

```
mqsireportproperties brk6 -c EISProviders -o SAP -r
```

- On z/OS, run this command by customizing and submitting BIPJADPR.

You can also connect to different versions of Siebel by creating a custom EISProviders configurable service and setting the location of the appropriate library files. For more information, see *Connecting to different versions of Siebel*.

What to do next

When you have set up the environment for the WebSphere Adapters nodes, you can perform the preparatory tasks that are listed in *Developing message flows that use WebSphere Adapters*.

Related concepts:

WebSphere Adapters nodes

A WebSphere Adapters node is a message flow node that is used to communicate with Enterprise Information Systems (EIS), such as SAP, Siebel, JD Edwards, and PeopleSoft.

Related tasks:

Connecting to an EIS by using the Adapter Connection wizard

Use the Adapter Connection wizard to create the resources that enable the WebSphere Adapters to connect to an Enterprise Information System (EIS).

Debugging message flows that contain WebSphere Adapters nodes

You can use various methods to monitor message flows that include WebSphere Adapters nodes.

Related reference:

WebSphere Adapters properties

Reference information about the properties that you set for WebSphere Adapters nodes.

Changing connection details for SAP adapters:

SAP nodes can get SAP connection details from either the adapter component or a configurable service. By using configurable services, you can change the connection details for adapters without the need to redeploy the adapters. To pick up new values when a configurable service is created or modified, you must reload the broker or integration server to which the adapter was deployed, by using the **mqsistop** and **mqsistart** commands, or the **mqsireload** command.

Before you begin

Before you start:

- Read WebSphere Adapters nodes and Configurable services for background information.

About this task

Use the SAPConnection configurable service to change connection details for an SAP adapter. The SAP node reads all connection properties from the adapter component that it is configured to use. If a configurable service exists that has the same name as the node's adapter component, the node uses the values that are defined in that configurable service to override the corresponding properties from the adapter. If any properties on the configurable service are set to an empty string, the values that are configured in the `.inadapter` or `.outadapter` file are used. The properties of the SAP configurable services are described in Configurable services properties.

Creating, changing, reporting, and deleting configurable services

Procedure

- To create a configurable service, use the IBM Integration Explorer; see “Using the IBM Integration Explorer to work with configurable services” on page 345 for more information, or the **mqsicreateconfigurable-service** command, as shown in the following example. This example creates an SAPConnection

configurable service for the SAP adapter *mySAPAdapter.outadapter* that connects to the SAP host *test.sap.ibm.com*, and uses client *001* for the connections into that server:

```
mqsicreateconfigurableservice IB9NODE -c SAPConnection  
-o mySAPAdapter.outadapter -n applicationServerHost,client  
-v test.sap.ibm.com,001
```

To pick up the new values in the configurable service, restart the integration server and message flow.

- To change a configurable service, use the IBM Integration Explorer, or the **mqsichangeproperties** command, as shown in the following example. This example changes the connections that are used by the adapter *mySAPAdapter.outadapter*. After you run this command, all adapters connect to the production system (*production.sap.ibm.com*) instead of the test system (*test.sap.ibm.com*):

```
mqsichangeproperties IB9NODE -c SAPConnection -o mySAPAdapter.outadapter  
-n applicationServerHost -v production.sap.ibm.com
```

To pick up the updated values in the configurable service, restart the integration server and message flow.

- To display all SAPConnection configurable services, use the IBM Integration Explorer, or the **mqsireportproperties** command, as shown in the following example:

```
mqsireportproperties IB9NODE -c SAPConnection -o AllReportableEntityNames -r
```

- You can delete a configurable service that you have created by using the IBM Integration Explorer, or the **mqsdeleteconfigurableservice** command, as shown in the following example:

```
mqsdeleteconfigurableservice IB9NODE -c SAPConnection  
-o mySAPAdapter.outadapter
```

You can delete only custom-named configurable services; you cannot delete IBM-defined configurable services.

Related concepts:



WebSphere Adapters nodes

A WebSphere Adapters node is a message flow node that is used to communicate with Enterprise Information Systems (EIS), such as SAP, Siebel, JD Edwards, and PeopleSoft.



Configurable services

Configurable services are typically runtime properties. You can use them to define properties that are related to external services on which the broker relies; for example, an SMTP server or a JMS provider.

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related reference:



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

mqsicreateconfigurable-service command

Use the **mqsicreateconfigurable-service** command to create an object name for a broker external resource.

mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.

mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.

mqsdeleteconfigurable-service command

Use the **mqsdeleteconfigurable-service** command to delete a configurable service, such as a JMS provider, JDBC provider, or FTP server, that you have created by using the **mqsicreateconfigurable-service** command.

SAPInput node

Use the SAPInput node to accept input from an SAP application.

SAPRequest node

Use the SAPRequest node to send requests to an SAP application.

Changing connection details for Siebel adapters:

Siebel nodes can get Siebel connection details from either the adapter component or a configurable service. By using configurable services, you can change the connection details for adapters without the need to redeploy the adapters. To pick up new values when a configurable service is created or modified, you must reload the broker or integration server to which the adapter was deployed, by using the **mqsistop** and **mqsistart** commands, or the **mqsireload** command.

Before you begin

Before you start:

- Read WebSphere Adapters nodes and Configurable services for background information.

About this task

Use the SiebelConnection configurable service to change connection details for a Siebel adapter. The Siebel node reads all connection properties from the adapter component that it is configured to use. If a configurable service exists that has the same name as the adapter component of the node, the node uses the values that are defined in that configurable service to override the corresponding properties from the adapter. If a configurable service is being used, all properties that are exposed by the configurable service are taken from the configurable service. The only properties that are taken from the adapter are those that you cannot set on the configurable service. The properties of the Siebel configurable service are described in Configurable services properties.

You can also connect to different versions of Siebel by creating a custom EISProviders configurable service and setting the location of the appropriate library files. For more information, see Connecting to different versions of Siebel.

Creating, changing, reporting, and deleting configurable services

Procedure

- To create a configurable service, use the IBM Integration Explorer; see “Using the IBM Integration Explorer to work with configurable services” on page 345 for more information, or the **mqscreateconfigurable** command, as shown in the following example. This example creates a SiebelConnection configurable service for the Siebel instance that is running on *my.siebel.qa.com*:

```
mqscreateconfigurable IB9NODE -c SiebelConnection -o mySiebelAdapter.outadapter  
-n connectionString -v "siebel://my.siebel.qa.com/SBA_80/SSEObjMgr_enu"
```

To pick up the new values in the configurable service, restart the integration server and message flow.

- To change a configurable service, use the IBM Integration Explorer, or the **mqschange** command, as shown in the following example. This example changes the connections that are used by the adapter *mySiebelAdapter.outadapter*. After you run this command, all adapters connect to the production system (*my.siebel.production.com*) instead of the test system (*my.siebel.qa.com*):

```
mqschange IB9NODE -c SiebelConnection -o mySiebelAdapter.outadapter -n connectionString  
-v "siebel://my.siebel.production.com/SBA_80/SSEObjMgr_enu"
```

To pick up the updated values in the configurable service, restart the integration server and message flow.

- To display all SiebelConnection configurable services, use the IBM Integration Explorer, or the **mqsireport** command, as shown in the following example:

```
mqsireport IB9NODE -c SiebelConnection -o AllReportableEntityNames -r
```

- You can delete a configurable service that you have created by using the IBM Integration Explorer, or the **mqsdelete** command, as shown in the following example:

```
mqsdelete IB9NODE -c SiebelConnection -o mySiebelAdapter.outadapter
```

You can delete only custom-named configurable services; you cannot delete IBM-defined configurable services.

Related concepts:



WebSphere Adapters nodes

A WebSphere Adapters node is a message flow node that is used to communicate with Enterprise Information Systems (EIS), such as SAP, Siebel, JD Edwards, and PeopleSoft.



Configurable services

Configurable services are typically runtime properties. You can use them to define properties that are related to external services on which the broker relies; for example, an SMTP server or a JMS provider.

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related reference:

Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

mqsicreateconfigurable command

Use the **mqsicreateconfigurable** command to create an object name for a broker external resource.

mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.

mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.

mqsdeleteconfigurable command

Use the **mqsdeleteconfigurable** command to delete a configurable service, such as a JMS provider, JDBC provider, or FTP server, that you have created by using the **mqsicreateconfigurable** command.

SiebelInput node

Use the SiebelInput node to interact with a Siebel application.

SiebelRequest node

Use the SiebelRequest node to interact with a Siebel application.

Changing connection details for PeopleSoft adapters:

PeopleSoft nodes can get PeopleSoft connection details from either the adapter component or a configurable service. By using configurable services, you can change the connection details for adapters without the need to redeploy the adapters. To pick up new values when a configurable service is created or modified, you must reload the broker or integration server to which the adapter was deployed, by using the **mqsistop** and **mqsistart** commands, or the **mqsireload** command.

Before you begin

Before you start:

- Read WebSphere Adapters nodes and Configurable services for background information.

About this task

Use the PeopleSoftConnection configurable service to change connection details for a PeopleSoft adapter. The PeopleSoft node reads all connection properties from the adapter component that it is configured to use. If a configurable service exists that has the same name as the node's adapter component, the node uses the values that are defined in that configurable service to override the corresponding properties from the adapter. If a configurable service is being used, all properties that are exposed by the configurable service are taken from the configurable service. The only properties that are taken from the adapter are those that you cannot set on the configurable service. The properties of the PeopleSoft configurable service are described in Configurable services properties.

Creating, changing, reporting, and deleting configurable services

Procedure

- To create a configurable service, use the IBM Integration Explorer; see “Using the IBM Integration Explorer to work with configurable services” on page 345 for more information, or the **mqsicreateconfigurable** command, as shown in the following example. This example creates a PeopleSoftConnection configurable service for the PeopleSoft instance that is running on *my.peoplesoft.qa.com*:

```
mqsicreateconfigurable IB9NODE -c PeopleSoftConnection -o myPeopleSoftAdapter.outadapter -n hostName,port -v "my.peoplesoft.qa.com",9000
```

To pick up the new values in the configurable service, restart the integration server and message flow.

- To change a configurable service, use the IBM Integration Explorer, or the **mqsichangeproperties** command, as shown in the following example. This example changes the connections that are used by the adapter *myPeopleSoftAdapter.outadapter*. After you run this command, all adapters connect to the production system (*my.peoplesoft.production.com*) instead of the test system (*my.peoplesoft.qa.com*):

```
mqsichangeproperties IB9NODE -c PeopleSoftConnection -o myPeopleSoftAdapter.outadapter -n hostName -v "my.peoplesoft.production.com"
```

To pick up the updated values in the configurable service, restart the integration server and message flow.

- To display all PeopleSoftConnection configurable services, use the IBM Integration Explorer, or the **mqsireportproperties** command, as shown in the following example:

```
mqsireportproperties IB9NODE -c PeopleSoftConnection -o AllReportableEntityNames -r
```

- You can delete a configurable service that you have created by using the IBM Integration Explorer, or the **mqsdeleteconfigurable** command, as shown in the following example:

```
mqsdeleteconfigurable IB9NODE -c PeopleSoftConnection -o myPeopleSoftAdapter.outadapter
```

You can delete only custom-named configurable services; you cannot delete IBM-defined configurable services.

Related concepts:



WebSphere Adapters nodes

A WebSphere Adapters node is a message flow node that is used to communicate with Enterprise Information Systems (EIS), such as SAP, Siebel, JD Edwards, and PeopleSoft.



Configurable services

Configurable services are typically runtime properties. You can use them to define properties that are related to external services on which the broker relies; for example, an SMTP server or a JMS provider.

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related reference:

Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

mqsicreateconfigurable command

Use the **mqsicreateconfigurable** command to create an object name for a broker external resource.

mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.

mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.

mqsdeleteconfigurable command

Use the **mqsdeleteconfigurable** command to delete a configurable service, such as a JMS provider, JDBC provider, or FTP server, that you have created by using the **mqsicreateconfigurable** command.

PeopleSoftInput node

Use the PeopleSoftInput node to interact with a PeopleSoft application.

PeopleSoftRequest node

Use the PeopleSoftRequest node to interact with a PeopleSoft application.

Changing connection details for JD Edwards adapters:

JD Edwards nodes can get JD Edwards EnterpriseOne connection details from either the adapter component or a configurable service. By using configurable services, you can change the connection details for adapters without the need to redeploy the adapters. To pick up new values when a configurable service is created or modified, you must reload the broker or integration server to which the adapter was deployed, by using the **mqsistop** and **mqsistart** commands, or the **mqsireload** command.

Before you begin

Before you start:

- Read WebSphere Adapters nodes and Configurable services for background information.

About this task

Use the JDEdwardsConnection configurable service to change connection details for a JD Edwards adapter. The JD Edwards node reads all connection properties from the adapter component that it is configured to use. If a configurable service exists that has the same name as the nodes adapter component, the node uses the values that are defined in that configurable service to override the corresponding properties from the adapter. If a configurable service is being used, all properties that are exposed by the configurable service are taken from the configurable service. The only properties that are taken from the adapter are the ones that you cannot set on the configurable service. The properties of the JD Edwards configurable service are described in Configurable services properties.

Creating, changing, reporting, and deleting configurable services

Procedure

- To create a configurable service, use the IBM Integration Explorer; see “Using the IBM Integration Explorer to work with configurable services” on page 345 for more information, or the **mqsicreateconfigurableservice** command, as shown in the following example. This example creates a JDEdwardsConnection configurable service to connect to the DV7333 development Environment for a user with the Role of administrator.

```
mqsicreateconfigurableservice MYBROKER -c JDEdwardsConnection -o myJdedwardsAdapter.outadapter -n Environment,Role -v "dv7333,administrator"
```

To pick up the new values in the configurable service, restart the integration server and message flow.

- To change a configurable service, use the IBM Integration Explorer, or the **mqsichangeproperties** command, as shown in the following example. This example changes the connection that is used by the adapter *myJdedwardsAdapter.outadapter* to connect to the PD7333 production Environment. After you run this command, all adapters connect to the production Environment (PD7333) instead of the development system (DV7333).

```
mqsichangeproperties MYBROKER -c JDEdwardsConnection -o myJdedwardsAdapter.outadapter -n Environment,Role -v PD7333,administrator
```

To pick up the updated values in the configurable service, restart the integration server and message flow.

- To display all JDEdwardsConnection configurable services, use the IBM Integration Explorer, or the **mqsireportproperties** command, as shown in the following example:

```
mqsireportproperties MYBROKER -c JDEdwardsConnection -o AllReportableEntityNames -r
```

- You can delete a configurable service that has been created by using the IBM Integration Explorer, or the **mqsdeleteconfigurableservice** command, as shown in the following example:

```
mqsdeleteconfigurableservice MYBROKER -c JDEdwardsConnection -o myJdedwardsAdapter.outadapter
```

You can delete only custom-named configurable services; you cannot delete IBM-defined configurable services.

Related concepts:



WebSphere Adapters nodes

A WebSphere Adapters node is a message flow node that is used to communicate with Enterprise Information Systems (EIS), such as SAP, Siebel, JD Edwards, and PeopleSoft.



Configurable services

Configurable services are typically runtime properties. You can use them to define properties that are related to external services on which the broker relies; for example, an SMTP server or a JMS provider.

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related reference:

Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

`mqscreateconfigurable-service` command

Use the `mqscreateconfigurable-service` command to create an object name for a broker external resource.

`mqschange-properties` command

Use the `mqschange-properties` command to modify broker properties and properties of broker resources.

`mqsreport-properties` command

Use the `mqsreport-properties` command to display properties that relate to a broker, an integration server, or a configurable service.

`mqsdeleteconfigurable-service` command

Use the `mqsdeleteconfigurable-service` command to delete a configurable service, such as a JMS provider, JDBC provider, or FTP server, that you have created by using the `mqscreateconfigurable-service` command.

JDEdwardsRequest node

Use the JDEdwardsRequest node to interact with a JD Edwards EnterpriseOne server.

Configuring EIS connections to expire after a specified time:

You can configure connections to SAP, Siebel, and PeopleSoft to expire after a specified time by using a configurable service.

About this task

You can use the `connectionIdleTimeout` property on a configurable service to control the number of connections to SAP, Siebel and PeopleSoft by closing connections that have not been used for a specified time.

Procedure

Use the `mqscreateconfigurable-service` command to set up connections that expire when they have not been used for a specified length of time. In the following example, the `SAPConnection` configurable service is configured to close connections when they have not been used for 120 seconds.

```
mqscreateconfigurable-service IB9NODE -c SAPConnection -o mySAPAdapter.outadapter  
-n connectionIdleTimeout -v 120
```

Related concepts:

WebSphere Adapters nodes

A WebSphere Adapters node is a message flow node that is used to communicate with Enterprise Information Systems (EIS), such as SAP, Siebel, JD Edwards, and PeopleSoft.

Related tasks:

Developing message flows that use WebSphere Adapters

For information about how to develop message flows that use WebSphere Adapters, see the following topics.

Related reference:



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.



mqscreateconfigurable command

Use the **mqscreateconfigurable** command to create an object name for a broker external resource.



mqsdeleteconfigurable command

Use the **mqsdeleteconfigurable** command to delete a configurable service, such as a JMS provider, JDBC provider, or FTP server, that you have created by using the **mqscreateconfigurable** command.



mqsreportproperties command

Use the **mqsreportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.



mqschangeproperties command

Use the **mqschangeproperties** command to modify broker properties and properties of broker resources.



WebSphere Adapters properties

Reference information about the properties that you set for WebSphere Adapters nodes.

Advanced configuration properties when using IBM Sterling Connect:Direct nodes:

CDInput and CDOOutput nodes can get connection details and staging directories in conjunction with a configurable service. To pick up new values when a configurable service is created or modified, you must reload the broker or integration server, by using the **mqsistop** and **mqsistart** commands, or the **mqsireload** command.

Before you begin

Before you start:

- Read IBM Sterling Connect:Direct® overview and concepts and Configurable services for background information.

About this task

Use the *CDServer* configurable service to change connection details for an IBM Sterling Connect:Direct node. The properties of the *CDServer* configurable services are described in *CDServer* configurable service.

Use the following scenarios as examples of how you use the staging directories.

Procedure

- **What should I do if I have IBM Integration Bus and IBM Sterling Connect:Direct on the same machine, and want the files to be staged for output within the IBM Integration Bus work path?**

You need not set any of the properties; the default settings allow this to happen.

- **Output**

1. **What should I do if I have IBM Integration Bus and IBM Sterling Connect:Direct on the same machine but want to use a specified location to create files for output?** Set the brokerPathToStagingDir to the location you want. You do not need to set the cdPathToStagingDir, as IBM Integration Bus assumes this to be the same as the brokerPathToStagingDir.
2. **Can I have IBM Integration Bus and IBM Sterling Connect:Direct on separate machines?** You can, but you must ensure that both machines have:
 - Access to the same shared filesystem.
 - The same timezone setting.

As IBM Sterling Connect:Direct does not support NFS file systems on z/OS, this configuration does not work on z/OS

3. **Can I have this shared filesystem mounted at different places on my IBM Integration Bus and IBM Sterling Connect:Direct machines for platforms other than Windows?** Set the brokerPathToStagingDir to be the location where it is mounted on the IBM Integration Bus machine, and the cdPathToStagingDir to where it is mounted on the IBM Sterling Connect:Direct machine.
4. **Can I have this shared filesystem mounted at different places on my IBM Integration Bus and IBM Sterling Connect:Direct machines for Windows platforms?** You must use the \\hostname\directory path syntax to the shared drive, rather than a mapped drive letter. Moreover, the user ID accessing the \\hostname\directory path, that is the:

IBM Integration Bus user ID, or

IBM Sterling Connect:Direct user ID

- depending on which one is running on Windows, must have full (write) access to the file system, using the same password.

Set the brokerPathToStagingDir to be the location where it is mounted on the IBM Integration Bus machine, and the cdPathToStagingDir to where it is mounted on the IBM Sterling Connect:Direct machine.

- **Input**

1. **Can I process files in the CDInput node if the IBM Integration Bus and receiving Connect:Direct server are on different machines?** You can, but you must ensure that both machines have:
 - Access to the same shared filesystem.
 - The same timezone setting.

As IBM Sterling Connect:Direct does not support NFS file systems on z/OS, this configuration does not work on z/OS

2. **Can I have this shared filesystem mounted at different places on my IBM Integration Bus and IBM Sterling Connect:Direct machines for platforms other than Windows?** Set the brokerPathToInputDir to be the location where it is mounted on the IBM Integration Bus machine, and the cdPathToInputDir to where it is mounted on the IBM Sterling Connect:Direct machine.
3. **Can I have this shared filesystem mounted at different places on my IBM Integration Bus and IBM Sterling Connect:Direct machines for Windows platforms?** You must use the UNC path syntax to the shared drive, rather than a mapped drive letter. Moreover, the user ID accessing the UNC path, that is the:

IBM Integration Bus user ID, or

IBM Sterling Connect:Direct user ID

- depending on which one is running on Windows, must have full (write) access to the file system, using the same password.

Set the `brokerPathToInputDir` to be the location where it is mounted on the IBM Integration Bus machine, and the `cdPathToInputDir` to where it is mounted on the IBM Sterling Connect:Direct machine.

Results

To create a configurable service, use the IBM Integration Explorer; see “Using the IBM Integration Explorer to work with configurable services” on page 345 for more information, or the `mqsicreateconfigurableservice` command.

```
mqsicreateconfigurableservice MYBROKER -c CDServer -o myCDServer
```

where

MYBROKER

is the name of the broker

CDServer

is the name of the configurable service

myCDServer

is the object name.

Note the `-o ObjectName` parameter can take any value.

Output

This example transfers a file using the `CDOutput` node called *CDOutput*, in a message flow called *messageflow1*, in integration server *default*, on broker *MYBROKER*.

The broker stages the file in the local file path `/tmp/cdtransfer/MYBROKER/default/messageflow1/CDOutput`.

The Connect:Direct server then tries to transfer the file using the file path `/cdserver/transfers/MYBROKER/default/messageflow1/CDOutput`.

These properties are used when the file systems are mounted differently on the broker and Connect:Direct server machines.

```
mqsichangeproperties MYBROKER -c CDServer  
-o myCDServer -n brokerPathToStagingDir,cdPathToStagingDir  
-v /tmp/cdtransfer,/cdserver/transfers
```

To pick up the updated values in the configurable service, restart the integration server and message flow.

Input

This example receives a file using the `CDInput` node called *CDInput*, in a message flow called *messageflow1*, in integration server *default*, on broker *MYBROKER*.

A file is transferred to the Connect:Direct server, to the directory `/cdserver/transfers/example1`.

IBM Integration Bus attempts to process the same file, but using the path `/tmp/cdtransfer/example1`.

These properties are used when the file systems are mounted differently on the IBM Integration Bus and Connect:Direct server machines.

```
mqsichangeproperties MYBROKER -c CDServer -o myCDServer  
-n brokerPathToInputDir,cdPathToInputDir  
-v /tmp/cdtransfer/example1,/cdserver/transfers/example1
```

To pick up the updated values in the configurable service, restart the integration server and message flow.

To display Connect:Direct server configurable services, use the

mqsireportproperties command, as shown in the following example:

```
mqsireportproperties MYBROKER -c CDServer -o AllReportableEntityNames -r
```

Related concepts:



Configurable services

Configurable services are typically runtime properties. You can use them to define properties that are related to external services on which the broker relies; for example, an SMTP server or a JMS provider.

Related tasks:



Initiating a managed file transfer using IBM Sterling Connect:Direct

Use a CDOutput node to send a file from a specified directory on your primary Connect:Direct server (PNODE) to a filename and directory on a secondary Connect:Direct server (SNODE).



Receiving a file using IBM Sterling Connect:Direct

Use the CDInput node to receive files from an IBM Sterling Connect:Direct network.

Related reference:



mqsicreateconfigurable service command

Use the **mqsicreateconfigurable service** command to create an object name for a broker external resource.



mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.



mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.



mqsdeleteconfigurable service command

Use the **mqsdeleteconfigurable service** command to delete a configurable service, such as a JMS provider, JDBC provider, or FTP server, that you have created by using the **mqsicreateconfigurable service** command.



CDServer configurable service

Select the objects and properties that you want to change for the CDServer configurable service.

Configuring for IMS:

You can use configurable services to enable the IMS nodes in the broker runtime environment to connect with the IMS system.

About this task

The following topics describe how to prepare the environment to connect to the IMS system, and how to change the connection details without the need to redeploy your message flow.

- “Preparing the environment for IMS nodes”
- “Changing connection information for the IMSRequest node” on page 432

Preparing the environment for IMS nodes:

Before you can use the IMS nodes, you must set up the broker runtime environment so that you can access the IMS system.

Before you begin

Before you start:

Read IBM Information Management System (IMS).

About this task

Complete the following steps to ensure that IBM Integration Bus can connect to the IMS system.

Procedure

1. Ensure that IMS Connect is installed and started on the IMS system.
2. If you do not want to configure IMS connection properties directly on the IMSRequest node, define a configurable service for each IMS system to which you want to connect.

For example, to create an IMSConnect configurable service for the IMS instance *IMSA* that is running on *test.ims.ibm.com*, port *9999*, run the

mqsicreateconfigurableservice command as shown:

```
mqsicreateconfigurableservice IB9NODE -c IMSConnect -o myIMSConnectService  
-n Hostname,PortNumber,DataStoreName -v test.ims.ibm.com,9999,IMSA
```

For details about how to create, change, and report configurable services, see “Changing connection information for the IMSRequest node” on page 432. You can use the IMSConnect configurable service to configure the IMSRequest node to use Secure Sockets Layer (SSL) protocol. For more information, see Securing the connection to IMS by using SSL.

3. Use the **mqsisetdbparms** command to set security details in the broker store.

For example, to associate a user ID and password pair with an IMS Connect connection, run the **mqsisetdbparms** command as shown:

```
mqsisetdbparms IB9NODE -n ims::mySecurityIdentity -u myuserid -p mypassword
```

Related concepts:



IBM Information Management System (IMS)

IMS is a message-based transaction manager and hierarchical-database manager for z/OS. External applications can use transactions to interact with applications that run inside IMS.

Related tasks:

“Changing connection information for the IMSRequest node”

You can create a configurable service that the IMSRequest node or message flow refers to at run time for connection information, instead of defining the connection properties on the node or the message flow. The advantage being that you can change the host name, performance, and security values without needing to redeploy your message flow.

Securing the connection to IMS by using SSL

Configure the IMSRequest node to communicate with IMS over the Secure Sockets Layer (SSL) protocol by creating a keystore file, and configuring the broker to use SSL.

Related reference:

IMSRequest node

Use the IMSRequest node to send a request to run a transaction on a local or remote IMS system, and wait for a response. IMS Connect must be configured and running on the IMS system.

mqsicreateconfigurable command

Use the **mqsicreateconfigurable** command to create an object name for a broker external resource.

mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.

mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.

mqsideleteconfigurable command

Use the **mqsideleteconfigurable** command to delete a configurable service, such as a JMS provider, JDBC provider, or FTP server, that you have created by using the **mqsicreateconfigurable** command.

mqsisetdbparms command

Use the **mqsisetdbparms** command to associate a specific user ID and password (or SSH identity file) with one or more resources that are accessed by the broker.

Changing connection information for the IMSRequest node:

You can create a configurable service that the IMSRequest node or message flow refers to at run time for connection information, instead of defining the connection properties on the node or the message flow. The advantage being that you can change the host name, performance, and security values without needing to redeploy your message flow.

Before you begin

Before you start:

- Read Configurable services to find out more about configurable services.
- Read IBM Information Management System (IMS) for background information.

About this task

Use the IMSConnect configurable service to change the connection information for the IMSRequest node. Two configurable services can connect to the same instance of IMS Connect. The properties of the IMSConnect configurable service are described in Configurable services properties.

You can use the IMSConnect configurable service to configure the IMSRequest node to use Secure Sockets Layer (SSL) protocol. For more information, see Securing the connection to IMS by using SSL.

Creating, changing, reporting, and deleting configurable services

Procedure

- To create a configurable service, use the IBM Integration Explorer; see “Using the IBM Integration Explorer to work with configurable services” on page 345 for more information, or the **mqsicreateconfigurableservice** command, as shown in the following example. This example creates an IMSConnect configurable service for the IMS instance IMSA that is running on test.ims.ibm.com port 9999:

```
mqsicreateconfigurableservice IB9NODE -c IMSConnect -o myIMSConnectService -n Hostname,PortNumber,DataStoreName -v test.ims.ibm.com,9999,IMSA
```
- To change a configurable service, use the IBM Integration Explorer, or the **mqsichangeproperties** command, as shown in the following example. You must stop and start the integration server for the change of property value to take effect. This example changes all the nodes that are configured to use the *myIMSConnectService* configurable service. After you run this command, the IMSRequest node connects to the production system (production.ims.ibm.com) instead of the test system (test.ims.ibm.com). The command also changes to coded character set identifier (CCSID) to 37.

```
mqsichangeproperties IB9NODE -c IMSConnect -o myIMSConnectService -n Hostname,CodedCharSetID -v production.ims.ibm.com,37
```

See Securing the connection to IMS by using SSL for information about how to turn on SSL support in the broker by setting the UseSSL and SSLEncryptionType IMSConnect configurable service properties.

- To display all IMSConnect configurable services, use the IBM Integration Explorer, or the **mqsireportproperties** command, as shown in the following example:
- You can delete a configurable service that you have created by using the IBM Integration Explorer, or the **mqsdeleteconfigurableservice** command, as shown in the following example:

```
mqsdeleteconfigurableservice IB9NODE -c IMSConnect -o myIMSConnectService
```

You can delete only custom-named configurable services; you cannot delete IBM-defined configurable services.

Related concepts:

 WebSphere Adapters nodes

A WebSphere Adapters node is a message flow node that is used to communicate with Enterprise Information Systems (EIS), such as SAP, Siebel, JD Edwards, and PeopleSoft.

Configurable services

Configurable services are typically runtime properties. You can use them to define properties that are related to external services on which the broker relies; for example, an SMTP server or a JMS provider.

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:

Securing the connection to IMS by using SSL

Configure the IMSRequest node to communicate with IMS over the Secure Sockets Layer (SSL) protocol by creating a keystore file, and configuring the broker to use SSL.

Resolving problems when you use IMS nodes

Advice for dealing with common problems that can arise when you develop message flows that contain IMS nodes.

Related reference:

Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

mqsicreateconfigurable command

Use the **mqsicreateconfigurable** command to create an object name for a broker external resource.

mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.

mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.

mqsdeleteconfigurable command

Use the **mqsdeleteconfigurable** command to delete a configurable service, such as a JMS provider, JDBC provider, or FTP server, that you have created by using the **mqsicreateconfigurable** command.

IMSRequest node

Use the IMSRequest node to send a request to run a transaction on a local or remote IMS system, and wait for a response. IMS Connect must be configured and running on the IMS system.

Defining where the CORBARquest node gets the object reference:

You can specify an object reference name either on the CORBARquest node or by using a configurable service.

Before you begin

Before you start:

- Read Common Object Request Broker Architecture (CORBA) and Configurable services for background information.

About this task

By using configurable services, you can specify the location from which the CORBARequest node gets the object reference without the need to redeploy the message flow. You can also use the configurable service to specify this location for multiple CORBARequest nodes. To pick up new values when a configurable service is created or modified, you must reload the broker or integration server to which the message flow was deployed, by using the **mqsistop** and **mqsistart** commands, or the **mqsireload** command.

The properties of the CORBA configurable services are described in Configurable services properties.

Creating, changing, reporting, and deleting configurable services

Procedure

- To create a configurable service, use the IBM Integration Explorer (as described in “Using the IBM Integration Explorer to work with configurable services” on page 345) or the **mqsicreateconfigurableservice** command shown in the following example.

This example creates a CORBA configurable service called "myCORBAService" that connects to the CORBA host on the local host on port 2809, and looks for the object reference called "Europe.region/Market.object". (For more information about how to specify the object reference, see CORBA naming service.)

```
mqsicreateconfigurableservice IB9NODE -c CORBA -o myCORBAService
-n namingService,objectReferenceName -v localhost:2809,Europe.region/Market.object
```

To pick up the new values in the configurable service, restart the integration server and message flow.

- To change a configurable service, use the IBM Integration Explorer or the **mqsichangeproperties** command shown in the following example.

This example changes the location of the object reference.

```
mqsichangeproperties IB9NODE -c CORBA -o myCORBAService -n namingService,objectReferenceName -v production.corba.ibm.com:2809,Europe.region/Market.object
```

To pick up the updated values in the configurable service, restart the integration server and message flow.

- To display all CORBA configurable services, use the IBM Integration Explorer or the **mqsireportproperties** command shown in the following example.

```
mqsireportproperties IB9NODE -c CORBA -o AllReportableEntityNames -r
```

- You can delete a configurable service that you have created by using the IBM Integration Explorer or the **mqsdeleteconfigurableservice** command shown in the following example.

```
mqsdeleteconfigurableservice IB9NODE -c CORBA -o myCORBAService
```

You can delete only custom-named configurable services; you cannot delete IBM-defined configurable services.

Related concepts:

Configurable services

Configurable services are typically runtime properties. You can use them to define properties that are related to external services on which the broker relies; for

example, an SMTP server or a JMS provider.

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:



Working with CORBA

Use CORBA nodes to connect to CORBA Internet Inter-Orb Protocol (IIOP) applications.

Related reference:



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.



mqscreateconfigurable command

Use the **mqscreateconfigurable** command to create an object name for a broker external resource.



mqschangeproperties command

Use the **mqschangeproperties** command to modify broker properties and properties of broker resources.



mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.



mqsdeleteconfigurable command

Use the **mqsdeleteconfigurable** command to delete a configurable service, such as a JMS provider, JDBC provider, or FTP server, that you have created by using the **mqscreateconfigurable** command.



CORBARequest node

Use the CORBARequest node to call an external CORBA application over Internet Inter-Orb Protocol (IIOP).

Configuring for CICS Transaction Server for z/OS:

You can use configurable services to enable the CICSRequest node in the broker runtime environment to connect with CICS Transaction Server for z/OS.

About this task

The following topics describe how to prepare the environment to connect to CICS, and how to change the connection details without the need to redeploy your message flow.

- “Preparing the environment for the CICSRequest node” on page 437
- “Changing connection information for the CICSRequest node” on page 439

Preparing the environment for the CICSRequest node:

Before you can use the CICSRequest node, you must configure IP InterCommunications (IPIC) protocol on the target CICS Transaction Server for z/OS.

Before you begin

Before you start:

Read CICS Transaction Server for z/OS connectivity for background information.

About this task

The CICSRequest node can send IPIC requests over TCP/IP to CICS Transaction Server for z/OS Version 3.2 and later. Complete the following steps on CICS to perform this configuration:

Procedure

1. Set the System Initialization (SIT) parameter **TCPIP=YES**.
2. Define the TCP/IP address and host name for CICS. By default, they are defined in the PROFILE.TCPIP and TCPIP.DATA data sets.
3. Add a TCP/IP listener to CICS by using the following **CEDA** command to define a TCPIPSERVICE resource in a group:

```
CEDA DEF TCPIPSERVICE(service-name) GROUP(group-name)
```

Ensure that the group in which you define the service is in the GRPLIST system initialization parameter, so that the listener starts when CICS is started. Key fields are explained as follows:

POrtnumber:

The port on which the TCP/IP service listens.

PROtocol:

The protocol of the service is IPIC.

TRansaction:

The transaction that CICS runs to handle incoming IPIC requests. Set the field to CISS, which is the default.

Backlog:

The Backlog field is the number of TCP/IP connection requests are sent to CICS, which are placed in a TCP/IP queue to be assigned an IPCONN connection to CICS. The default value is 1. Do not use a value of 0; a value of 0 indicates that no TCP/IP connection requests are to be assigned an IPCONN connection to CICS, which disables incoming connection requests.

Ipaddress:

The IP address, in dotted decimal form, on which the TCPIPSERVICE resource listens. For configurations with more than one IP stack, specify ANY to make the TCPIPSERVICE resource listen on all addresses.

SOcketclose:

Whether CICS waits before closing the socket after issuing a receive for incoming data on that socket. To ensure that the connection from the CICSRequest node always remains open, set SOcketclose to NO for IPIC connections.

SSI: Whether the CICS TCP/IP service is to use Secure Sockets Layer (SSL) protocol for encryption and authentication. Valid values are NO, YES, or CLIENTAUTH. Where:

- NO indicates that SSL is not to be used.
- YES indicates that the personal certificate of the CICS region must be trusted by IBM Integration Bus.
- CLIENTAUTH indicates that the personal certificate of the CICS region must be trusted by IBM Integration Bus, and the IBM Integration Bus personal certificate must be trusted by CICS.

The CICSRequest node does not support a separate truststore, so the keystore file must provide both personal and signer certificates. For more information, see *Securing the connection to CICS Transaction Server for z/OS by using SSL*.

4. Use the following command to install the TCPIPSERVICE definition:

```
CEDA INS TCPIPSERVICE(service-name) GROUP(group-name)
```

Related concepts:



CICS Transaction Server for z/OS overview

CICS Transaction Server for z/OS provides general-purpose transaction processing software for z/OS. CICS is a powerful application server that meets the transaction-processing needs of both large and small enterprises.

Related tasks:

“Changing connection information for the CICSRequest node” on page 439

You can create a configurable service that the CICSRequest node or message flow refers to at run time for connection information, instead of defining the connection properties on the node or the message flow. The advantage being that you can change the host name and performance values without needing to redeploy your message flow.

Related reference:



CICSRequest node

Use the CICSRequest node to call CICS Transaction Server for z/OS programs over TCP/IP-based IP InterCommunications (IPIC) protocol.



mqsicreateconfigurable command

Use the **mqsicreateconfigurable** command to create an object name for a broker external resource.



mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.



mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.



mqsdeleteconfigurable command

Use the **mqsdeleteconfigurable** command to delete a configurable service, such as a JMS provider, JDBC provider, or FTP server, that you have created by using the **mqsicreateconfigurable** command.



mqssetdbparms command

Use the **mqssetdbparms** command to associate a specific user ID and password (or SSH identity file) with one or more resources that are accessed by the broker.

Changing connection information for the CICSRequest node:

You can create a configurable service that the CICSRequest node or message flow refers to at run time for connection information, instead of defining the connection properties on the node or the message flow. The advantage being that you can change the host name and performance values without needing to redeploy your message flow.

Before you begin

Before you start:

- Read Configurable services to find out more about configurable services.
- Read CICS Transaction Server for z/OS overview for background information.

About this task

Use the CICSConnection configurable service to change the CICS Transaction Server for z/OS connection information for the CICSRequest node. The advantage being that you can change the host name, performance, and security identity values without needing to redeploy your message flow. The properties of the CICSConnection configurable service are described in Configurable services properties.

You can use the CICSConnection configurable service to configure the CICSRequest node to use Secure Sockets Layer (SSL) protocol. For more information, see Securing the connection to CICS Transaction Server for z/OS by using SSL.

Creating, changing, reporting, and deleting configurable services

Procedure

- To create a configurable service, use the IBM Integration Explorer; see “Using the IBM Integration Explorer to work with configurable services” on page 345 for more information, or the **mqscreateconfigurable** command, as shown in the following example. This example creates a CICSConnection configurable service for the CICS instance that is running at test.cics.ibm.com port 12345. The broker is identified by APPLID *BRKApp* and qualifier *BRKQual*. The connection timeout is 10 seconds, the request timeout is 5 seconds, and the security identity is identified by *mySecurityIdentity* in this example:

```
mqscreateconfigurable IB9NODE -c CICSConnection -o myCICSConnectionService  
-n cicsServer,clientApplid,clientQualifier,connectionTimeoutSecs,requestTimeoutSecs,  
securityIdentity  
-v tcp://test.cics.ibm.com:12345,BRKApp,BRKQual,10,5,mySecurityIdentity
```
- To change a configurable service, use the IBM Integration Explorer, or the **mqschangeproperties** command, as shown in the following example. You must stop and start the integration server for the change of property value to take effect. This example changes the CICSRequest node that is configured to use the *myCICSConnectionService* configurable service. After you run this command, the CICSRequest node connects to the production system (tcp://production.cics.ibm.com:12345) instead of the test system (tcp://test.cics.ibm.com:12345).

```
mqschangeproperties IB9NODE -c CICSConnection -o myCICSConnectionService  
-n cicsServer -v tcp://production.cics.ibm.com:12345
```


See Securing the connection to CICS Transaction Server for z/OS by using SSL for information about how to turn on SSL support in the broker by setting the `cicsServer CICSConnection` configurable service property.

- To display all CICSConnection configurable services, use the IBM Integration Explorer, or the **`mqsireportproperties`** command, as shown in the following example:

```
mqsireportproperties IB9NODE -c CICSConnection -o AllReportableEntityNames -r
```

- You can delete a configurable service that you have created by using the IBM Integration Explorer, or the **`mqsdeleteconfigurable-service`** command, as shown in the following example:

```
mqsdeleteconfigurable-service IB9NODE -c CICSConnection -o myCICSConnectionService
```

You can delete only custom-named configurable services; you cannot delete IBM-defined configurable services.

Related concepts:



CICS Transaction Server for z/OS connectivity

Use the CICSRequest node to connect IBM Integration Bus with CICS Transaction Server for z/OS applications.



Configurable services

Configurable services are typically runtime properties. You can use them to define properties that are related to external services on which the broker relies; for example, an SMTP server or a JMS provider.

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related reference:



CICSRequest node

Use the CICSRequest node to call CICS Transaction Server for z/OS programs over TCP/IP-based IP InterCommunications (IPIC) protocol.



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.



`mqscreateconfigurable-service` command

Use the **`mqscreateconfigurable-service`** command to create an object name for a broker external resource.



`mqsdeleteconfigurable-service` command

Use the **`mqsdeleteconfigurable-service`** command to delete a configurable service, such as a JMS provider, JDBC provider, or FTP server, that you have created by using the **`mqscreateconfigurable-service`** command.



`mqschange-properties` command

Use the **`mqschange-properties`** command to modify broker properties and properties of broker resources.



`mqsireport-properties` command

Use the **`mqsireport-properties`** command to display properties that relate to a broker, an integration server, or a configurable service.

Preparing the environment for WebSphere MQ File Transfer Edition nodes:

Prepare the file system and queue managers, and determine the name of the broker agent.

About this task

- “Preparing the file system”
- “Preparing the queue manager”
- “Setting the coordination queue manager”
- “Naming integration servers” on page 442
- “Determining the agent name” on page 443

Preparing the file system:

IBM Integration Bus uses a location in its work path to store transfers to remote agents. It uses another location as the default directory for received files. The high-level directory path for both locations is:

- *workpath/common/FTE*

Ensure that enough space is available here for files that you will transfer to and from the broker by WebSphere MQ File Transfer Edition.

Preparing the queue manager:

IBM Integration Bus tries to create all the required artifacts on the queue manager for the agent, and the coordination queue manager, if appropriate. It might not be possible to create all artifacts, due to the configuration of your machine, or permissions. If it fails to create them, or you want to create them yourself in advance, see “Scripts to create artifacts required for WebSphere MQ File Transfer Edition” on page 444.

Setting the coordination queue manager:

When a message flow that contains a WebSphere MQ File Transfer Edition node is deployed to an integration server, an agent is automatically created and started in that integration server. By default, the agent uses the broker's local queue manager as the coordination queue manager.

- If the broker's queue manager is being used as the coordination queue manager, the broker configures it as a coordination queue manager.
- If you are using a different queue manager as the coordination queue manager, refer to the WebSphere File Transfer Edition product documentation for details of how to configure it as a coordination queue manager.

Unless you have previously defined the coordination queue manager, the agent is temporary; it is deleted when the flow is undeployed or the broker is stopped. This behavior is acceptable in a test environment. However, for production, the administrator must specify the coordination queue manager for the integration server. Specifying a coordination queue manager:

Ensures that the correct queue manager is used when the agent is created.

Makes the agent permanent. If a coordination queue manager has been defined, the agent is deleted only after you undefine the coordination queue manager (for example, by setting it to an empty string), and restart the integration server.

A warning is written to the log if the coordination queue manager is not changed from the default.

The following state diagram illustrates how the presence of nodes and a defined coordination queue manager affect the state of the agent.

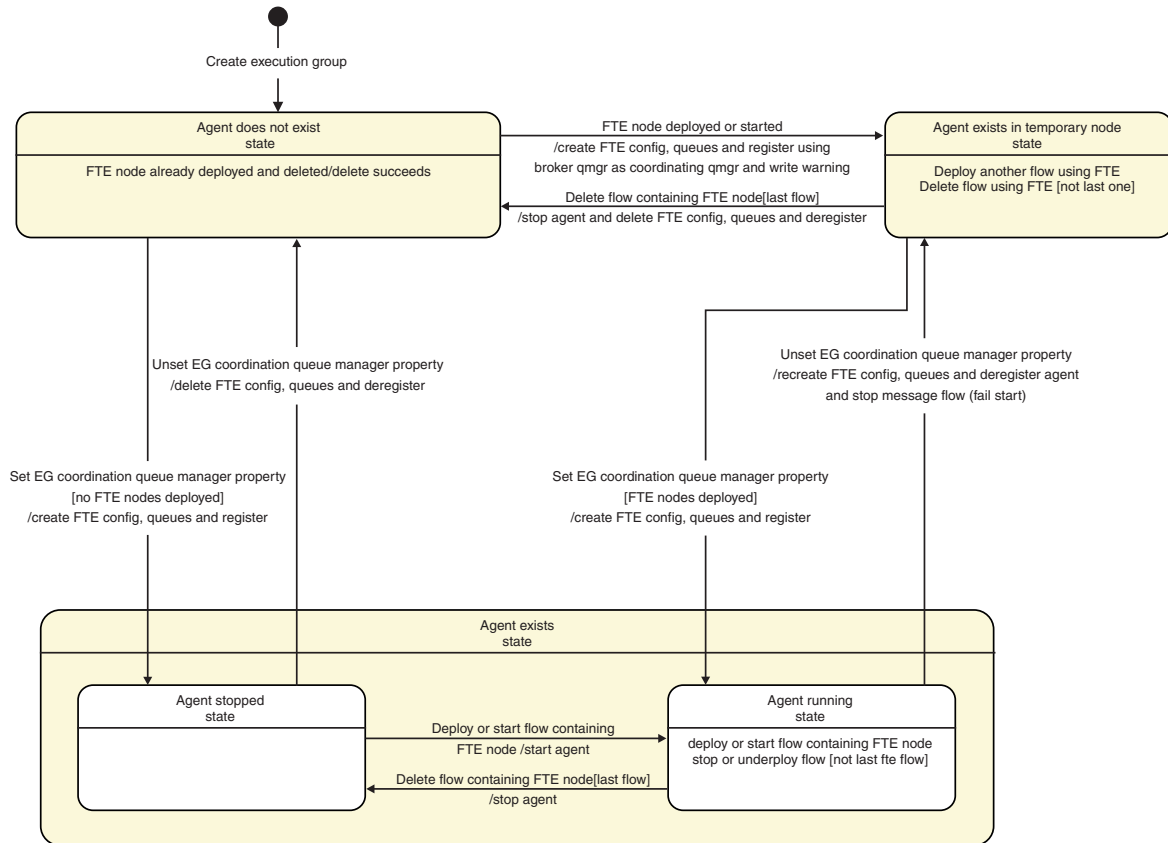


Figure 5. State machine diagram showing how the presence of nodes and a defined coordination queue manager affect the state of the agent. This diagram summarizes the information in the preceding paragraph.

About this task

Use one of the following methods to set the coordination queue manager.

Procedure

- Optional: Use the **mqsichangeproperties** command.
For example, to set the coordination queue manager to *QM2* for integration server *myExecutionGroup* in broker *IB9NODE*:

```
mqsichangeproperties IB9NODE -e myExecutionGroup
-o FTEAgent -n coordinationQMgr -v QM2
```

After running this command, you must reload the integration server for the change to take effect.

- Optional: Use the IBM Integration Explorer. Right-click the integration server, and select the appropriate options from the pop-up menu.

Naming integration servers:

About this task

The integration server name is used to form the queue name for WebSphere MQ File Transfer Edition queues. Consequently, the names of your integration servers must conform to the rules for naming WebSphere MQ objects. You cannot deploy a flow that contains a WebSphere MQ File Transfer Edition node unless this requirement is met. Permitted characters are:

- Uppercase A-Z
- Lowercase a-z (but there are restrictions on the use of lowercase letters for z/OS console support)

On systems using EBCDIC Katakana you cannot use lowercase characters.

- Numerics 0-9
- Period (.)
- Forward slash (/)
- Underscore (_)
- Percent sign (%)

See the WebSphere MQ Version 7 product documentation online for full details of naming requirements.

Determining the agent name:

About this task

To send a file to a given integration server, users need to know the name of the agent that the broker creates. The agent name is derived from *Broker.IntegrationServer*, and is not configurable. The total name length is limited to 28 characters, with a maximum 12 characters for the broker name, and 15 characters for the integration server. Broker and integration server names longer than these limits are truncated to form the agent name. The name must be a valid format for generating MQ Series queue name. Ensure that:

- The broker name is 12 characters or fewer (or at least unique in the first 12 characters).
- The integration server names are 15 characters or fewer (or at least unique in the first 15 characters).
- The broker and integration servers do not contain any characters that are invalid for queue names.
- The *broker.integrationserver* tuples are all unique, even if case is ignored.

The value used is written to the event log in message BIP3358. Use one of the following methods to determine the agent name.

Procedure

1. Optional: Use the **mqsireportproperties** command.

For example, to display the FTE agent name for integration server FTESAMPLE in broker IB9NODE:

```
mqsireportproperties IB9NODE -e FTESAMPLE -o FTEAgent -n agentName
```


If the agent has been created, the command returns the agent name. If the agent has not been created, the command returns an empty string.

2. Optional: Use the IBM Integration Explorer. Right-click the integration server, and select the appropriate options from the pop-up menu.


Related concepts:


 Managed file transfers using WebSphere MQ File Transfer Edition Transfer files, with file transfer metadata, in a timely and reliable manner.


Related tasks:


 Changing the location of the work path
The work path directory is the location where a component stores internal data, such as installation logs, component details, and trace output. The shared-classes directory is also located in the work path directory and is used for deployed Java code. If the work path directory does not have enough capacity, redirect the directory to another file system that has enough capacity.

Related reference:

 **mqsichangeproperties** command
Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.

 **mqsireportproperties** command
Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.

 FTEInput node
Use the FTEInput node to receive files using the WebSphere MQ File Transfer Edition.

 FTEOutput node
Use the FTEOutput node to write messages to files by using the WebSphere MQ File Transfer Edition.

Scripts to create artifacts required for WebSphere MQ File Transfer Edition:

WebSphere MQ script (MQSC) commands for use if IBM Integration Bus cannot create all the required artifacts, or you want to create them yourself.

Queues for the FTE agent

Before running the following command, replace *Agent name* with the name of your FTE agent; see “Determining the agent name” on page 443.

```
DEFINE QLOCAL(SYSTEM.FTE.COMMAND.Agent name) +
  DEFPRTY(0) +
  DEFSOPT(SHARED) +
  GET(ENABLED) +
  MAXDEPTH(5000) +
  MAXMSGL(4194304) +
  MSGDLVSQ(PRIORITY) +
  PUT(ENABLED) +
  RETINTVL(99999999) +
  SHARE +
  NOTRIGGER +
  USAGE(NORMAL) +
  REPLACE
DEFINE QLOCAL(SYSTEM.FTE.DATA.Agent name) +
  DEFPRTY(0) +
  DEFSOPT(SHARED) +
  GET(ENABLED) +
  MAXDEPTH(5000) +
  MAXMSGL(4194304) +
```

```

MSGDLVSQ(PRIORITY) +
PUT(ENABLED) +
RETINTVL(999999999) +
SHARE +
NOTRIGGER +
USAGE(NORMAL) +
REPLACE
DEFINE QLOCAL(SYSTEM.FTE.REPLY.Agent name) +
DEFPRTY(0) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(5000) +
MAXMSGL(4194304) +
MSGDLVSQ(PRIORITY) +
PUT(ENABLED) +
RETINTVL(999999999) +
SHARE +
NOTRIGGER +
USAGE(NORMAL) +
REPLACE
DEFINE QLOCAL(SYSTEM.FTE.STATE.Agent name) +
DEFPRTY(0) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(5000) +
MAXMSGL(4194304) +
MSGDLVSQ(PRIORITY) +
PUT(ENABLED) +
RETINTVL(999999999) +
SHARE +
NOTRIGGER +
USAGE(NORMAL) +
REPLACE
DEFINE QLOCAL(SYSTEM.FTE.EVENT.Agent name) +
DEFPRTY(0) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(5000) +
MAXMSGL(4194304) +
MSGDLVSQ(PRIORITY) +
PUT(ENABLED) +
RETINTVL(999999999) +
SHARE +
NOTRIGGER +
USAGE(NORMAL) +
REPLACE
DEFINE QLOCAL(SYSTEM.FTE.AUTHAGT1.Agent name) +
DEFPRTY(0) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(0) +
MAXMSGL(0) +
MSGDLVSQ(PRIORITY) +
PUT(ENABLED) +
RETINTVL(999999999) +
SHARE +
NOTRIGGER +
USAGE(NORMAL) +
REPLACE
DEFINE QLOCAL(SYSTEM.FTE.AUTHTRN1.Agent name) +

```

```

DEFPRTY(0) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(0) +
MAXMSGL(0) +
MSGDLVSQ(PRIORITY) +
PUT(ENABLED) +
RETINTVL(999999999) +
SHARE +
NOTRIGGER +
USAGE(NORMAL) +
REPLACE
DEFINE QLOCAL(SYSTEM.FTE.AUTHOPS1.Agent name) +
DEFPRTY(0) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(0) +
MAXMSGL(0) +
MSGDLVSQ(PRIORITY) +
PUT(ENABLED) +
RETINTVL(999999999) +
SHARE +
NOTRIGGER +
USAGE(NORMAL) +
REPLACE
DEFINE QLOCAL(SYSTEM.FTE.AUTHSCH1.Agent name) +
DEFPRTY(0) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(0) +
MAXMSGL(0) +
MSGDLVSQ(PRIORITY) +
PUT(ENABLED) +
RETINTVL(999999999) +
SHARE +
NOTRIGGER +
USAGE(NORMAL) +
REPLACE
DEFINE QLOCAL(SYSTEM.FTE.AUTHMON1.Agent name) +
DEFPRTY(0) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(0) +
MAXMSGL(0) +
MSGDLVSQ(PRIORITY) +
PUT(ENABLED) +
RETINTVL(999999999) +
SHARE +
NOTRIGGER +
USAGE(NORMAL) +
REPLACE
DEFINE QLOCAL(SYSTEM.FTE.AUTHADM1.Agent name) +
DEFPRTY(0) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(0) +
MAXMSGL(0) +
MSGDLVSQ(PRIORITY) +
PUT(ENABLED) +
RETINTVL(999999999) +

```



```
SHARE +
NOTRIGGER +
USAGE(NORMAL) +
REPLACE
```

Queues for the coordination queue manager

```
DEFINE TOPIC('SYSTEM.FTE') TOPICSTR('SYSTEM.FTE') REPLACE
ALTER TOPIC('SYSTEM.FTE') NPMSGDLV(ALLAVAIL) PMSGDLV(ALLAVAIL)
DEFINE QLOCAL(SYSTEM.FTE) LIKE(SYSTEM.BROKER.DEFAULT.STREAM) REPLACE
ALTER QLOCAL(SYSTEM.FTE) DESCR('Stream for WMQFTE Pub/Sub interface')
* Altering namelist: SYSTEM.QPUBSUB.QUEUE.NAMELIST
* Value prior to alteration:
DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)
ALTER NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST) +
  NAMES(SYSTEM.BROKER.DEFAULT.STREAM+
    ,SYSTEM.BROKER.ADMIN.STREAM,SYSTEM.FTE)
* Altering PSMODE. Value prior to alteration:
DISPLAY QMGR PSMODE
ALTER QMGR PSMODE(ENABLED)
DEFINE QLOCAL(SYSTEM.FTE.DATABASELOGGER.REJECT) +
  DESCR('Messages rejected by the FTE database logger.') +
  DEFPRTY(0) +
  DEFSOPT(SHARED) +
  GET(ENABLED) +
  MAXDEPTH(999999999) +
  MAXMSGL(4194304) +
  MSGDLVSQ(PRIORITY) +
  PUT(ENABLED) +
  RETINTVL(999999999) +
  SHARE +
  NOTRIGGER +
  USAGE(NORMAL) +
  REPLACE
DEFINE QLOCAL(SYSTEM.FTE.DATABASELOGGER.COMMAND) +
  DESCR('Command messages to control the FTE database logger.') +
  DEFPRTY(0) +
  DEFSOPT(SHARED) +
  GET(ENABLED) +
  MAXDEPTH(999999999) +
  MAXMSGL(4194304) +
  MSGDLVSQ(PRIORITY) +
  PUT(ENABLED) +
  RETINTVL(5000) +
  SHARE +
  NOTRIGGER +
  USAGE(NORMAL) +
  REPLACE
```

Related tasks:

“Preparing the environment for WebSphere MQ File Transfer Edition nodes” on page 441
Prepare the file system and queue managers, and determine the name of the broker agent.

Configuring for JMS:

You can use configurable services to configure IBM Integration Bus for JMS.

About this task

- “Configuring the broker to enable a JMS provider's proprietary API”
- “Connecting to different versions of the same JMS provider” on page 450
- “Configuring the JMSInput node for batch message processing” on page 451

Configuring the broker to enable a JMS provider's proprietary API:

Some JMS providers provide an alternative interface to the standard JMS specification for particular JMS API calls. In these cases, IBM supplies a Java class to interface with that proprietary API.

About this task

For example, BEA WebLogic uses a component called a *Client Interposed Transaction Manager* to allow a JMS client to obtain a reference to the XAResource that is associated with a user transaction.

If the WebSphere IBM Integration JMS nodes use BEA WebLogic as the JMS provider, and the nodes must participate in a globally coordinated message flow, you must modify the configurable services properties that are associated with that vendor. The following table shows the properties that have been added to the configurable service for BEA WebLogic.

JMS provider	Property	Purpose	Default value
BEA_WebLogic	proprietaryAPIHandler	The name of the IBM supplied Java class to interface with a JMS provider's proprietary API.	com.ibm.broker.apihandler. BEAWebLogicAPIHandler
	proprietaryAPIAttr1	The Initial Context Factory class name for the vendor	weblogic.jndi. WLInitialContextFactory
	proprietaryAPIAttr2	The URL of the WebLogic bindings	<i>URL JNDI bindings</i>
	proprietaryAPIAttr3	The DNS name of the JMS server	<i>Server name</i>

In the list of JMS provider configurable services, the name of the IBM supplied Java class is set to the default value for the `proprietaryAPIHandler` property. Typically, you do not need to change this value, unless you are instructed to do so by an IBM Service team representative.

Procedure

- Use the `mqsichangeproperties` command to modify values of the properties for this JMS provider.

The following example shows how to change the values of the properties `proprietaryAPIAttr2` and `proprietaryAPIAttr3` for the JMS provider configurable service definition called `BEA_Weblogic`, where these properties represent the URL of the WebLogic bindings and the DNS Server name of the BEA WebLogic JMS Server:

```
mqsichangeproperties IB9NODE -c JMSProvers -o BEA_Weblogic  
-n proprietaryAPIAttr2,proprietaryAPIAttr3 -v t3://9.20.94.16:7001,BEAServerName
```

You must stop and start the integration server for the changes of property values to take effect.

- Use the `mqsireportproperties` command to display the properties for a JMS provider.

The following example shows how to display the properties for all the broker's JMS provider resources (the default JMS provider resources and those configurable services that are defined with the **mqsicreateconfigurableservice** command):

```
mqsireportproperties IB9NODE -c JMSProviders -o BEA_WebLogic -r
```

The result of this command has the following format:

```
ReportableEntityName=''  
JMSProviders  
  BEA_WebLogic=''  
    jarsURL='default_Path'  
    nativeLibs='default_Path'  
    proprietaryAPIAttr1='weblogic.jndi.WLInitialContextFactory'  
    proprietaryAPIAttr2='t3://9.20.94.16:7001'  
    proprietaryAPIAttr3='BEAServerName'  
    proprietaryAPIAttr4='default_none'  
    proprietaryAPIAttr5='default_none'  
    proprietaryAPIHandler='com.ibm.broker.apihandler.BEAWebLogicAPIHandler'
```

The default location for the JMS provider JAR files is the broker's shared-classes directory. You can specify an alternative location for the JAR files by using the **mqsichangeproperties** command, as shown in the following example:

```
mqsichangeproperties IB9NODE -c JMSProviders -o BEA_WebLogic -n jarsURL  
-v /var/mqsi/WebLogic
```

On Windows, the file location cannot be a mapped network drive on a remote Windows computer; the directory must be local or on a Storage Area Network (SAN) disk.

- Use the **mqsicreateconfigurableservice** command to add a JMS provider. The following example shows how to add a JMS provider called BEAV91 for broker IB9NODE, specifying the name of an IBM supplied Java class called `com.ibm.broker.apihandler.BEAWebLogicAPIHandler` to handle vendor-specific API calls:

```
mqsicreateconfigurableservice IB9NODE -c JMSProviders -o BEAV91  
-n proprietaryAPIHandler,proprietaryAPIAttr1,proprietaryAPIAttr2,proprietaryAPIAttr3  
-v com.ibm.broker.apihandler.BEAWebLogicAPIHandler,weblogic.jndi.WLInitialContextFactory,  
t3://9.20.94.16:7001,BEAServerName
```

- If you have defined a user-defined JMS provider configurable service, set the value for the `proprietaryAPIHandler` property manually.

Related concepts:

Configurable services

Configurable services are typically runtime properties. You can use them to define properties that are related to external services on which the broker relies; for example, an SMTP server or a JMS provider.

Related reference:

Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

mqsicreateconfigurableservice command

Use the **mqsicreateconfigurableservice** command to create an object name for a broker external resource.

mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and

properties of broker resources.



mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.



JMSInput node

Use the JMSInput node to receive messages from JMS destinations. JMS destinations are accessed through a connection to a JMS provider.



JMSOutput node

Use the JMSOutput node to send messages to JMS destinations.



SOAPInput node

Use the SOAPInput node to process client SOAP messages, so that the broker operates as a SOAP Web Services provider.



SOAPRequest node

Use the SOAPRequest node to send a SOAP request to the remote Web service.



SOAPAsyncRequest node

Use the SOAPAsyncRequest node with the SOAPAsyncResponse node to construct a pair of message flows that call a Web service asynchronously.

Connecting to different versions of the same JMS provider:

To use different versions of the same JMS provider, create a configurable service for the JMS provider, and set the **jarsURL** property to a unique path.

About this task

To connect to two versions of the same JMS provider (for example, JBoss), complete the following steps.

Procedure

1. Create a separate JMSProviders configurable service for each version of the JMS provider.
For more information, see **mqsicreateconfigurable-service** command or “Creating a new configurable service” on page 347.
2. Set the **jarsURL** property of the JMSProviders configurable service to a unique path.
For a description of the properties of the JMSProviders configurable service, see JMSProviders configurable service.

Related concepts:



Connection to different JMS providers

The JMSInput, JMSReceive and JMSOutput nodes are compatible with all JMS providers that conform to the Java Message Service Specification, version 1.1.

Related tasks:

“Creating a new configurable service” on page 347

Use the IBM Integration Explorer to create a new configurable service to define properties for an external service on which the integration node (broker) relies.



Working with JMS

Learn about the concepts and tasks involved in configuring message flows to

support JMS messages.

Related reference:

 **mqsicreateconfigurable-service** command

Use the **mqsicreateconfigurable-service** command to create an object name for a broker external resource.

 Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

 Troubleshooting JMS nodes

Review possible problems with nodes using JMS transport.

 JMSInput node

Use the JMSInput node to receive messages from JMS destinations. JMS destinations are accessed through a connection to a JMS provider.

 JMSOutput node

Use the JMSOutput node to send messages to JMS destinations.

Configuring the JMSInput node for batch message processing:

Configure JMS message flows to send a batch acknowledgment for receipt of non-transactional JMS messages.

About this task

When the JMSInput node works in non-transactional mode, receipt and acknowledgment of messages take place in one step, followed by the message processing. In some scenarios, this acknowledgment response to the JMS server for each message might create an unacceptable level of network traffic. For example, using this messaging model to receive JMS messages across a wide area network that is already handling large volumes of traffic might result in non-optimal throughput rates for JMS messages.

The JMSInput node can acknowledge message receipt in batches rather than individually for non-transactional messages. Batch acknowledgment is enabled using the JMSProviders configurable service properties **clientAckBatchSize** and **clientAckBatchTime**. You can set these properties separately, or use them together, to tune the number of messages that are received and processed by the node before an acknowledgment response is returned to the source JMS server.

clientAckBatchSize

This is an integer value that represents the threshold number of messages received before the batch acknowledgment is sent.

clientAckBatchTime

This is an integer value that represents the length, in milliseconds, of a repeating interval. At the end of each interval a batch acknowledgment is sent for all unacknowledged non-transactional JMS messages that were received during the preceding interval.

A batch acknowledgment is also sent when:

- There are no more input messages on the JMS server

- An error occurs during message processing. In this case, all previous messages in the batch that were successfully processed are first acknowledged, before handling the error.
- The message flow stops.

To disable batch acknowledgment, set both **clientAckBatchSize** and **clientAckBatchTime** to 0.

Related concepts:



Configurable services

Configurable services are typically runtime properties. You can use them to define properties that are related to external services on which the broker relies; for example, an SMTP server or a JMS provider.

Related reference:



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.



mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.



mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.



JMSInput node

Use the JMSInput node to receive messages from JMS destinations. JMS destinations are accessed through a connection to a JMS provider.

Importing a policy set and policy set binding:

Use the **mqsichangeproperties** command to import a policy set and associated binding.

About this task

This topic shows how to import policy set *myPolicySet* to broker *myBroker* from a file called *myPolicySet.xml*. The associated binding is *myPolicySetBinding*, which you import from *myPolicySetBinding.xml*.

Procedure

1. Create a configurable service for the policy set, if one does not already exist.


```
mqsicreateconfigurableservice myBroker -c PolicySets -o myPolicySet
```

 BIP8071I: Successful command completion.
2. Create a configurable service for the policy set binding, if one does not already exist.


```
mqsicreateconfigurableservice myBroker -c PolicySetBindings -o myPolicySetBinding
```

 BIP8071I: Successful command completion.
3. Import the policy set.


```
mqsichangeproperties myBroker -c PolicySets -o myPolicySet -n ws-security -p myPolicySet.xml
```
4. Import the policy set binding.

```
mqschangeproperties myBroker -c PolicySetBindings -o myPolicySetBinding
-n ws-security -p myPolicySetBinding.xml
```

5. Change the value of the associatedPolicySet attribute. Set it to the name of the policy set with which this policy set binding was originally associated.

```
mqschangeproperties myBroker -c PolicySetBindings -o myPolicySetBinding
-n associatedPolicySet -v myPolicySet
```

6. You must stop and start the integration server for the changes of property values to take effect.

Related concepts:

“Policy sets” on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

Related tasks:

“Exporting a policy set and policy binding” on page 503

Use the **mqsireportproperties** command to export a policy set and associated binding to a file.

Related reference:

 **mqschangeproperties** command

Use the **mqschangeproperties** command to modify broker properties and properties of broker resources.

 **mqscreateconfigurable-service** command

Use the **mqscreateconfigurable-service** command to create an object name for a broker external resource.

Configuring for email:

You can use configurable services to enable the EmailOutput node and EmailInput node in the broker runtime environment to connect with email servers.

About this task

The following topics describe how to prepare the environment to connect to email servers, and how to change the connection details without the need to redeploy your message flow.

- Changing connection information for the EmailOutput node
- Changing connection information for the EmailInput node

Configuring properties for TCP/IP:

If you are deploying message flows that contain TCP/IP Nodes, and those nodes use a configurable service, set up your TCPIPClient/Server configurable services before the message flows are deployed.

Before you begin

Before you start:

- For background information about the TCP/IP transport, see TCP/IP data transfer and Connection management.
- Consider using the default configurable service instead of creating your own. For more information, see TCPIPClientInput node.

Procedure

- To create a TCPIPClient configurable service, use the IBM Integration Explorer; see “Using the IBM Integration Explorer to work with configurable services” on page 345 for more information, or the **mqsicreateconfigurableservice** command, as shown in the following example. There is no need to stop and start the integration server for changes to a TCPIPClient configurable service to take effect. This example creates a TCPIPClient configurable service named *myTCPIPClientConfigurableServiceName* that connects to the TCP/IP host *localhost*, and uses port *12345* for the connections into that server:

```
mqsicreateconfigurableservice myBroker -c TCPIPClient -o myTCPIPClientConfigurableServiceName -n Hostname,Port -v localhost,12345
BIP80711: Successful command completion.
```

- To change the properties of a TCPIPClient configurable service, use the IBM Integration Explorer, or the **mqsichangeproperties** command, as shown in the following example. There is no need to stop and start the integration server for changes to a TCPIPClient configurable service to take effect. This example changes the connections that are used by the TCPIPClient configurable service *myTCPIPClientConfigurableServiceName*.

```
mqsichangeproperties myBroker -c TCPIPClient -o myTCPIPClientConfigurableServiceName -n Hostname,AlternativePort -v newhost,otherhost:45678
BIP80711: Successful command completion.
```

- To create a TCPIPServer configurable service, use the IBM Integration Explorer; see “Using the IBM Integration Explorer to work with configurable services” on page 345 for more information, or the **mqsicreateconfigurableservice** command, as shown in the following example. You must stop and start the integration server for changes to a TCPIPServer configurable service to take effect. This example creates a TCPIPServer configurable service named *myTCPIPServerConfigurableServiceName* that listens on port *12345*, and has a maximum of *1000* connections into that server:

```
mqsicreateconfigurableservice myBroker -c TCPIPServer -o myTCPIPServerConfigurableServiceName -n Port,MaximumConnections -v 12345,1000
BIP80711: Successful command completion.
```

- To change the properties of a TCPIPServer configurable service, use the IBM Integration Explorer, or the **mqsichangeproperties** command, as shown in the following example. You must stop and start the integration server for changes to a TCPIPServer configurable service to take effect. This example changes the connections that are used by the TCPIPServer configurable service *myTCPIPServerConfigurableServiceName*.

```
mqsichangeproperties myBroker -c TCPIPServer -o myTCPIPServerConfigurableServiceName -n Port,ExpireConnectionSec -v 54321,60
BIP80711: Successful command completion.
```

- To configure a TCPIPClient socket to make 100 connections at deployment or startup time, use the **mqsicreateconfigurableservice** command. By default, the client connections are not made until they are required by one of the TCP/IP nodes. There is no need to stop and start the integration server for changes to a TCPIPClient configurable service to take effect. In this example, the TCPIPClient configurable service called *Task10* is configured to run on port *14150*, and 100 connections are created.

```
mqsicreateconfigurableservice IB9NODE -c TCPIPClient -o Task10
-n Port,MinimumConnections -v 14150,100
```

- To configure a TCPIPServer socket so that connections expire after a specified time, use the **mqsicreateconfigurableservice** command. The TCP/IP node can specify either the port to be used or the name of the configurable service. You must stop and start the integration server for changes to a TCPIPServer configurable service to take effect. In this example, the TCPIPServer configurable service called *Task9* is configured to run on port *14149*. The connections expire when they have not been used for 5 seconds.

```
mqsicreateconfigurableservice BRK6 -c TCPIPServer -o Task9
-n Port,ExpireConnectionSec -v 14149,5
```

Related concepts:



Connection management

TCP/IP connections are requested by the client connection manager and accepted by the server connection manager.



TCP/IP data transfer

You can use IBM Integration Bus to connect to applications that use raw TCP/IP sockets for transferring data.

Related reference:



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.



`mqsicreateconfigurable-service` command

Use the `mqsicreateconfigurable-service` command to create an object name for a broker external resource.



`mqsichangeproperties` command

Use the `mqsichangeproperties` command to modify broker properties and properties of broker resources.



TCPIPClientInput node

Use the TCPIPClientInput node to create a client connection to a raw TCP/IP socket, and to receive data over that connection.



TCPIPServerOutput node

Use the TCPIPServerOutput node to create a server connection to a raw TCP/IP socket, and to send data over the connection to an external application.



TCPIPServerInput node

Use the TCPIPServerInput node to create a server connection to a raw TCPIP socket, and to receive data over that connection.



TCPIPClientReceive node

Use the TCPIPClientReceive node to receive data over a client TCP/IP connection.



TCPIPServerReceive node

Use the TCPIPServerReceive node to receive data over a server TCP/IP connection.

Configuring internal resources required by flows

You can use configurable services to configure the internal resources that are required by some message flows.

About this task

The following topics describe how to configure the storage of events for certain message flow nodes.

- “Configuring the storage of events for aggregation nodes” on page 456
- “Configuring the storage of events for Collector nodes” on page 458
- “Configuring the storage of events for Resequencing nodes” on page 460
- “Configuring the storage of events for timeout nodes” on page 463
- “Configuring the XPath cache” on page 465
- “Configuring monitoring event sources using a monitoring profile” on page 466

Configuring the storage of events for aggregation nodes:

You can use an Aggregation configurable service to control the storage of events for AggregateControl and AggregateReply nodes.

About this task

By default, the storage queues used by all aggregation nodes are:

- SYSTEM.BROKER.AGGR.CONTROL
- SYSTEM.BROKER.AGGR.REPLY
- SYSTEM.BROKER.AGGR.REQUEST
- SYSTEM.BROKER.AGGR.UNKNOWN
- SYSTEM.BROKER.AGGR.TIMEOUT

However, you can control the queues that are used by different aggregation nodes by creating alternative queues containing a *QueuePrefix*, and using an Aggregation configurable service to specify the names of those queues for storing events.

Follow these steps to specify the queues that are used to store event states, and to set the expiry time of an aggregation:

Procedure

1. Create the storage queues to be used by the aggregation nodes. The following queues are required:
 - SYSTEM.BROKER.AGGR.*QueuePrefix*.CONTROL
 - SYSTEM.BROKER.AGGR.*QueuePrefix*.REPLY
 - SYSTEM.BROKER.AGGR.*QueuePrefix*.REQUEST
 - SYSTEM.BROKER.AGGR.*QueuePrefix*.UNKNOWN
 - SYSTEM.BROKER.AGGR.*QueuePrefix*.TIMEOUT

The *QueuePrefix* variable can contain any characters that are valid in a WebSphere MQ queue name, but must be no longer than eight characters and must not begin or end with a period (.). For example, SET1 and SET.1 are valid queue prefixes, but .SET1 and SET1. are invalid.

If you do not create the storage queues, IBM Integration Bus creates the set of queues when the node is deployed; these queues are based on the default queues. If the queues cannot be created, the message flow is not deployed.

2. Use the **mqsicreateconfigurable** command to create an Aggregation configurable service. You can create a configurable service to be used with either a specific aggregation or with all aggregations in an integration server.
 - a. If the configurable service is to be used with a specific aggregation, ensure that the name of the configurable service is the same as the name that you specify in the Aggregate name property on the AggregateControl and AggregateReply nodes. If the configurable service is to be used with all aggregations in an integration server, create the configurable service with the same name as the integration server.
 - b. Set the **Queue prefix** property to the required value.
 - c. Optional: Set the **Timeout** property to control the expiry time of an aggregation.

For example, create a configurable service called `myAggregation`, which specifies queues prefixed with `SYSTEM.BROKER.AGGR.SET1` and a timeout of 60 seconds:

```
mqscreateconfigurableservice MYBROKER -c Aggregation -o myAggregation  
-n queuePrefix,timeoutSeconds -v SET1,60
```

You can use the **mqsdeleteconfigurable**service command to delete the Aggregation configurable service. However, the storage queues are not deleted automatically when the configurable service is deleted, so you must delete them separately. For more information, see Configurable services properties

3. In the AggregateControl and AggregateReply nodes:
 - a. Ensure that the name of the Aggregation configurable service is the same as the name specified in the Aggregate name property on the **Basic** tab; for example, myAggregation. If no Aggregation configurable service exists with the same name as the Aggregate name property, and if a configurable service exists with the same name as the integration server, that configurable service is used instead.
 - b. Optional: Use the **mqschange**properties and **mqsireport**properties commands to change or view the properties of the configurable service. Alternatively, you can use the IBM Integration Explorer to view or modify a configurable service. For more information about working with configurable services, see “Using the IBM Integration Explorer to work with configurable services” on page 345.

What to do next

The properties for the configurable service are not used by the broker until you restart or redeploy the message flow, or restart the broker.

Related concepts:

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:

“Viewing configurable services” on page 348

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view the properties of configurable services defined on your integration node.

“Modifying an IBM defined configurable service” on page 349

You can create a new configurable service for an external service on which the broker relies, by modifying predefined configurable services provided by IBM. Use the IBM Integration Explorer to view and modify existing configurable services.

“Modifying a configurable service” on page 350

Use the IBM Integration Explorer to view and modify existing configurable services.

“Deleting a configurable service” on page 353

Use the IBM Integration Explorer to delete custom configurable services.

Related reference:



AggregateControl node

Use the AggregateControl node to mark the beginning of a fan-out of requests that are part of an aggregation.

AggregateRequest node

Use the AggregateRequest node to record the fact that request messages have been sent. This node also collects information that helps the AggregateReply node to construct the compound response message.

AggregateReply node

Use the AggregateReply node to mark the end of an aggregation fan-in. This node collects replies and combines them into a single compound message.

Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

`mqscreateconfigurable-service` command

Use the `mqscreateconfigurable-service` command to create an object name for a broker external resource.

`mqschange-properties` command

Use the `mqschange-properties` command to modify broker properties and properties of broker resources.

`mqsreport-properties` command

Use the `mqsreport-properties` command to display properties that relate to a broker, an integration server, or a configurable service.

Configuring the storage of events for Collector nodes:

You can use a Collector configurable service to control the storage of events for Collector nodes.

About this task

By default, the storage queues used by all Collector nodes are:

- SYSTEM.BROKER.EDA.EVENTS
- SYSTEM.BROKER.EDA.COLLECTIONS

These queues are also used by the Resequencer node.

However, you can control the queues that are used by different Collector nodes by creating alternative queues that contain a *QueuePrefix* variable, and by using a Collector configurable service to specify the names of those queues for storing events.

Follow these steps to specify the queues that are used to store event states, and to set the expiry for the collection:

Procedure

1. Create the storage queues to be used by the Collector node. The following queues are required:
 - SYSTEM.BROKER.EDA.*QueuePrefix*.EVENTS
 - SYSTEM.BROKER.EDA.*QueuePrefix*.COLLECTIONS

The *QueuePrefix* variable can contain any characters that are valid in a WebSphere MQ queue name, but must be no longer than eight characters and

must not begin or end with a period (.). For example, SET1 and SET.1 are valid queue prefixes, but .SET1 and SET1. are invalid.

If you do not create the storage queues, IBM Integration Bus creates the set of queues when the node is deployed; these queues are based on the default queues. If the queues cannot be created, the message flow is not deployed.

2. Use the **mqsicreateconfigurable-service** command to create a Collector configurable service. You can create a configurable service to be used with either a specific collection or with all collections in an integration server.
 - a. If you are creating a configurable service to be used with a specific collection, ensure that the name of the configurable service is the same as the name that you specify in the Configurable service property on the Collector node. If you are creating a configurable service to be used with all collections in the integration server, ensure that the configurable service has the same name as the integration server.
 - b. Set the **Queue prefix** property to the required value.
 - c. Optional: Set the **Collection expiry** property.

For example, create a Collector configurable service called myCollectorService, which uses queues prefixed with SYSTEM.BROKER.EDA.SET1, and with a collection expiry of 60 seconds:

```
mqsicreateconfigurable-service MYBROKER -c Collector -o myCollectorService  
-n queuePrefix,collectionExpirySeconds -v SET1,60
```

You can use the **mqsdeleteconfigurable-service** command to delete the Collector configurable service. However, the storage queues are not deleted automatically when the configurable service is deleted, so you must delete them separately.

For more information, see Configurable services properties

3. In the Collector node:
 - a. If the configurable service is to be used for a specific collection, specify the name of the configurable service in the Configurable service property on the **Advanced** tab; for example, myCollectorService. If you do not set the Configurable service property, and if a configurable service exists with the same name as the integration server, that configurable service is used instead.
 - b. Optional: Use the **mqschange-properties** and **mqsireport-properties** commands to change or view the properties of the configurable service. Alternatively, you can use the IBM Integration Explorer to view or modify a configurable service. For more information about working with configurable services, see “Using the IBM Integration Explorer to work with configurable services” on page 345.

What to do next

The properties for the configurable service are not used by the broker until you restart or redeploy the message flow, or restart the broker.

Related concepts:

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:

“Viewing configurable services” on page 348

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view the properties of configurable services defined on your integration node.

“Modifying an IBM defined configurable service” on page 349

You can create a new configurable service for an external service on which the broker relies, by modifying predefined configurable services provided by IBM. Use the IBM Integration Explorer to view and modify existing configurable services.

“Modifying a configurable service” on page 350

Use the IBM Integration Explorer to view and modify existing configurable services.

“Deleting a configurable service” on page 353

Use the IBM Integration Explorer to delete custom configurable services.

Related reference:



Collector node

Use the Collector node to create message collections based on rules that you configure in the node.



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.



mqsicreateconfigurable-service command

Use the **mqsicreateconfigurable-service** command to create an object name for a broker external resource.



mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.



mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.

Configuring the storage of events for Resequencing nodes:

You can use a Resequencing configurable service to control the storage of events for Resequencing nodes.

About this task

By default, the storage queues used by all Resequencing nodes are:

- SYSTEM.BROKER.EDA.EVENTS
- SYSTEM.BROKER.EDA.COLLECTIONS

These queues are also used by the Collector node.

However, you can control the queues that are used by different Resequencing nodes by creating alternative queues that contain a *QueuePrefix* variable, and by using a Resequencing configurable service to specify the names of those queues for storing events.

Follow these steps to specify the queues that are used to store event states, and to set the timeout and the start and end of the sequence:

Procedure

1. Create the storage queues to be used by the Resequencing node. The following queues are required:

- SYSTEM.BROKER.EDA.QueuePrefix.EVENTS
- SYSTEM.BROKER.EDA.QueuePrefix.COLLECTIONS

The *QueuePrefix* variable can contain any characters that are valid in a WebSphere MQ queue name, but must be no longer than eight characters and must not begin or end with a period (.). For example, SET1 and SET.1 are valid queue prefixes, but .SET1 and SET1. are invalid.

If you do not create the storage queues, IBM Integration Bus creates the set of queues when the node is deployed; these queues are based on the default queues. If the queues cannot be created, the message flow is not deployed.

2. Use the **mqsicreateconfigurable** command to create a Resequencing configurable service. You can create a configurable service to be used with either a specific sequence or with all sequences in an integration server.
 - a. If you are creating a configurable service to be used with a specific sequence, ensure that the name of the configurable service is the same as the name that you specify in the Configurable service property on the Resequencing node. If you are creating a configurable service to be used with all sequences in the integration server, ensure that the configurable service has the same name as the integration server.
 - b. Set the **Queue prefix** property to the required value.
 - c. Optional: Set the **Missing message timeout**, **Start of sequence**, and **End of sequence** properties.

For example, create a Resequencing configurable service called `myResequencingService`, which uses queues prefixed with `SYSTEM.BROKER.EDA.SET1`, with a missing message timeout of 60 seconds, and which waits five seconds before determining the start and end numbers in a sequence:

```
mqsicreateconfigurable MYBROKER -c Resequencing -o myResequencingService  
-n queuePrefix,missingMessageTimeoutSeconds,startSequenceSeconds,endSequenceSeconds -v SET1,60,5,5
```

You can use the **mqsdeleteconfigurable** command to delete the Resequencing configurable service. However, the storage queues are not deleted automatically when the configurable service is deleted, so you must delete them separately. For more information, see Configurable services properties

3. In the Resequencing node:
 - a. If the configurable service is to be used for a specific sequence, specify the name of the configurable service on the **Advanced** tab; for example, `myResequencingService`. If you do not set the Configurable service property, and if a configurable service exists with the same name as the integration server, that configurable service is used instead.
 - b. Optional: Use the **mqsichangeproperties** and **mqsireportproperties** commands to change or view the properties of the configurable service. Alternatively, you can use the IBM Integration Explorer to view or modify a configurable service. For more information about working with configurable services, see “Using the IBM Integration Explorer to work with configurable services” on page 345.

What to do next

The properties for the configurable service are not used by the broker until you restart or redeploy the message flow, or restart the broker.

Related concepts:

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.

Related tasks:

“Viewing configurable services” on page 348

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view the properties of configurable services defined on your integration node.

“Modifying an IBM defined configurable service” on page 349

You can create a new configurable service for an external service on which the broker relies, by modifying predefined configurable services provided by IBM. Use the IBM Integration Explorer to view and modify existing configurable services.

“Modifying a configurable service” on page 350

Use the IBM Integration Explorer to view and modify existing configurable services.

“Deleting a configurable service” on page 353

Use the IBM Integration Explorer to delete custom configurable services.



Developing integration solutions

IBM Integration Bus provides a flexible environment in which you can develop integration solutions to transform, enrich, route, and process your business messages and data. You can integrate client applications that use different protocols and message formats.

Related reference:



Resequence node

Use the Resequence node to control the sequence in which a group (or groups) of incoming messages are propagated in a message flow.



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.



`mqsicreateconfigurable-service` command

Use the `mqsicreateconfigurable-service` command to create an object name for a broker external resource.



`mqsichange-properties` command

Use the `mqsichange-properties` command to modify broker properties and properties of broker resources.



`mqsi-report-properties` command

Use the `mqsi-report-properties` command to display properties that relate to a broker, an integration server, or a configurable service.

Configuring the storage of events for timeout nodes:

You can use a Timer configurable service to control the storage of events for TimeoutNotification and TimeoutControl nodes.

About this task

By default, the storage queue used by all timeout nodes is the SYSTEM.BROKER.TIMEOUT.QUEUE.

However, you can control the queues that are used by different timeout nodes by creating alternative queues that contain a *QueuePrefix* variable, and by using a Timer configurable service to specify the names of those queues for storing events.

Follow these steps to specify the queue that is used to store event states:

Procedure

1. Create the storage queue to be used by the timeout nodes. The following queue is required:

- SYSTEM.BROKER.TIMEOUT.*QueuePrefix*.QUEUE

The *QueuePrefix* variable can contain any characters that are valid in a WebSphere MQ queue name, but must be no longer than eight characters and must not begin or end with a period (.). For example, SET1 and SET.1 are valid queue prefixes, but .SET1 and SET1. are invalid.

If you do not create the storage queue, IBM Integration Bus creates the queue when the node is deployed; this queue is based on the default queue. If the queue cannot be created, the message flow is not deployed.

2. Use the **mqscreateconfigurable** command to create a Timer configurable service. You can create a configurable service to be used with either specific timeout requests or with all timeout requests in an integration server.
 - a. If the configurable service is to be used with specific timeout requests, create the configurable service with the same name as the Unique identifier property on the TimeoutNotification and TimeoutControl nodes. If the configurable service is to be used with all timeout requests in an integration server, create the configurable service with the same name as the integration server.
 - b. Set the **Queue prefix** property to the required value.

For example, create a Timer configurable service that uses a queue prefixed with SYSTEM.BROKER.TIMEOUT.SET1:

```
mqscreateconfigurable IB9NODE -c Timer -o myTimer  
-n queuePrefix -v SET1
```

You can use the **msideleteconfigurable** command to delete the Timer configurable service. However, the storage queue is not deleted automatically when the configurable service is deleted, so you must delete it separately. For more information, see Configurable services properties.

3. In the TimeoutNotification and TimeoutControl nodes:
 - a. Ensure that the name of the Timer configurable service is the same as the name specified in the Unique Identifier property on the **Basic** tab; for example, myTimer. If there is no Timer configurable service with the same name as the Unique Identifier, and if there is a configurable service with the same name as the integration server, that configurable service is used instead.

- b. Optional: Use the `mqsichangeproperties` and `mqsireportproperties` commands to change or view the properties of the configurable service. Alternatively, you can use the IBM Integration Explorer to view or modify a configurable service. For more information about working with configurable services, see “Using the IBM Integration Explorer to work with configurable services” on page 345.

What to do next

The properties for the configurable service are not used by the broker until you restart or redeploy the message flow, or restart the broker.

Related concepts:

“Using the IBM Integration Explorer to work with configurable services” on page 345

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view, add, modify and delete configurable services.



Configuring timeout flows

Use the TimeoutControl and TimeoutNotification nodes in message flows to process timeout requests or to generate timeout notifications at specified intervals.



Sending timeout request messages

To set a controlled timeout, send a message with a set of elements with well known names to a TimeoutControl node. These elements control the properties of the timeout to be created or deleted.

Related tasks:

“Viewing configurable services” on page 348

Configurable services are used to define properties that are related to external services on which the integration node (broker) relies. Use the IBM Integration Explorer to view the properties of configurable services defined on your integration node.

“Modifying an IBM defined configurable service” on page 349

You can create a new configurable service for an external service on which the broker relies, by modifying predefined configurable services provided by IBM. Use the IBM Integration Explorer to view and modify existing configurable services.

“Modifying a configurable service” on page 350

Use the IBM Integration Explorer to view and modify existing configurable services.

“Deleting a configurable service” on page 353

Use the IBM Integration Explorer to delete custom configurable services.

Related reference:



TimeoutNotification node

Use the TimeoutNotification node to manage timeout-dependent message flows.



TimeoutControl node

Use the TimeoutControl node to process an input message that contains a timeout request.



Configurable services properties

The supplied configurable services, and the configurable services that you create, are defined by their names and properties. You can use the supplied services.

mqsicreateconfigurable-service command

Use the **mqsicreateconfigurable-service** command to create an object name for a broker external resource.

mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.

mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.

Configuring the XPath cache:

The integration server XPath cache size might become a performance bottleneck for customers who use many XPath expressions. Altering the size of the XPath cache might improve message flow performance.

An integration server (or 'execution group' in WebSphere Message Broker Version 8.0 and earlier versions) keeps a cache of compiled XPath expressions to help reduce the processor usage of parsing and re-creating XPath expressions that are used repeatedly during message flow execution. This cache is shared by all message flows within an integration server. The default size of this cache is 100 elements. However, this default size might become a performance bottleneck for customers who use many XPath expressions with a single flow invocation completely invalidating the cache.

It might be necessary to alter the size of this cache for optimal message flow performance if many XPath expressions are created for each Message Flow invocation. In a highly multi-threaded environment where many XPath expressions are evaluated on each message flow invocation, it might be necessary to disable the cache to remove thread contention on the cache.

The property can be set by running the following **mqsichangeproperties** command:

```
mqsichangeproperties <broker> -e <eg> -o ExecutionGroup -n compiledXPathCacheSizeEntries -v <value>
```

where *<value>* is the size of the cache to be set. The size can be set to any value greater than or equal to 100. A value of 0 means that the cache is disabled. The default value is 100.

The configured value can be reported by running the following **mqsireportproperties** command:

```
mqsireportproperties <broker> -e <eg> -o ExecutionGroup -n compiledXPathCacheSizeEntries
```

and can also be reported as part of the other ExecutionGroup level properties:

```
mqsireportproperties <broker> -e <eg> -o ExecutionGroup -a
```

To disable the cache for broker *BRK1* and integration server *EG1*, run the following command:

```
mqsichangeproperties BRK1 -e EG1 -o ExecutionGroup -n compiledXPathCacheSizeEntries -v 0
```

Related concepts:

XPath overview

The XML Path Language (XPath) is used to uniquely identify or address parts of an XML document. An XPath expression can be used to search through an XML document, and extract information from any part of the document, such as an element or attribute (referred to as a *node* in XML) in it. XPath can be used alone or in conjunction with XSLT.

Related tasks:

Using XPath

XPath provides an alternative method to ESQL for entering expressions in the property fields of specific built-in nodes.

Configuring monitoring event sources using a monitoring profile:

You can create a monitoring profile and use the **mqsichangeflowmonitoring** command to configure your message flows to emit monitoring events.

Before you begin

Before you start:

Read the following topics:

- Business-level monitoring
- Monitoring basics

You must have a message flow that contains a node to which you want to add a monitoring event.

You can use XPath 1.0 expressions to configure a monitoring event.

Creating a monitoring profile:

About this task

First create a monitoring profile XML file. This is a file that lists the event sources in the message flow that will emit events, and defines the properties of each event.

Procedure

Follow the guidance at Monitoring profile to create your monitoring profile XML file.

Applying a monitoring profile:

About this task

When you have created a monitoring profile XML file, follow these steps to apply it.

Procedure

1. Use the **mqsicreateconfigurable** command to create a configurable service for the monitoring profile. In the following command example, replace *myBroker* with the name of your broker, and *myMonitoringProfile* with the name of your monitoring profile.

```
mqsicreateconfigurable myBroker -c MonitoringProfiles  
-o myMonitoringProfile
```

2. Use the **mqsichangeproperties** command to associate your monitoring profile XML file with the configurable service. In the following command example, replace *myBroker* with the name of your broker, *myMonitoringProfile* with the name of your monitoring profile, and *myMonitoringProfile.xml* with the name of the monitoring profile XML file.

```
mqsichangeproperties myBroker -c MonitoringProfiles -o myMonitoringProfile  
-n profileProperties -p myMonitoringProfile.xml
```

Set the `useParserNameInMonitoringPayload` property to `TRUE` to force the `wmb:applicationData/wmb:complexContent/wmb:elementName` attribute to hold the name of the input node parser, if present. See `MonitoringProfiles` configurable service for details.

```
mqsichangeproperties myBroker -c MonitoringProfiles -o myMonitoringProfile  
-n useParserNameInMonitoringPayload -v TRUE
```

3. Use the **mqsichangeflowmonitoring** command to apply a monitoring profile configurable service to one or more message flows.

- Apply a monitoring profile to a single message flow *messageflow1* in integration server *IS1*:

```
mqsichangeflowmonitoring myBroker -e IS1  
-f messageflow1 -m myMonitoringProfile
```

- Apply a monitoring profile to all message flows in all integration servers:

```
mqsichangeflowmonitoring myBroker -g -j -m myMonitoringProfile
```

Monitoring for the flow is inactive; applying the monitoring profile does not activate it.

4. Alternatively, use the broker archive editor to apply a monitoring profile configurable service to one or more message flows, by setting message flow property **Monitoring Profile Name**.

- a. In the IBM Integration Toolkit, switch to the Integration Development perspective.

- b. In the Application Development view, right-click the BAR file, then click **Open with > Broker Archive Editor**.

- c. Click the **Manage and Configure** tab.

- d. Click the message flow on which you want to set the monitoring profile configurable service. The properties that you can configure for the message flow are displayed in the **Properties** view.

- e. In the **Monitoring Profile Name** field, enter the name of a monitoring profile.

- f. Save the BAR file.

- g. Deploy the BAR file.

Monitoring for the flow is inactive; deploying the BAR file does not activate it.

5. Activate monitoring for the flow using the `mqsichangeflowmonitoring -c` command.

- Activate monitoring for a single message flow *messageflow1* in integration server *IS1*:

```
mqsichangeflowmonitoring myBroker -e IS1  
-f messageflow1 -c active
```

- Activate monitoring for all message flows in all integration servers:

```
mqsichangeflowmonitoring myBroker -g -j -c active
```

Updating a monitoring profile:

Procedure

1. Follow the guidance at Monitoring profile to update your monitoring profile XML file.
2. Use the **mqsichangeproperties** command to update the configurable service to use the new XML file. For example:

```
mqsichangeproperties myBroker -c MonitoringProfiles -o myMonitoringProfile -n profileProperties -p myMonitoringProfile.xml
```
3. You must stop and start the integration server for the changes of property values to take effect.

Related concepts:

Monitoring basics

Message flows can be configured to emit events. The events can be read and used by other applications for transaction monitoring, transaction auditing, and business process monitoring.

Configurable services

Configurable services are typically runtime properties. You can use them to define properties that are related to external services on which the broker relies; for example, an SMTP server or a JMS provider.

Related tasks:

Business-level monitoring

You can configure your message flow to emit event messages that can be used to support transaction monitoring and auditing, and business process monitoring.

Activating monitoring

Use the **mqsichangeflowmonitoring** command to activate monitoring after you have configured monitoring event sources.

Related reference:

Monitoring profile

To customize events after a message flow has been deployed, but without redeploying the flow, use a monitoring profile configurable service. By using this service, you can apply a monitoring profile to one or more message flows.

mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.

mqsicreateconfigurable-service command

Use the **mqsicreateconfigurable-service** command to create an object name for a broker external resource.

mqsichangeflowmonitoring command

Use the **mqsichangeflowmonitoring** command to enable monitoring of message flows.

mqsireportflowmonitoring command

Use the **mqsireportflowmonitoring** command to display the current options for monitoring that have been set using the **mqsichangeflowmonitoring** command.

MonitoringProfiles configurable service

Select the objects and properties that you want to change for the MonitoringProfiles configurable service.



Example XPath expressions for event filtering

Use numeric, string, or Boolean expressions when configuring an event source, to determine whether the event is emitted.

Web Services Reliable Messaging

IBM Integration Bus supports WS-RM (Web Services Reliable Messaging), which allows two systems to reliably exchange messages with each other.

Web Services Reliable Messaging (WS-RM) is an OASIS standard that allows two systems to reliably exchange SOAP messages with each other. The purpose of WS-RM is to ensure delivery of messages in situations such as the destination endpoint being temporarily unavailable (for example, in the case of a server restart) or the message path crossing multiple transport connections, any of which might fail (for example, across a firewall). WS-RM offers greater reliability when using HTTP transport, but has a performance impact.

WS-RM is applicable only to HTTP transport. If you configure WS-RM on a message flow that uses JMS transport, the WS-RM settings are not used when the flow is deployed.

Systems that implement WS-RM retransmit messages that have not been successfully delivered and acknowledged, and prevent duplicate messages from being delivered to the application destination. WS-RM is a web services protocol and can be used with WS-Security and WS-Addressing.

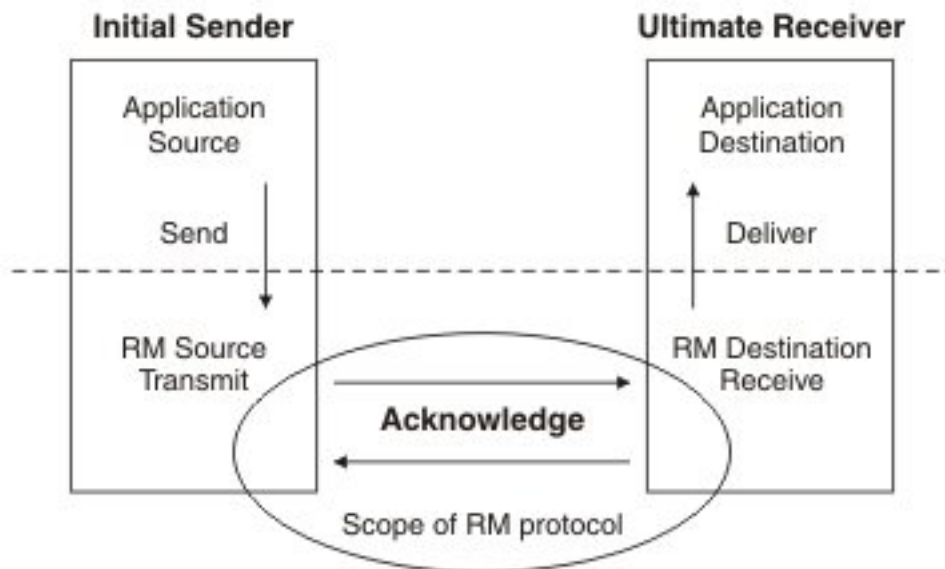


Figure 6. The initial sender transmits a message from the application source to the reliable messaging source. The reliable messaging source transmits the message to the reliable messaging destination, which acknowledges receipt of the message. The reliable messaging destination then delivers the message to the application destination, and so it reaches the Application Destination.

Reliable messaging takes place between two endpoints known as the *reliable messaging source* and the *reliable messaging destination*. Before the messages are sent, the reliable source and reliable destination perform a message exchange to establish a *Sequence*. A Sequence is identified by a unique identifier and comprises

a sequence of messages which are numbered starting from one. Sending a group of messages in a Sequence ensures the reliability of all the messages in that Sequence.

The reliable messaging source sends each message one or more times to the reliable messaging destination. The destination sends back acknowledgment for each message it receives to show that the message has been successfully received. If the reliable messaging source does not receive an acknowledgment that a message has been received by the destination, it sends the message again until an acknowledgment is received.

When all messages in a sequence have been successfully received by the destination and acknowledgment received by the source, the source sends a `TerminateSequence` message to instruct the destination that the message sequence is complete.

If the client is waiting for messages that have not been delivered, it can initiate a `WS-MakeConnection` request. `WS-MakeConnection` is a specification that describes how messages can be exchanged between a server and a client using a transport-specific back-channel. The client's `MakeConnection` request allows the server to respond with any queued messages that have not been received by the client.

IBM Integration Bus does not support using HTTP compression or SSL with WS-RM.

Message order

WS-RM specifies a number of delivery assurances to be supported by the provider. The `InOrder` assurance asserts that messages are delivered to the application destination by the RM Destination in the order in which they were sent, according to their sequence numbers. For example, if the RM Destination receives messages in the order `m1`, `m3`, `m2`, it first delivers `m1`, then retains `m3` until it receives `m2`, and then delivers `m2` and `m3` to the application destination. Messages are not persisted in the event of a broker restart.

The following diagram shows an example scenario where `InOrder` is not used, and messages need not be delivered in the order in which they were sent. The second message in the sequence is lost, but the third message is processed even though the second message has not been delivered.

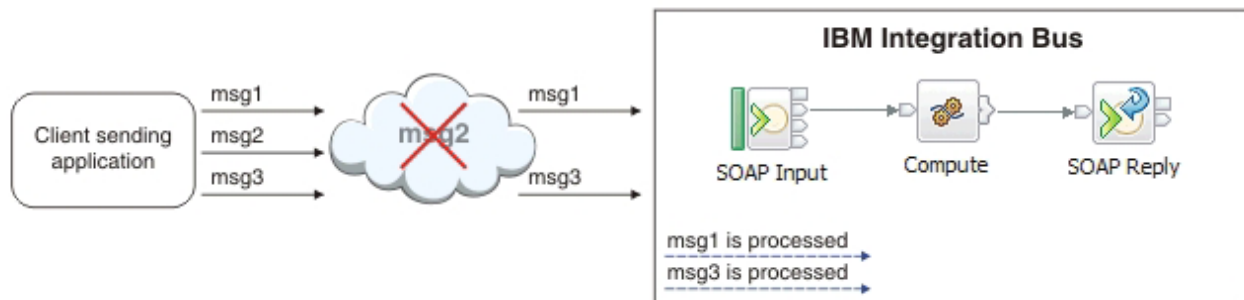


Figure 7. With `InOrder` set to `false`, all messages are processed when they are delivered.

In the following scenario the option is selected, and messages must be delivered in the order in which they are sent. The second message in the sequence is lost, and so the third message is not processed until `msg2` can be delivered.

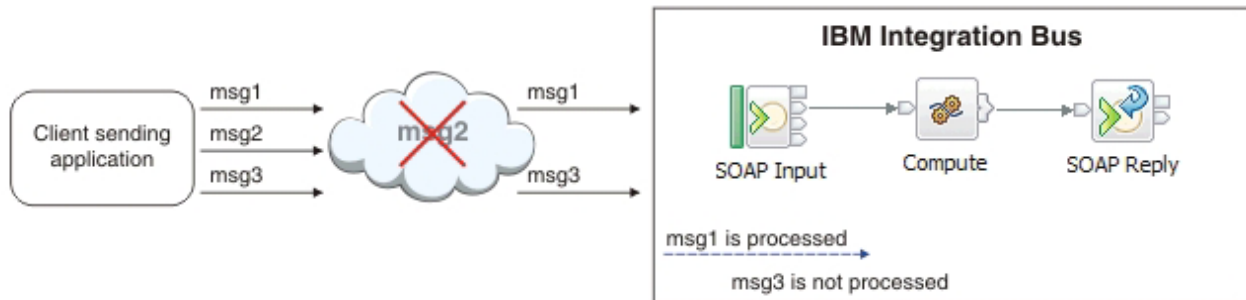


Figure 8. With InOrder set to True, message 3 is not processed because message 2 has not yet been delivered.

The InOrder assurance affects only inbound message processing, for example, when the policy set that asks for InOrder delivery is associated with a SOAPInput node or an inbound message flow. When InOrder delivery is enabled for inbound SOAP message processing, the messages are processed by the flow according to their message number as assigned by the RM Source.

The InOrder assurance does not apply to outbound message processing. If InOrder delivery is selected for outbound SOAP message processing, for example, with a SOAPRequest node, messages are not necessarily delivered to the application destination in the same order in which they are sent from the message flow.

For information about scalability when using InOrder with WS-RM, see Tuning SOAP processing for scalability and performance.

Related concepts:

“Policy sets” on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

 Tuning SOAP processing for scalability and performance

You can improve performance when processing SOAP messages by configuring the number of additional instances on the message flow.

 Example reliable message exchange

IBM Integration Bus supports Web Services Reliable Messaging (WS-RM), which allows two systems to reliably exchange messages with each other.

 WS-MakeConnection

WS-MakeConnection is a specification that describes how messages can be exchanged between a server and a client using a transport-specific back-channel.

Related tasks:

“Configuring WS-RM”

Configure reliable messaging for web services.

OASIS Standard for WS-RM Specification

Configuring WS-RM

Configure reliable messaging for web services.

About this task

Use the Policy Sets and Policy Set Bindings editor in the IBM Integration Explorer to configure the following aspects of WS-RM for SOAP messages sent over HTTP transport:

“Message Order”

“Client compatibility”

A message flow with WS-RM enabled can still process messages that do not use the WS-RM protocol.

Message Order: About this task

Message order is one of the WS-RM message delivery assurances. You can configure messages to be processed by the message flow in the order in which they were sent. For more information, see “Message order” on page 470.

To enable this message order assurance on the policy set, complete the following steps:

Procedure

1. In the IBM Integration Explorer, right-click on the broker with which you want to work, and click **Properties**.
2. In the Properties window, select the Security tab, and click **Policy Sets**.
3. Create a policy set and add a new WS-RM policy to it.
4. Select the WS-RM policy, and select the option under **Message Order** to ensure that messages are delivered in the order that they are sent.
5. Associate the policy set with a message flow or node; see “Associating policy sets and bindings with message flows and nodes” on page 502.

Results

If WS-RM is enabled and messages are configured to be delivered in the order in which they were sent, this means that later messages are not delivered to the application destination until all earlier messages have arrived and have been delivered. This could result in a large number of blocked undelivered messages if an earlier message is lost.

Enabling the WS-RM InOrder assurance for inbound messages might have an impact on scalability when configuring additional instances. For more information, see Tuning SOAP processing for scalability and performance.

Client compatibility:

About this task

IBM Integration Bus provides a client compatibility option for working with some WS-RM providers, for example, Microsoft .NET, that use an older version of the protocol. When consuming a web service from one of these providers, ensure that the Client compatibility option Microsoft .NET WCF is selected. For other providers, select Default.

Procedure

1. In the IBM Integration Explorer, right-click on the broker with which you want to work, and click **Properties**.
2. In the Properties window, select the Security tab, and click **Policy Sets**.
3. Create a policy set and add a new WS-RM policy to it. Alternatively, modify an existing policy.
4. Select the WS-RM policy, and select the appropriate option under **Client compatibility** to ensure compatibility with the WS-RM provider.
5. Associate the policy set with a message flow or node; see “Associating policy sets and bindings with message flows and nodes” on page 502.

Related concepts:

“Web Services Reliable Messaging” on page 469

IBM Integration Bus supports WS-RM (Web Services Reliable Messaging), which allows two systems to reliably exchange messages with each other.



Tuning SOAP processing for scalability and performance

You can improve performance when processing SOAP messages by configuring the number of additional instances on the message flow.



Example reliable message exchange

IBM Integration Bus supports Web Services Reliable Messaging (WS-RM), which allows two systems to reliably exchange messages with each other.

“Policy sets” on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

Related tasks:

“Associating policy sets and bindings with message flows and nodes” on page 502

Use the Broker Archive editor to associate policy sets and bindings with message flows and nodes, so that they are available to the broker at run time.

Related reference:



Policy Sets and Policy Set Bindings editor

The Policy Sets and Policy Set Bindings editor is the default editor for editing, saving, importing, and exporting a policy set or binding. You can define all the WS-Security policies for a single node, or a set of nodes. The policy set can be associated by the administrator with either a message flow or a node.

WS-Security

Web Services Security (WS-Security) describes enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication. WS-Security mechanisms can be used to accommodate a wide variety of security models and encryption technologies.

WS-Security is a message-level standard that is based on securing SOAP messages through XML digital signature, confidentiality through XML encryption, and credential propagation through security tokens. The Web services security specification defines the facilities for protecting the integrity and confidentiality of a message and provides mechanisms for associating security-related claims with the message.

WS-Security provides a general-purpose mechanism for associating security tokens with messages. No specific type of security token is required by WS-Security. It is designed to be extensible, for example, to support multiple security token formats.

WS-Security also describes how to encode binary security tokens and attach them to SOAP messages. Specifically, the WS-Security profile specifications describe how to encode the following tokens:

- Username tokens
- X.509 certificates
- SAML assertions
- Kerberos tickets
- LTPA binary tokens

With WS-Security, the domain of these mechanisms can be extended by carrying authentication information in Web services requests. WS-Security also includes extensibility mechanisms that can be used to further describe the credentials that are included with a message. WS-Security is a building block that can be used in conjunction with other Web service protocols to address a wide variety of application security requirements.

There are numerous advantages to using WS-Security.

- Different parts of a message can be secured in a variety of ways. For example, you can use integrity on the security token (user ID and password) and confidentiality on the SOAP message body.
- Intermediaries can be used and end-to-end message-level security can be provided through any number of intermediaries.
- WS-Security works across multiple transports and is independent of the underlying transport protocol.
- Authentication of both individual users and multiple party identities is possible.

Traditional Web security mechanisms, such as HTTPS, might be insufficient to manage the security requirements of all Web service scenarios. For example, when an application sends a SOAP message using HTTPS, the message is secured only for the HTTPS connection, meaning during the transport of the message between the service requester (the client) and the service. However, the application might require that the message data be secured beyond the HTTPS connection, or even beyond the transport layer. By securing Web services at the message level, message-level security is capable of meeting these expanded requirements.

Message-level security, or securing Web services at the message level, addresses the same security requirements as for traditional Web security. These security requirements include: identity, authentication, authorization, integrity, confidentiality, nonrepudiation, and basic message exchange. Both traditional Web and message-level security share many of the same mechanisms for handling security, including digital certificates, encryption, and digital signatures.

With message-level security, the SOAP message itself either contains the information needed to secure the message or it contains information about where to get that information to handle security needs. The SOAP message also contains information relevant to the protocols and procedures for processing the specified message-level security. However, message-level security is not tied to any particular transport mechanism. Because the security information is part of the message, it is independent of a transport protocol, such as HTTPS.

The client adds to the SOAP message header security information that applies to that particular message. When the message is received, the Web service endpoint, using the security information in the header, verifies the secured message and validates it against the policy. For example, the service endpoint might verify the message signature and check that the message has not been tampered with. It is possible to add signature and encryption information to the SOAP message headers, as well as other information such as security tokens for identity (for example, an X.509 certificate) that are bound to the SOAP message content.

Without message-level security, the SOAP message is sent in clear text, and personal information such as a user ID or an account number is not protected. Without applying message-level security, there is only a SOAP body under the SOAP envelope in the SOAP message. By applying features from the WS-Security specification, the SOAP security header is inserted under the SOAP envelope in the SOAP message when the SOAP body is signed and encrypted.

To keep the integrity or confidentiality of the message, digital signatures and encryption are typically applied.

- Confidentiality specifies the confidentiality constraints that are applied to generated messages. This includes specifying which message parts within the generated message must be encrypted, and the message parts to attach encrypted Nonce and time stamp elements to.
- Integrity is provided by applying a digital signature to a SOAP message. Confidentiality is applied by SOAP message encryption.

You can add an authentication mechanism by inserting various types of security tokens, such as the Username token (element). When the Username token is received by the Web service server, the user name and password are extracted and verified. Only when the user name and password combination is valid, will the message be accepted and processed at the server. Using the Username token is just one of the ways of implementing authentication. This mechanism is also known as basic authentication.

The OASIS Web Services Security Specification provides a set of mechanisms to help developers of Web Services secure SOAP message exchanges. For details of the OASIS Web Services Security Specification, see OASIS Standard for WS-Security Specification.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“WS-Security capabilities” on page 507

Web service security capabilities are supported by the broker.



Public key cryptography

All encryption systems rely on the concept of a key. A key is the basis for a transformation, usually mathematical, of an ordinary message into an unreadable message. For centuries, most encryption systems have relied on private key encryption. Public key encryption is the only challenge to private key encryption that has appeared in the last 30 years.



Digital certificates

Certificates provide a way of authenticating users. Instead of requiring each participant in an application to authenticate every user, third-party authentication relies on the use of digital certificates.

Digital signatures

A digital signature is a number that is attached to a document. For example, in an authentication system that uses public-key encryption, digital signatures are used to sign certificates.

“Policy sets” on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

WS-Security mechanisms

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

Authentication

This mechanism uses a security token to validate the user and determine whether a client is valid in a particular context. A client can be a user, computer, or application. Without authentication, an attacker can use spoofing techniques to send a modified SOAP message to the service provider.

In authentication, a security token is inserted in the request message. Depending on the type of security token that is being used, the security token can also be inserted in the response message. The following types of security token are supported for authentication:

- Username tokens
- X.509 certificates
- SAML assertions
- Kerberos tickets
- LTPA binary tokens

Username tokens are used to validate user names and passwords. When a Web service server receives a username token, the user name and password are extracted and passed to a user registry for verification. If the user name and password combination is valid, the result is returned to the server and the message is accepted and processed. When used in authentication, username tokens are typically passed only in the request message, not the response message.

X.509 tokens are validated by using a certificate path.

The broker support for SAML assertions is restricted to passing the token to a WS-Trust security token server (STS) for validation.

Kerberos tickets are validated against the host's Kerberos keytab file.

The broker support for LTPA binary tokens is restricted to passing the token to a WS-Trust STS for validation.

All types of token must be protected. For this reason, if you send them over an untrusted network, take one of the following precautions:

- Use HTTPS
- Configure the policy set to protect the appropriate elements in the SOAP header

Integrity

This mechanism uses message signing to ensure that information is not changed, altered, or lost accidentally. When integrity is implemented, an XML digital signature is generated on the contents of a SOAP message. If unauthorized changes are made to the message data, the signature is not validated. Without integrity, an attacker can use tampering techniques to intercept a SOAP message between the Web service client and server, and modify it.

Confidentiality

This mechanism uses message encryption to ensure that no party or process can access or disclose the information in the message. When a SOAP message is encrypted, only a service that knows the appropriate key can decrypt and read the message.

Related concepts:

“WS-Security” on page 473

Web Services Security (WS-Security) describes enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication. WS-Security mechanisms can be used to accommodate a wide variety of security models and encryption technologies.

Public key cryptography

All encryption systems rely on the concept of a key. A key is the basis for a transformation, usually mathematical, of an ordinary message into an unreadable message. For centuries, most encryption systems have relied on private key encryption. Public key encryption is the only challenge to private key encryption that has appeared in the last 30 years.

Digital certificates

Certificates provide a way of authenticating users. Instead of requiring each participant in an application to authenticate every user, third-party authentication relies on the use of digital certificates.

Digital signatures

A digital signature is a number that is attached to a document. For example, in an authentication system that uses public-key encryption, digital signatures are used to sign certificates.

“Policy sets” on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

Implementing WS-Security

Configure authentication, XML encryption, XML signature, and message expiration by using the Policy Sets and Policy Set Bindings editor.

About this task

You use the Policy Sets and Policy Set Bindings editor in the IBM Integration Explorer to configure the following aspects of WS-Security:

“Authentication” on page 478

“Confidentiality” on page 479

“Integrity” on page 480

“Expiration” on page 481

Authentication:
About this task

The following tokens are supported:

- Username
- X.509
- SAML assertions
- Kerberos tickets
- LTPA binary tokens

Configuring authentication with username tokens:

1. In the IBM Integration Explorer, right-click on the broker with which you want to work, and click **Properties**.
2. In the Properties window, select the Security tab, and click **Policy Sets**.
3. Create a policy set and add UserName authentication tokens to it; see Policy Sets and Policy Set Bindings editor: Authentication tokens panel.
4. Further configure any X.509 authentication tokens defined in the associated policy set; see Policy Sets and Policy Set Bindings editor: Authentication and Protection Tokens panel.
5. Configure a security profile; see “Message flow security and security profiles” on page 505.
6. Associate the policy set with a message flow or node; see “Associating policy sets and bindings with message flows and nodes” on page 502.

Configuring authentication with X.509 tokens:

1. If you are using the broker's truststore to hold the trusted certificate, you must configure it; see “Viewing and setting keystore and truststore runtime properties at broker level” on page 497 or “Viewing and setting keystore and truststore runtime properties at integration server level” on page 499 depending on where you want to set keystore and truststore runtime properties.
2. Create a policy set and add UserName and X.509 authentication tokens to it; see Policy Sets and Policy Set Bindings editor: Authentication tokens panel.
3. Configure the certificate mode for either broker truststore or an external security provider; see Policy Sets and Policy Set Bindings editor: Authentication and Protection Tokens panel.
4. If you are using an external security provider, configure a security profile; see “Message flow security and security profiles” on page 505.
5. Associate the policy set with a message flow or node; see “Associating policy sets and bindings with message flows and nodes” on page 502

Configuring authentication with SAML assertions:

1. In the IBM Integration Explorer, right-click on the broker with which you want to work, and click **Properties**.
2. In the Properties window, select the Security tab, and click **Policy Sets**.
3. Create a policy set and add SAML pass-through 1.1 or SAML pass-through 2.0 tokens to it; see Policy Sets and Policy Set Bindings editor: Authentication tokens panel. SAML pass-through does not enforce subject confirmation, but the assertion is simply provided as a

token to be processed in the external Security Token Server specified in the security profile that is associated with the node.

4. Configure a security profile. The security profile must be configured to use a WS-Trust v1.3 STS. For more information, see “Message flow security and security profiles” on page 505.
5. Associate the policy set with a message flow or node; see “Associating policy sets and bindings with message flows and nodes” on page 502.

Configuring authentication with Kerberos tickets:

1. In the IBM Integration Explorer, right-click on the broker with which you want to work, and click **Properties**.
2. In the Properties window, select the Security tab, and click **Policy Sets**.
3. Create a policy set and add your Kerberos token type as symmetric tokens; see Policy Sets and Policy Set Bindings editor: Message Level Protection panel.
4. Associate the policy set with a message flow or node; see “Associating policy sets and bindings with message flows and nodes” on page 502.
5. Configure the host's Kerberos keytab file. For more information about Kerberos configuration, see the documentation for your broker's host system. For example, for Windows, see the "Step-by-Step Guide to Kerberos 5 (krb5 1.0) Interoperability", which you can access at <http://technet.microsoft.com/en-us/library/>.

Configuring authentication with LTPA binary tokens:

1. In the IBM Integration Explorer, right-click on the broker with which you want to work, and click **Properties**.
2. In the Properties window, select the Security tab, and click **Policy Sets**.
3. Create a policy set and add LTPA tokens to it; see Policy Sets and Policy Set Bindings editor: Authentication tokens panel. The LTPA binary token is passed through to the external Security Token Server (STS) specified in the security profile that is associated with the node.
4. Configure a security profile. The security profile must be configured to use a WS-Trust v1.3 STS. For more information, see “Message flow security and security profiles” on page 505.
5. Associate the policy set with a message flow or node; see “Associating policy sets and bindings with message flows and nodes” on page 502.

Confidentiality:

About this task

Confidentiality is provided by XML encryption, and requires either X.509 tokens or Kerberos tickets.

Configuring XML encryption with X.509 tokens:

1. If you are using the broker's truststore to hold the trusted certificate, you must configure it; see “Viewing and setting keystore and truststore runtime properties at broker level” on page 497 or “Viewing and setting keystore and truststore runtime properties at integration server level” on page 499, depending on where you want to set keystore and truststore runtime properties.
2. In the IBM Integration Explorer, right-click on the broker with which you want to work, and click **Properties**.
3. In the Properties window, select the **Security** tab, and click **Policy Sets**.

4. Create a policy set, enable XML encryption, create encryption tokens, and select the encryption algorithms that you will use; see Policy Sets and Policy Set Bindings editor: Message Level Protection panel.
5. Define which parts of a message are to be encrypted; see Policy Sets and Policy Set Bindings editor: Message Part Protection panel.
6. Further configure message part encryption; see Policy Sets and Policy Set Bindings editor: Message Part Policies panel.
7. Further configure the keystore and truststore; see Policy Sets and Policy Set Bindings editor: Key Information panel.
8. Associate the policy set with a message flow or node; see “Associating policy sets and bindings with message flows and nodes” on page 502.

Configuring XML encryption with Kerberos tickets:

1. Configure your host for Kerberos, providing a `krb.conf` configuration file. This step is required on all operating systems, including Windows.
2. Provide the broker with the Kerberos client credentials for accessing the Kerberos Key Distribution Center (KDC). These credentials (which are required for SOAPRequest nodes) can be provided in the Broker properties tree, or by using the `mqsisetdbparms` command. The credentials are taken in order of priority:
 - The node has a security profile with the **propagation** property set to *True* and the Properties tree username and password token is present. If no Username and password token exists, an exception is thrown.
 - `mqsisetdbparms kerberos::<realm>::<integration server name>`
 - `mqsisetdbparms kerberos::<realm>`
 - `mqsisetdbparms kerberos::kerberos`
3. In the IBM Integration Explorer, right-click on the broker with which you want to work, and click **Properties**.
4. In the Properties window, select the **Security** tab, and click **Policy Sets**.
5. Create a policy set and add the required Kerberos token type as Symmetric Tokens; see Policy Sets and Policy Set Bindings editor: Message Level Protection panel.

Integrity:

About this task

Integrity is provided by XML signature, and requires either X.509 tokens or Kerberos tickets.

Configuring XML signature with X.509 tokens:

1. If you are using the broker's truststore to hold the trusted certificate, you must configure it; see “Viewing and setting keystore and truststore runtime properties at broker level” on page 497 or “Viewing and setting keystore and truststore runtime properties at integration server level” on page 499 depending on where you want to set keystore and truststore runtime properties.
2. In the IBM Integration Explorer, right-click on the broker with which you want to work, and click **Properties**.
3. In the Properties window, select the Security tab, and click **Policy Sets**.
4. Create a policy set, enable XML signature, and create signature tokens; see Policy Sets and Policy Set Bindings editor: Message Level Protection panel.

5. Define which parts of a message are to be signed; see Policy Sets and Policy Set Bindings editor: Message Part Protection panel.
6. Further configure message part signature; see Policy Sets and Policy Set Bindings editor: Message Part Policies panel.
7. Further configure the keystore and truststore; see Policy Sets and Policy Set Bindings editor: Key Information panel.
8. Associate the policy set with a message flow or node; see “Associating policy sets and bindings with message flows and nodes” on page 502.

Configuring XML signature with Kerberos tickets:

1. Configure your host for Kerberos, providing a `krb.conf` configuration file. This step is required on all operating systems, including Windows.
2. Provide the broker with the Kerberos client credentials for accessing the Kerberos Key Distribution Center (KDC). These credentials (which are required for SOAPRequest nodes) can be provided in the Broker properties tree, or by using the `mqsisetdbparms` command. The credentials are taken in the following order of priority:
 - The node has a security profile with the **propagation** property set to *True* and the Properties tree username and password token is present. If no Username and password token exists, an exception is thrown.
 - `mqsisetdbparms kerberos::<realm>::<integration server name>`
 - `mqsisetdbparms kerberos::<realm>`
 - `mqsisetdbparms kerberos::kerberos`
3. In the IBM Integration Explorer, right-click on the broker with which you want to work, and click **Properties**.
4. In the Properties window, select the **Security** tab, and click **Policy Sets**.
5. Create a policy set and add the required Kerberos token type as Symmetric Tokens; see Policy Sets and Policy Set Bindings editor: Message Level Protection panel.

Expiration: Procedure

To configure message expiration, see Policy Sets and Policy Set Bindings editor: Message Expiration panel.

Related concepts:

“Policy sets” on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

“WS-Security capabilities” on page 507

Web service security capabilities are supported by the broker.

Related tasks:

“Viewing and setting keystore and truststore runtime properties at broker level” on page 497

Configure the message broker to refer to a keystore, a truststore, or both, before deploying any message flows that require policy set or bindings for signature, encryption, or X.509 Authentication.

“Viewing and setting keystore and truststore runtime properties at integration server level” on page 499

Configure an integration server to refer to a keystore, a truststore, or both, before deploying any message flows that require policy set or bindings for signature,

encryption, or X.509 Authentication.

“Associating policy sets and bindings with message flows and nodes” on page 502
Use the Broker Archive editor to associate policy sets and bindings with message flows and nodes, so that they are available to the broker at run time.

Related reference:



Policy Sets and Policy Set Bindings editor

The Policy Sets and Policy Set Bindings editor is the default editor for editing, saving, importing, and exporting a policy set or binding. You can define all the WS-Security policies for a single node, or a set of nodes. The policy set can be associated by the administrator with either a message flow or a node.



mqsisetdbparms command

Use the **mqsisetdbparms** command to associate a specific user ID and password (or SSH identity file) with one or more resources that are accessed by the broker.



mqsireportdbparms command

Use the **mqsireportdbparms** command to list all parameters that are set for a specific broker.

Kerberos-based WS-Security

You can use Kerberos authentication with WS-Security either as a service or as a client.

Kerberos is a network authentication protocol that enables mutual authentication with symmetric keys. Users and services on a network authenticate with each other through a Key Distribution Center (KDC), as a trusted third party. IBM Integration Bus provides support for Kerberos either as a service or as a client.

You can use message flows to call web services that are secured with Kerberos by using a SOAP Request node. You can also provide web services that are secured with Kerberos by using SOAP Input Nodes. The WS-Security header passes Kerberos tokens. You can sign and encrypt either parts or all of a SOAP message by using Kerberos tokens. Signing and encrypting messages provides message integrity, confidentiality, and authenticity.

For information about Kerberos terminology and concepts, see Concepts for Kerberos security.

You can configure IBM Integration Bus to act as a Kerberos secured service or as a client to a Kerberos secured service. Install a Kerberos KDC and configure it to contain all the principals that are required by the environment, then configure for a service or client. For more information about configuring Kerberos, see your host Kerberos documentation.

- For the steps needed to embed IBM Integration Bus as a client, see [Configuring IBM Integration Bus as a client to a Kerberos secured Service](#).
- For the steps needed to configure IBM Integration Bus as a secured service, see [Configuring IBM Integration Bus as a Kerberos secured Service](#).
- For the steps needed to configure separate Kerberos configuration files for each integration server, see [“Configuring separate Kerberos configuration files for each integration server”](#) on page 488.

Related concepts:

“WS-Security” on page 473

Web Services Security (WS-Security) describes enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication. WS-Security mechanisms can be used to accommodate a wide variety of security models and encryption technologies.

“WS-Security capabilities” on page 507

Web service security capabilities are supported by the broker.

Related reference:



Policy Sets and Policy Set Bindings editor

The Policy Sets and Policy Set Bindings editor is the default editor for editing, saving, importing, and exporting a policy set or binding. You can define all the WS-Security policies for a single node, or a set of nodes. The policy set can be associated by the administrator with either a message flow or a node.

Kerberos security concepts:

Learn about Kerberos authentication and WS-Security concepts.

Concepts

Kerberos configuration file

The configuration file contains information that is important for authentication and access. The configuration file contains the key distribution center (KDC) realm and location, supported encryption types, and keytab file location. Clients use the configuration file to authenticate with the KDC and request access to networked services. Kerberos secured services also use the configuration file to locate the keytab file that contains the private key that is associated with it.

Kerberos keytab file

The keytab file contains the principal and encrypted private key that is associated with the principal. The keytab file is created by exporting a principal from the KDC. Using the keytab file, a service can check the authenticity of a client and provide authentication without contacting the KDC.

Key Distribution Center (KDC)

The Key Distribution Center stores user and service principals with their associated key. A combination of either a user name and a password or a service name and a password provide the key. The KDC also provides an authentication server and a server that grants tickets.

Principals

Networked users and Services are known as principals. They authenticate with a Key Distribution Center (KDC).

Realm The unique range of control that is provided by the KDC. By convention the realm is the DNS domain name that is converted to uppercase.

Service Principle Name (SPN)

The service principal name represents a unique, networked service. For a client to use a Kerberos secured service, the client must authenticate with a KDC and provide the SPN for the service. The ticketing server provides a ticket to the client that allows it to authenticate itself to the service.

Related concepts:

“WS-Security” on page 473

Web Services Security (WS-Security) describes enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication. WS-Security mechanisms can be used to accommodate a wide variety of security models and encryption technologies.

Related reference:

“Kerberos token capabilities for authentication and authorization” on page 528

This topic describes the broker Web services capability for authentication, authorization, or both using a Kerberos token.



Policy Sets and Policy Set Bindings editor

The Policy Sets and Policy Set Bindings editor is the default editor for editing, saving, importing, and exporting a policy set or binding. You can define all the WS-Security policies for a single node, or a set of nodes. The policy set can be associated by the administrator with either a message flow or a node.

Configuring IBM Integration Bus as a client to a Kerberos secured service:

You can configure IBM Integration Bus to operate as a client to a Kerberos secured service for message integrity, confidentiality, and authenticity.

Before you begin

You must have access to a Key Distribution Center (KDC) and a server that is hosting the service. For more information about configuring Kerberos, see your host Kerberos documentation.

About this task

Procedure

1. Set the user credentials that are used to authenticate with the KDC.
 - You can configure the credentials at the broker level by issuing a **mqsisetdbparms**. For example,
`mqsisetdbparms brokerName -n SPN::realm -u username -p password`
 - You can also set the user credentials at the integration server level. For example, you can set a specific realm in any integration server with
`mqsisetdbparms brokerName -n kerberos::realm1::ExecutionGroup1 -u clientId -p password`
 - You can also use the Properties tree to set the credentials by using the following ESQL in a compute node:

```
SET OutputRoot.Properties.IdentitySourceType = 'usernameAndPassword';  
SET OutputRoot.Properties.IdentitySourceToken = Username;  
SET OutputRoot.Properties.IdentitySourcePassword = Password;
```
2. Create a Kerberos configuration file. Using the configuration file the client can authenticate with the KDC.

For more information about Kerberos-based WS-Security that is supported in SOAP nodes, see Message flow security and security profiles.

When you use Kerberos for security, the default Kerberos configuration file is the one on your workstation. The location for the configuration file differs depending on the system. The usual locations are:

- For Windows - C:\Windows\krb5.ini and C:\WINNT\krb5.ini
- For Linux - /etc/krb5.conf , UNIX (AIX) /etc/krb5/krb5.conf
- For z/OS - /krb5/krb5.conf

You can configure Kerberos configuration files for use by a broker or integration server.

The following sample Kerberos configuration file shows typical values for the variables. The variables *default_realm*, *default_keytab_name*, and the names in the *realms* are among the values you change in the configuration file, depending on your network and location of the configuration file.

```
[libdefaults]
default_realm = MYREALM.EXAMPLE.COM
default_keytab_name = FILE:c:\Windows\krb5.keytab
default_tkt_enctypes = rc4-hmac
default_tgs_enctypes = rc4-hmac
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
[realms]
MYREALM.EXAMPLE.COM = {
kdc = kdc.myrealm.example.com
admin_server = kdc.myrealm.example.com
}
```

For example, you can set the variables for an IBM Integration Bus level Kerberos configuration with

```
mqsichangeproperties brokerName -o BrokerRegistry -n kerberosConfigFile -v kerberosConfigLocation
```

For example, you can set the variables for an integration server level Kerberos configuration with

```
mqsichangeproperties brokerName -e integrationServerName -o ComIbmJVMManager -n kerberosConfigFile -v kerberosConfigLocation
```

3. Configure a policy set and binding that is associated with the SOAPRequest node for the BAR containing the message flow.
 - To configure the Kerberos token with a policy set, see Policy Sets and Policy Set Bindings editor: Message Level Protection panel.
 - To configure a consumer message part policy, see Policy Sets and Policy Set Bindings editor: Message Part Protection panel.
 - To optionally sign or encrypt message parts using the policy set, see Policy Sets and Policy Set Bindings editor: Message Part Policies panel.
 - To configure Kerberos specific settings such as target service name and target service realm, see Policy Sets and Policy Set Bindings editor: Message Part Policies panel.

Results


You have configured IBM Integration Bus to operate as a client to a Kerberos secured service.

Related concepts:

“Kerberos-based WS-Security” on page 482

You can use Kerberos authentication with WS-Security either as a service or as a client.

Related tasks:

 Accessing the Properties tree

The Properties tree has its own correlation name, Properties, and you must use this in all ESQL statements that refer to or set the content of this tree.

Related reference:

“Kerberos token capabilities for authentication and authorization” on page 528
This topic describes the broker Web services capability for authentication, authorization, or both using a Kerberos token.

mqsisetdbparms command

Use the **mqsisetdbparms** command to associate a specific user ID and password (or SSH identity file) with one or more resources that are accessed by the broker.

Configuring IBM Integration Bus as a Kerberos secured service:

You can configure IBM Integration Bus to operate as a Kerberos secured service for message integrity, confidentiality, and authenticity.

Before you begin

You must have access to a Key Distribution Center (KDC) and a server that is hosting the Kerberos secured service. For more information about configuring Kerberos, see your host Kerberos documentation.

About this task

Use this task to configure Kerberos as a secured service for IBM Integration Bus.

Procedure

1. Export a keytab that contains the private key of the service principal from the KDC. For example:

```
ktpass -out c:\Windows\krb5.keytab -princ SomePrincipal@YourDomain -crypto RC4-HMAC-NT mapUser Username -pass Password -mapOp set
```

where

out *filename*

Specifies the name and path of the keytab file to be generated.

princ *principal_name*

Specifies the principal name.

crypto *encryption_type*

Specifies the encryption type.

mapUser *username*

Maps the name of a Kerberos principal to a local account.

pass *password*

Specifies the password to use for this principal name.

mapOp *attribute*

Defines how the mapping attribute is set. The attribute alternatives are either `add` or `set`.

2. Copy the keytab file to the server that hosts the service. You can copy the file to the server by exporting the keytab file and transferring it to the server, for instance by using FTP. The Kerberos configuration file contains a reference to the keytab file in the form of a file URL (such as: `/home/user/my.keytab`). Because the reference is in the configuration file on the server, the server service can take on the Kerberos principal that is defined in the keytab.
3. Create a Kerberos configuration file that specifies the location of the keytab file on the local workstation.

You can use more than one service principle name per broker per Kerberos realm. Use your workstation default Kerberos configuration file when you are using Kerberos for security. The location for the configuration file differs depending on the system. The usual locations are:

- Windows: C:\Windows\krb5.ini and C:\WINNT\krb5.ini
- Linux : /etc/krb5.conf
- UNIX (AIX): /etc/krb5/krb5.conf
- z/OS: /krb5/krb5.conf

Different Kerberos configuration files can be configured for use by the broker and integration servers.

The following sample Kerberos configuration file shows typical values for the variables. The variables *default_realm*, *default_keytab_name*, and the names in the *realms* are among the values you change in the configuration file, depending on your network and location of the configuration file.

```
[libdefaults]
default_realm = MYREALM.EXAMPLE.COM
default_keytab_name = FILE:c:\Windows\krb5.keytab
default_tkt_enctypes = rc4-hmac
default_tgs_enctypes = rc4-hmac
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
[realms]
MYREALM.EXAMPLE.COM = {
kdc = kdc.myrealm.example.com
admin_server = kdc.myrealm.example.com
}
```

4. Create a new Kerberos configuration file to use unique Kerberos Service Principals per broker or per integration server. You do so by specifying the keytab file that contains the required service principal.
5. Use one of these **mqsichangeproperties** commands to specify the location of your new configuration file.
 - For a broker level Kerberos configuration:

```
mqsichangeproperties brokerName -o BrokerRegistry -n kerberosConfigFile -v kerberosConfigLocation
```
 - For an integration server level Kerberos configuration:

```
mqsichangeproperties brokerName -e integrationServerName -o ComIbmJVMManager -n kerberosConfigFile -v kerberosConfigLocation
```
6. Configure a policy set and binding that is associated with the SOAPInput node for the BAR containing the message flow.
 - To configure the Kerberos token with a policy set, see Policy Sets and Policy Set Bindings editor: Message Level Protection panel.
 - To configure a provider message part policy, see Policy Sets and Policy Set Bindings editor: Message Part Protection panel.
 - To sign or encrypt message parts optionally by using a policy set, see Policy Sets and Policy Set Bindings editor: Message Part Policies panel.
 - To configure Kerberos specific settings (such as, target service name and target service realm), see Policy Sets and Policy Set Bindings editor: Message Part Policies panel.

Results

You configured IBM Integration Bus to be a Kerberos secured service.

Related concepts:

“Kerberos-based WS-Security” on page 482

You can use Kerberos authentication with WS-Security either as a service or as a client.

Related tasks:

“Configuring separate Kerberos configuration files for each integration server”
You can create multiple Kerberos configuration files and then use a separate Kerberos configuration file with each integration server.

Related reference:

“Kerberos token capabilities for authentication and authorization” on page 528
This topic describes the broker Web services capability for authentication, authorization, or both using a Kerberos token.

**mqsisetdbparms** command

Use the **mqsisetdbparms** command to associate a specific user ID and password (or SSH identity file) with one or more resources that are accessed by the broker.

Configuring separate Kerberos configuration files for each integration server:

You can create multiple Kerberos configuration files and then use a separate Kerberos configuration file with each integration server.

Before you begin**Before you start:**

You must have access to a Key Distribution Center (KDC) and a server that is hosting the Kerberos-secured service, or you must have an existing Kerberos configuration file. For more information about configuring Kerberos, see your host Kerberos documentation.

About this task

Each integration server in an integration node has a separate JVM, and each JVM has a configuration setting for a Kerberos configuration file. Complete the following steps to configure a separate Kerberos configuration file for each integration server:

Procedure

1. Create a Kerberos configuration file or identify an existing Kerberos configuration file.
When you use Kerberos for security, the default Kerberos configuration file is the one on your workstation. The location for the configuration file differs depending on the system. The usual locations are as follows:
 - For Windows: C:\Windows\krb5.ini or C:\WINNT\krb5.ini
 - For Linux: /etc/krb5.conf
 - For UNIX (AIX): /etc/krb5/krb5.conf
 - For z/OS: /krb5/krb5.conf
2. Create a new directory for each integration server. For example, on Windows, you might create the following directories:
 - *work_path*/krb5/server1 (to hold the copy of the Kerberos configuration file for integration server "server1")
 - *work_path*/krb5/server2 (to hold the copy of the Kerberos configuration file for integration server "server2")

where *work_path* is the machine-wide IBM Integration Bus working directory. By using this directory, you can manage the configuration of both the active and the standby integration nodes in a high availability deployment.

Note: To verify the machine-wide IBM Integration Bus working directory, enter the following command in a command console:

```
echo %MQSI_WORKPATH%
```

3. Add a copy of the default Kerberos configuration file to each of the new directories.
4. If you have an integration server that exposes one or more services, generate a keytab file for each service and place it in the directory for that integration server. For information about creating keytab files, see “Configuring IBM Integration Bus as a Kerberos secured service” on page 486.
5. Edit each of the Kerberos configuration files and add the path to the keytab file (if required) and the realm information. For example, on Windows, you might add the following entries:

```
default_keytab_name=work_path/krb5/server1/krb.keytab
default_realm=server1Realm
```

where *server1Realm* is the realm name for the integration server "server1". If you have to support multiple realms, you can also add entries for each realm into the [realms] section. The following code shows an example of the configuration for multiple realms:

```
[libdefaults]
    default_keytab_name = FILE:/krb/test.keytab
    default_realm = REALM2.MYCO.COM
#    default_realm = REALM1.MYCO.COM
    default_tkt_enctypes = rc4-hmac
    default_tgs_enctypes = rc4-hmac
```

```
[realms]
REALM1.MYCO.COM = {
    kdc = kdc1.myco.com:88
    admin_server = kdc.myco.ibm.com:749
    default_domain = myco.com
}
REALM2.MYCO.COM = {
    kdc = kdc2.myco.com:88
    default_domain = realm2.myco.com
}
```

```
[domain_realm]
# defines how to map from internet domain names to Kerberos realms
REALM2.MYCO.COM.MYCO.COM = REALM2.MYCO.COM
.myco.com = REALM1.MYCO.COM
.test.myco.com = REALM1.MYCO.COM
```

6. For each integration server, run the following command on your integration node:

```
mqsichangeproperties node -e server -n jvmSystemProperty -v "-Djava.security.krb5.conf=path"
```

where *node* is the name of your integration node, *server* is the name of your integration server, and *path* is the path to your Kerberos configuration file.

7. Remove the `java.security.krb5.conf` setting from **IBM_JAVA_OPTIONS** environment variable and restart your integration node.

Results

Results

You configured a separate Kerberos configuration file (and optional keytab files) for each integration server.

Related concepts:

“Kerberos-based WS-Security” on page 482

You can use Kerberos authentication with WS-Security either as a service or as a client.

Related reference:

“Kerberos token capabilities for authentication and authorization” on page 528

This topic describes the broker Web services capability for authentication, authorization, or both using a Kerberos token.

Policy sets

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

A *policy set* is a container for the WS-Security and WS-RM policy types.

A policy set *binding* is associated with a policy set and contains information that is specific to the environment and platform, such as information about keys.

Use policy sets and bindings to define the following items for both request and response SOAP messages:

- Authentication for the following tokens:
 - Username tokens (requires a security profile to specify the external security provider)
 - X.509 certificates (requires the broker keystore and truststore, or a security profile to specify the external security provider)
 - SAML assertions, using SAML 1.1 or 2.0 pass-through (requires a security profile to specify the external security provider)
 - LTPA tokens, using LTPA pass-through (requires a security profile to specify the external security provider)
- Asymmetric encryption (confidentiality) using X.509 certificates (requires the broker keystore and truststore)
- Symmetric encryption (confidentiality) using Kerberos tokens (requires the host to be configured for Kerberos)
- Asymmetric signature (integrity) (requires the broker keystore and truststore)

Either the whole SOAP message body, or specific parts of the SOAP message header and body can be encrypted and signed.

You administer policy sets and bindings from IBM Integration Explorer, which can add, delete, display and edit policy sets and bindings. Any changes to policy sets or bindings in the toolkit are saved directly to the associated broker. You must stop and then restart the message flow for the new configuration information to take effect.

You can also export and import policy sets and bindings from a broker.

- “Exporting a policy set and policy binding” on page 503
- “Importing a policy set and policy set binding” on page 452

Policy sets and their associated bindings must be saved and restored together.

Policy sets are associated with a message flow, a node or both in the Broker Archive editor. For convenience, you can specify settings for *provider* and *consumer* at the message flow level. The provider setting applies to all SOAPInput and SOAPReply nodes in the message flow. The consumer setting applies to all SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes. Individual policy set and binding assignments can be applied at the node level in the Broker Archive editor, and these take precedence over the flow-level provider and consumer settings. The default setting is *none*, meaning that no policy set and bindings are to be used.

Several nodes in the same message flow can refer to the same policy set and bindings. It is the responsibility of the administrator to ensure that the required policy sets are available to the broker at run time. An error is reported if the broker cannot find the associated policy set or bindings.

The rest of this topic describes some of the terms that you will meet when configuring policy sets and bindings.

Default policy set and bindings

When a broker is created, a default policy set and bindings are created called WSS10Default. This default contains a limited security policy which specifies that a Username token is present in request messages (inbound) to SOAPInput nodes in the associated message flow. A default WS-RM policy set called WSRMDefault is also created.

The default policy set binding refers to the default WSS10Default policy set, but not to WSRMDefault. They are not editable.

Consumer and provider nodes

Nodes are either consumers or providers.

Consumer nodes

- SOAPRequest
- SOAPAsyncRequest
- SOAPAsyncResponse

Provider nodes

- SOAPInput
- SOAPReply

Request and response

Request and response is a message exchange pattern (MEP). It describes a client that sends a SOAP Request message to a Web services server, which in turn sends a Response SOAP message back to the client. The Request message is always the SOAP message from the client to the server, and the Response message is always the SOAP message reply from server to the client. The following table describes this pattern in relation to the IBM Integration Bus SOAP nodes:

Node	Broker viewpoint	Request	Response
SOAPInput	SOAP message inbound	Inbound message	Not applicable

Node	Broker viewpoint	Request	Response
SOAPReply	SOAP message outbound	Not applicable	Outbound message
SOAPRequest	SOAP message outbound followed by a SOAP message inbound	Outbound message	Inbound message
SOAPAsyncRequest	SOAP message outbound	Outbound message	Not applicable
SOAPAsyncResponse	SOAP message inbound	Not applicable	Inbound message

Initiator and recipient

Initiator and recipient are roles defined in the exchange of SOAP messages.

Initiator

The role that sends the initial message in a message exchange.

Recipient

The targeted role to process the initial message in a message exchange.

The following table describes these roles in relation to the SOAP nodes:

Node	Broker viewpoint	Initiator	Recipient
SOAPInput	SOAP message inbound	External client sending SOAP message to the broker.	SOAPInput node
SOAPReply	SOAP message outbound	External client that sent the original SOAP message to the broker.	SOAPReply node
SOAPRequest (outbound)	SOAP message outbound followed by a SOAP message inbound	SOAPRequest node	External provider receiving the SOAP message
SOAPRequest (inbound)	SOAP message outbound followed by a SOAP message inbound	SOAPRequest node	External provider receiving the SOAP message
SOAPAsyncRequest	SOAP message outbound	SOAPAsyncRequest node	External provider receiving the SOAP message
SOAPAsyncResponse	SOAP message inbound	SOAPAsyncRequest node	External provider receiving the SOAP message

SOAPInput and SOAPReply nodes

In this diagram, the broker acts as recipient. A SOAPInput node receives a message from a client (initiator). A SOAPReply node replies. Inbound and outbound messages are signed and encrypted.

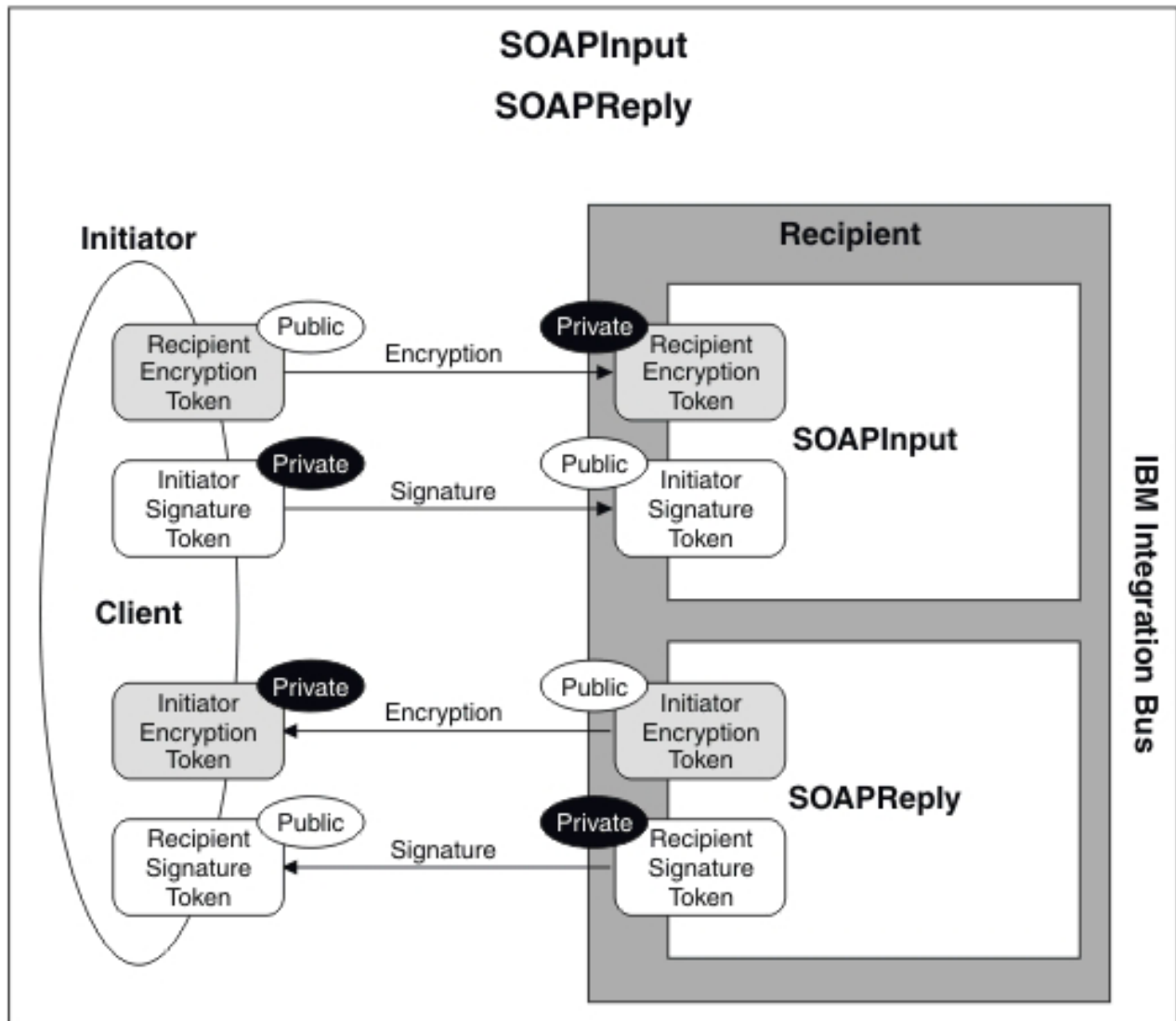


Figure 9. This graphic shows the interactions between message broker and client when the SOAP Input and SOAP Reply nodes are used.

In the request:

- The initiator uses the broker's public encryption token to encrypt the message, and its own private signature token to sign it.
- The broker uses its own private encryption token to decrypt the message, and the initiator's public signature token to verify the signature.

In the response:

- The broker uses the initiator's public encryption token to encrypt the message, and its own private signature token to sign the message.
- The initiator uses its own private encryption token to decrypt the message, and the broker's public signature token to verify the signature.

SOAPRequest node

This diagram shows the broker acting as an initiator. It uses the SOAPRequest node to make a synchronous request to an external provider (the recipient).

Inbound and outbound messages are signed and encrypted. Use of tokens is similar to the example of the asynchronous SOAP nodes, shown earlier.

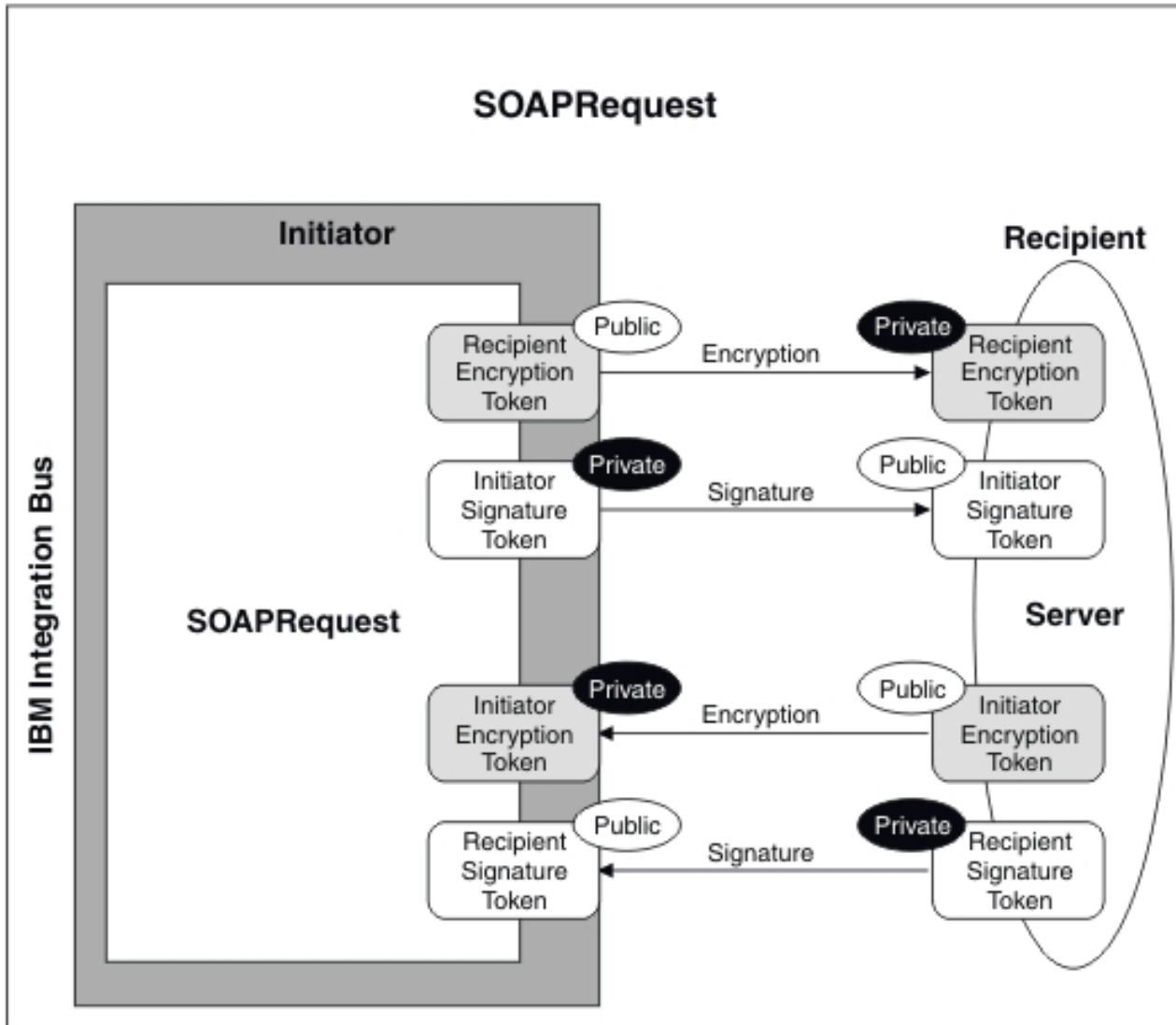


Figure 10. This graphic shows the interactions between message broker and server when the SOAP Request node is used.

In the request:

- The broker uses the recipient's public encryption token to encrypt the message, and its own private signature token to sign the message.
- The recipient uses its own private encryption token to decrypt the message, and the broker's public signature token to verify the signature.

In the response:

- The recipient uses the broker's public encryption token to encrypt the message, and its own private signature token to sign the message.
- The broker uses its own private encryption token to decrypt the message, and the initiator's public signature token to verify the signature.

Asynchronous SOAP nodes

This diagram shows the broker acting as an initiator. It uses the asynchronous SOAP nodes to make a request to an external provider (the recipient). Inbound and outbound messages are signed and encrypted.

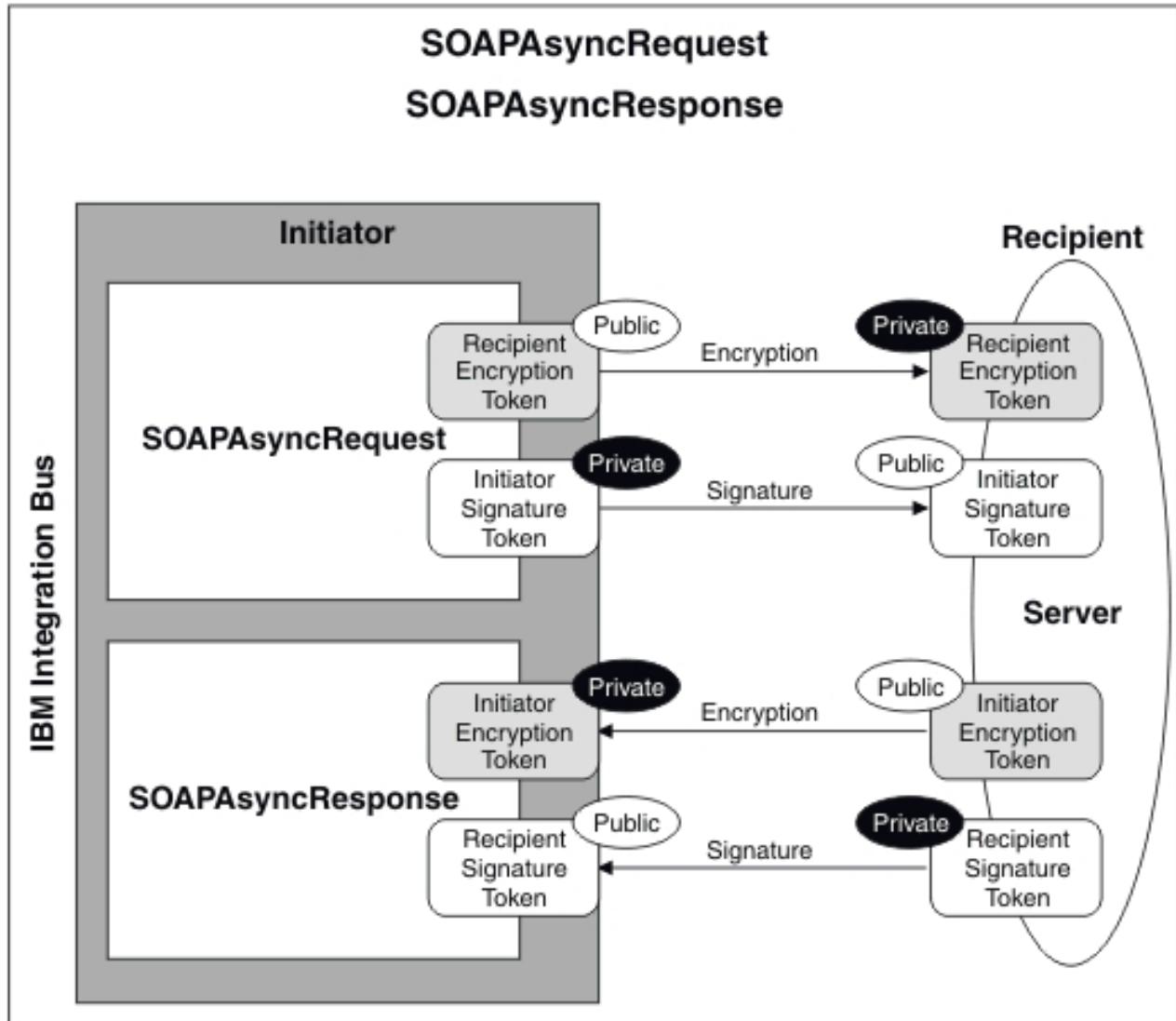


Figure 11. This graphic shows the interactions between message broker and server when the asynchronous SOAP nodes are used.

In the request:

- The broker uses the recipient's public encryption token to encrypt the message, and its own private signature token to sign the message.
- The recipient uses its own private encryption token to decrypt the message, and the broker's public signature token to verify the signature.

In the response:

- The recipient uses the broker's public encryption token to encrypt the message, and its own private signature token to sign the message.

- The broker uses its own private encryption token to decrypt the message, and the initiator's public signature token to verify the signature.

Related concepts:

“WS-Security” on page 473


Web Services Security (WS-Security) describes enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication. WS-Security mechanisms can be used to accommodate a wide variety of security models and encryption technologies.

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“Web Services Reliable Messaging” on page 469

IBM Integration Bus supports WS-RM (Web Services Reliable Messaging), which allows two systems to reliably exchange messages with each other.

 Public key cryptography

All encryption systems rely on the concept of a key. A key is the basis for a transformation, usually mathematical, of an ordinary message into an unreadable message. For centuries, most encryption systems have relied on private key encryption. Public key encryption is the only challenge to private key encryption that has appeared in the last 30 years.

 Digital certificates

Certificates provide a way of authenticating users. Instead of requiring each participant in an application to authenticate every user, third-party authentication relies on the use of digital certificates.

 Digital signatures

A digital signature is a number that is attached to a document. For example, in an authentication system that uses public-key encryption, digital signatures are used to sign certificates.

Related tasks:

“Viewing and setting keystore and truststore runtime properties at broker level” on page 497

Configure the message broker to refer to a keystore, a truststore, or both, before deploying any message flows that require policy set or bindings for signature, encryption, or X.509 Authentication.

“Viewing and setting keystore and truststore runtime properties at integration server level” on page 499

Configure an integration server to refer to a keystore, a truststore, or both, before deploying any message flows that require policy set or bindings for signature, encryption, or X.509 Authentication.

“Implementing WS-Security” on page 477

Configure authentication, XML encryption, XML signature, and message expiration by using the Policy Sets and Policy Set Bindings editor.

“Associating policy sets and bindings with message flows and nodes” on page 502

Use the Broker Archive editor to associate policy sets and bindings with message flows and nodes, so that they are available to the broker at run time.

“Exporting a policy set and policy binding” on page 503

Use the **mqsiexportproperties** command to export a policy set and associated binding to a file.

“Importing a policy set and policy set binding” on page 452

Use the **mqsichangeproperties** command to import a policy set and associated

binding.

Related reference:



Policy Sets and Policy Set Bindings editor

The Policy Sets and Policy Set Bindings editor is the default editor for editing, saving, importing, and exporting a policy set or binding. You can define all the WS-Security policies for a single node, or a set of nodes. The policy set can be associated by the administrator with either a message flow or a node.

Viewing and setting keystore and truststore runtime properties at broker level:

Configure the message broker to refer to a keystore, a truststore, or both, before deploying any message flows that require policy set or bindings for signature, encryption, or X.509 Authentication.

About this task

Keystores and truststores are both keystores. They differ only in the way they are used.

- Put all private keys and public key certificates (PKC) in the keystore.

- Put all trusted root certificate authority (CA) certificates in the truststore. These certificates are used to establish the trust of any inbound public key certificates.

The only supported type of store is Java keystore (JKS).

Each instance of a broker can be configured to refer to one keystore and one truststore.

The following properties of the broker registry component must be defined correctly for policy sets and bindings:

brokerKeystoreFile

The directory and file location of the keystore.

brokerTruststoreFile

The directory and file location of the truststore.

If you want to check what security properties you have set for a broker, use the **mqsireportdbparms** command.

Listing existing broker registry entries:

About this task

To display all broker registry values, run the command:

```
mqsireportproperties broker_name -o BrokerRegistry -a
```

This returns entries like these:

```
BrokerRegistry=' '  
  uuid='BrokerRegistry'  
  brokerKeystoreType='JKS'  
  brokerKeystoreFile=' '  
  brokerKeystorePass='brokerKeystore::password'  
  brokerTruststoreType='JKS'  
  brokerTruststoreFile=' '  
  brokerTruststorePass='brokerTruststore::password'  
  httpConnectorPortRange=' '  
  httpsConnectorPortRange=' '
```

Updating the broker reference to a keystore:

About this task

To update the broker reference to a keystore, use the following command:

```
mqsichangeproperties broker_name -o BrokerRegistry
  -n brokerKeystoreFile
  -v c:\keystore\server.keystore
```

Where c:\keystore\server.keystore is the keystore to be referenced.

Updating the broker reference to a truststore:

About this task

To update the broker reference to a truststore, use the following command:

```
mqsichangeproperties broker_name -o BrokerRegistry
  -n brokerTruststoreFile
  -v c:\truststore\server.truststore
```

Where c:\truststore\server.truststore is the truststore to be referenced.

Updating the broker with the keystore password:

About this task

Keystores and truststores normally require passwords for access. Use the **mqsisetdbparms** command to add these passwords to the broker runtime component. If you want to check a password that you have set, use the **mqsireportdbparms** command.

```
mqsisetdbparms broker_name
  -n brokerKeystore::password
  -u temp -p pa55word
```

The user ID, which can be any value, is not required to access the keystore.

Updating the broker with the truststore password:

About this task

To update the broker with the truststore password, use the following command:

```
mqsisetdbparms broker_name
  -n brokerTruststore::password
  -u temp -p pa55word
```

The user ID, which can be any value, is not required to access the keystore.

Updating the broker with a private key password:

About this task

Private keys in the keystore might have their own individual passwords. These can be configured based on the alias name that is specified for the key in the Policy sets and bindings editor. If a key password based on the alias is not found, the keystore password is used. The following command updates the broker with the private key password for the key whose alias is *encKey*.

```
mqsisetdbparms broker_name
  -n brokerTruststore::keypass::encKey
  -u temp -p pa55word
```

The user ID, which can be any value, is not required to access the keystore.

Related concepts:

“Policy sets” on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

Related tasks:

“Implementing WS-Security” on page 477

Configure authentication, XML encryption, XML signature, and message expiration by using the Policy Sets and Policy Set Bindings editor.

“Associating policy sets and bindings with message flows and nodes” on page 502

Use the Broker Archive editor to associate policy sets and bindings with message flows and nodes, so that they are available to the broker at run time.

“Viewing and setting keystore and truststore runtime properties at integration server level”

Configure an integration server to refer to a keystore, a truststore, or both, before deploying any message flows that require policy set or bindings for signature, encryption, or X.509 Authentication.

Related reference:



Policy Sets and Policy Set Bindings editor

The Policy Sets and Policy Set Bindings editor is the default editor for editing, saving, importing, and exporting a policy set or binding. You can define all the WS-Security policies for a single node, or a set of nodes. The policy set can be associated by the administrator with either a message flow or a node.



mqsisetdbparms command

Use the **mqsisetdbparms** command to associate a specific user ID and password (or SSH identity file) with one or more resources that are accessed by the broker.



mqsireportdbparms command

Use the **mqsireportdbparms** command to list all parameters that are set for a specific broker.

Viewing and setting keystore and truststore runtime properties at integration server level:

Configure an integration server to refer to a keystore, a truststore, or both, before deploying any message flows that require policy set or bindings for signature, encryption, or X.509 Authentication.

About this task

An integration server is a named grouping of message flows that have been assigned to a broker. The broker enforces a degree of isolation between message flows in distinct integration servers by ensuring that they run in separate address spaces, or as unique processes. For more information about integration servers, see Integration servers.

Integration server keystore and truststore runtime property values override equivalent property values on the broker, if any are set.

Keystores can contain two kinds of entries: key entries and trusted certificate entries. If a keystore is used to contain trusted certificates, it is typically referred to as a truststore. IBM Integration Bus can refer to a keystore and a truststore per integration server. When the broker is encrypting or decrypting, it uses entries in

its keystore; if the broker is verifying a signature or performing X.509 authentication, it uses entries in its truststore.

The following sample demonstrates the use of viewing and setting keystore and truststore runtime properties at integration server level:

- Address Book

You can view information about samples only when you use the information center that is integrated with the IBM Integration Toolkit or the online information center. You can run samples only when you use the information center that is integrated with the IBM Integration Toolkit.

Displaying integration server level properties:

About this task

To display integration server level properties, run the command:

```
mqsireportproperties broker_name -o ComIbmJVMMManager -a -e integration_server
```

Updating the integration server reference to a keystore:

About this task

To update the broker reference to a keystore at an integration server level, use the following command:

```
mqschangeproperties broker_name -e integration_server -o ComIbmJVMMManager  
-n keystoreFile  
-v c:\keystore\server.keystore,JKS
```

where c:\keystore\server.keystore,JKS is a Java keystore (JKS).

Updating the integration server reference to a truststore:

About this task

To update the broker reference to a truststore at an integration server level, use the following command:

```
mqschangeproperties broker_name -e integration_server -o ComIbmJVMMManager  
-n truststoreFile  
-v c:\truststore\server.truststore
```

where c:\truststore\server.truststore is the truststore to be referenced.

Updating the keystore and truststore passwords:

About this task

The commands used to update the keystore and truststore passwords at integration server level are the same as those used when setting keystore and truststore runtime properties at broker level.

- To update the broker with the keystore password; see “Updating the broker with the keystore password” on page 498.
- To update the broker with the truststore password; see “Updating the broker with the truststore password” on page 498.
- To update the broker with a private key password; see “Updating the broker with a private key password” on page 498.

To use the default broker password for the keystore, the `keystorePass` parameter must be blank, or it must be set to `brokerKeystore::password`. To use a password other than the default broker password, use the following commands:

```
mqsischangeproperties broker_name -e integration_server -o ComIbmJVMManger -n keystorePass -v integration_server::keystorePass
```

```
mqsisetdbparms broker_name -n integration_server::keystorePass -u na -p password
```

To use the default broker password for the truststore, the `truststorePass` parameter must be blank, or it must be set to `brokerTruststore::password`. To use a password other than the default broker password, use the following commands:

```
mqsischangeproperties broker_name -e integration_server -o ComIbmJVMManger -n truststorePass -v integration_server::truststorePass
```

```
mqsisetdbparms broker_name -n integration_server::truststorePass -u na -p password
```

Adding new certificates to a keystore or truststore:

About this task

If you add new certificates to a keystore or truststore, to ensure that the new certificates are picked up, you must reload the Java virtual machine (JVM). You can reload the JVM by restarting the integration server.

Related concepts:

“Policy sets” on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

Related tasks:

“Implementing WS-Security” on page 477

Configure authentication, XML encryption, XML signature, and message expiration by using the Policy Sets and Policy Set Bindings editor.

“Associating policy sets and bindings with message flows and nodes” on page 502

Use the Broker Archive editor to associate policy sets and bindings with message flows and nodes, so that they are available to the broker at run time.

“Viewing and setting keystore and truststore runtime properties at broker level” on page 497

Configure the message broker to refer to a keystore, a truststore, or both, before deploying any message flows that require policy set or bindings for signature, encryption, or X.509 Authentication.



Stopping an integration server using the IBM Integration Toolkit or IBM Integration Explorer

You can stop all the message flows in an integration server by using the IBM Integration Toolkit or IBM Integration Explorer.



Starting an integration server using the IBM Integration Toolkit or IBM Integration Explorer

You can start an integration server and all the deployed message flows in an integration server by using the IBM Integration Toolkit or IBM Integration Explorer.

Related reference:



Policy Sets and Policy Set Bindings editor

The Policy Sets and Policy Set Bindings editor is the default editor for editing, saving, importing, and exporting a policy set or binding. You can define all the WS-Security policies for a single node, or a set of nodes. The policy set can be

associated by the administrator with either a message flow or a node.

Associating policy sets and bindings with message flows and nodes:

Use the Broker Archive editor to associate policy sets and bindings with message flows and nodes, so that they are available to the broker at run time.

Before you begin

Before you start

Use the Policy Sets and Policy Set Bindings editor to create and configure policy sets and bindings.

About this task

Associations can be made between policy sets and message flow, or specific nodes. Associations made with a flow apply to all nodes described in the Policy Set and Bindings file. Associations at the flow level are defined as being either for consumer or provider nodes.

An association at the node level overrides any association made at the flow level. You do not enter information about consumer or provider for an association at node level.

Procedure

1. In the IBM Integration Toolkit or IBM Integration Explorer, open the BAR file in the Broker Archive editor.
2. Click the **Manage and Configure** tab.
3. Click the message flow or node that you want to associate with a policy set and binding. The properties that you can configure for the message flow or for the node are displayed in the **Properties** view.
4. If you are configuring a message flow, enter values in the following fields in the **Properties** view, as appropriate:
 - Provider Policy Set Bindings**
 - Provider Policy Set**
 - Consumer Policy Set Bindings**
 - Consumer Policy Set**
5. If you are configuring a node, enter values in the following fields in the **Properties** view:
 - Policy Set**
 - Policy Set Bindings**

Example

What to do next

For new associations to take effect, the BAR file must be redeployed and the message flows stopped and restarted.

Related concepts:

"Policy sets" on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply,

SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

Related tasks:

“Viewing and setting keystore and truststore runtime properties at broker level” on page 497

Configure the message broker to refer to a keystore, a truststore, or both, before deploying any message flows that require policy set or bindings for signature, encryption, or X.509 Authentication.

“Viewing and setting keystore and truststore runtime properties at integration server level” on page 499

Configure an integration server to refer to a keystore, a truststore, or both, before deploying any message flows that require policy set or bindings for signature, encryption, or X.509 Authentication.

“Implementing WS-Security” on page 477

Configure authentication, XML encryption, XML signature, and message expiration by using the Policy Sets and Policy Set Bindings editor.

Related reference:



Policy Sets and Policy Set Bindings editor

The Policy Sets and Policy Set Bindings editor is the default editor for editing, saving, importing, and exporting a policy set or binding. You can define all the WS-Security policies for a single node, or a set of nodes. The policy set can be associated by the administrator with either a message flow or a node.

Exporting a policy set and policy binding:

Use the `mqsiexportproperties` command to export a policy set and associated binding to a file.

About this task

This topic shows how to export policy set *myPolicySet* from broker *myBroker* to a file called *myPolicySet.xml*. The associated binding is *myPolicySetBinding*, which you export to *myPolicySetBinding.xml*.

Procedure

1. Export the policy set to a file:

```
mqsiexportproperties myBroker -c PolicySets -o myPolicySet -n ws-security -p myPolicySet.xml
```

2. Export the policy set binding to a file:

```
mqsiexportproperties myBroker -c PolicySetBindings -o myPolicySetBinding -n ws-security -p myPolicySetBinding.xml
```

What to do next

Make a note of the policy set that is associated to the binding; you will need this information when you import the policy set and binding. This command displays the policy set associated with a binding:

```
mqsiexportproperties myBroker -c PolicySetBindings -o myPolicySetBinding -n associatedPolicySet
```

This displays:

```
PolicySetBindings myPolicySetBinding associatedPolicySet='myPolicySet'  
BIP8071I: Successful command completion.
```

Note: To export a policy set with more than one policy, you must run the **mqsireportproperties** command once for each policy. For example, a policy set might contain a WS-Security policy and a WS-RM policy.

Related tasks:

“Importing a policy set and policy set binding” on page 452

Use the **mqsichangeproperties** command to import a policy set and associated binding.

Related reference:



mqsireportproperties command

Use the **mqsireportproperties** command to display properties that relate to a broker, an integration server, or a configurable service.

Importing a policy set and policy set binding:

Use the **mqsichangeproperties** command to import a policy set and associated binding.

About this task

This topic shows how to import policy set *myPolicySet* to broker *myBroker* from a file called *myPolicySet.xml*. The associated binding is *myPolicySetBinding*, which you import from *myPolicySetBinding.xml*.

Procedure

1. Create a configurable service for the policy set, if one does not already exist.
`mqsicreateconfigurableservice myBroker -c PolicySets -o myPolicySet`
BIP8071I: Successful command completion.
2. Create a configurable service for the policy set binding, if one does not already exist.
`mqsicreateconfigurableservice myBroker -c PolicySetBindings -o myPolicySetBinding`
BIP8071I: Successful command completion.
3. Import the policy set.
`mqsichangeproperties myBroker -c PolicySets -o myPolicySet -n ws-security -p myPolicySet.xml`
4. Import the policy set binding.
`mqsichangeproperties myBroker -c PolicySetBindings -o myPolicySetBinding -n ws-security -p myPolicySetBinding.xml`
5. Change the value of the associatedPolicySet attribute. Set it to the name of the policy set with which this policy set binding was originally associated.
`mqsichangeproperties myBroker -c PolicySetBindings -o myPolicySetBinding -n associatedPolicySet -v myPolicySet`
6. You must stop and start the integration server for the changes of property values to take effect.

Related concepts:

“Policy sets” on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

Related tasks:

“Exporting a policy set and policy binding” on page 503

Use the **mqsireportproperties** command to export a policy set and associated binding to a file.

Related reference:



mqsichangeproperties command

Use the **mqsichangeproperties** command to modify broker properties and properties of broker resources.



mqsicreateconfigurable-service command

Use the **mqsicreateconfigurable-service** command to create an object name for a broker external resource.

Message flow security and security profiles

IBM Integration Bus provides a security manager for implementing message flow security, so that end-to-end processing of a message through a message flow is secured based on an identity carried in that message instance.

For details of the supported external providers and the operation of the message flow security manager, see Message flow security overview. For information about the token types that are supported by the SOAP nodes and by external security providers, see Identity.

When the message flow is a Web service implemented by using SOAP nodes and the identity is to be taken from the “WS-Security” on page 473 SOAP headers, the SOAP nodes are the Policy Enforcement Point (PEP) and the external provider defined by the Security profiles is the Policy Decision Point (PDP).

The following configuration is required to implement message flow security based on an identity carried in WS_Security tokens.

- “Policy sets” on page 490 define the type of tokens used for the identity.
 - To work with a Username and Password identity, configure the policy and binding for Username token “Authentication” on page 478.
 - To work with a X.509 Certificate identity, configure the policy and binding for X.509 certificate token “Authentication” on page 478.

In the Policy Set Binding, set the X.509 certificate Authentication Token certificates mode to Trust Any. You set it this way (and not to Trust Store) so that the certificate is passed to the security provider defined by the Security Profile. Setting it to Trust Store will cause the certificate to be validated in the local Broker Trust Store. For more details, see Policy Sets and Policy Set Bindings editor: Authentication and Protection Tokens panel.
 - To work with a SAML assertion token, configure the policy and binding for SAML token “Authentication” on page 478.
- The message flow security operation and external provider are defined by the Security profiles

As an alternative to message flow security and an external PDP, the broker's truststore can be used as a local PDP for X.509 certificate authentication. For WS-Security signing and encryption using only the local broker capability, you must configure the broker's truststore. For details, see “Viewing and setting keystore and truststore runtime properties at broker level” on page 497, or “Viewing and setting keystore and truststore runtime properties at integration server level” on page 499.

Kerberos based WS-Security is supported in the SOAP nodes. When you use Kerberos for security, the SOAP node's WS-Security processing links directly with the host's Kerberos infrastructure. The broker host must be configured for

Kerberos, providing a `krb.conf` file to define the Kerberos Key Distribution Center (KDC) and default realm. A Kerberos keytab file must also be configured. For more information about configuring Kerberos, see your host's Kerberos documentation.

To work with Kerberos WS-Security in SOAP nodes, create a policy set and bindings specifying Kerberos symmetric encryption tokens on the Message Level Protection panel; see Policy Sets and Policy Set Bindings editor: Message Level Protection panel. Also configure the required settings on the Kerberos settings panel, as described in Policy Sets and Policy Set Bindings editor: Kerberos settings panel, and then associate this policy set and bindings with the SOAP node. You can also associate SOAP nodes with a security profile that sets only propagation, so that Kerberos can be used to:

- Extract the service principal as a Username token from SOAP input nodes
- Propagate the Kerberos Key Distribution Center (KDC) credentials as a Username and password to SOAP request nodes.

Related concepts:



Message flow security overview

IBM Integration Bus provides a security manager, which enables you to control access to individual messages in a message flow, using the identity of the message.

“WS-Security” on page 473

Web Services Security (WS-Security) describes enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication. WS-Security mechanisms can be used to accommodate a wide variety of security models and encryption technologies.



Security profiles

A security profile defines the security operations that are to be performed in a message flow at SecurityPEP nodes and security enabled input and output nodes.

“Policy sets” on page 490

Policy sets and bindings define and configure your WS-Security and WS-RM requirements, supported by IBM Integration Bus, for the SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

“Username token capabilities” on page 508

This topic describes WS-Security username token capabilities of the broker.

“X.509 certificate token capabilities” on page 512

Various WS-Services Security X.509 certificate token profile standards are supported by IBM Integration Bus.

Related tasks:



Setting up message flow security

Set up security on a message flow to control access based on the identity of a message passing through the message flow.



Integration Bus server security

You must consider several security aspects when you are setting up brokers running on Windows, Linux, z/OS, or UNIX platforms.

Creating a security profile

You can create a security profile for use with Lightweight Directory Access Protocol (LDAP) or a WS-Trust V1.3 compliant security token server (STS), such as Tivoli Federated Identity Manager (TFIM) V6.2. Support is also provided for TFIM V6.1, for compatibility with previous versions of IBM Integration Bus. You can create the security profile by using either the `mqsicreateconfigurableservice` command or an editor in the IBM Integration Explorer.

“Viewing and setting keystore and truststore runtime properties at broker level” on page 497

Configure the message broker to refer to a keystore, a truststore, or both, before deploying any message flows that require policy set or bindings for signature, encryption, or X.509 Authentication.

Related reference:

Policy Sets and Policy Set Bindings editor: Authentication and Protection Tokens panel

Use this panel, which is in the Policy Set Bindings section of the editor, to further configure any X.509 authentication tokens defined in the associated policy set.

WS-Security capabilities

Web service security capabilities are supported by the broker.

Web service security mechanisms are defined by OASIS standards. See OASIS Standard for WS-Security Specification

For information about the token profile standards, see:

- OASIS Web Services Security Username Token Profile
- OASIS Web Services Security X.509 Certificate Token Profile
- OASIS Web Services Security Kerberos Token Profile

SAML pass-through support is provided, which enables interoperability with WS-Security SAML profiles, without performing subject confirmation processing. This means that it does not provide validation of the trust relationship between the SAML subject and message content signatures. For information about the SAML token profile standards, see:

- OASIS Web Services Security SAML Token Profile 1.1
- SAML V2 Specification

LTPA pass-through support is also provided, which enables LTPA binary tokens to be passed to an external security token server (STS) for processing.

For more information about using the token profiles, see the following topics:

- “Username token capabilities” on page 508
- “X.509 certificate token capabilities” on page 512
- “SAML token capabilities” on page 521
- “Kerberos token capabilities” on page 526
- “LTPA token capabilities” on page 531

Related tasks:

Setting up message flow security

Set up security on a message flow to control access based on the identity of a message passing through the message flow.

Integration Bus server security

You must consider several security aspects when you are setting up brokers running on Windows, Linux, z/OS, or UNIX platforms.

OASIS Standard for WS-Security Specification

OASIS Web Services Security Username Token Profile

OASIS Web Services Security X.509 Certificate Token Profile

Username token capabilities:

This topic describes WS-Security username token capabilities of the broker.

For details of using WS-Security username token, see the following capabilities:

- “Username token capabilities for encryption, decryption, signing, and verifying”
- “Username token capabilities for authentication and authorization” on page 509
- “Username token capabilities for identity mapping” on page 510
- “Username token capabilities for extraction and propagation” on page 511

Related tasks:

Setting up message flow security

Set up security on a message flow to control access based on the identity of a message passing through the message flow.

Integration Bus server security

You must consider several security aspects when you are setting up brokers running on Windows, Linux, z/OS, or UNIX platforms.

Username token capabilities for encryption, decryption, signing, and verifying:

For Web services, you cannot complete encryption, decryption, signing, and verification by using username tokens.

The username token is not applicable, or supported, for the following in any configuration or direction:

- Encryption
- Decryption
- Signing
- Verification

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“Username token capabilities”

This topic describes WS-Security username token capabilities of the broker.

Related reference:

“Username token capabilities for authentication and authorization” on page 509

For Web services, you can complete authentication and authorization by using a Web Services Security username token.

“Username token capabilities for identity mapping” on page 510

For Web services, you can map an identity by using a username token.

“Username token capabilities for extraction and propagation” on page 511

This topic describes broker capability for extraction, propagation, or both using a

username token in Web services.

Username token capabilities for authentication and authorization:

For Web services, you can complete authentication and authorization by using a Web Services Security username token.

For authentication, the Web Services Security username token must include both the username and the optional password.

The Web Services Security username token “Authentication” on page 476 and Authorization is supported only in the following configuration:

Capability

- Authenticate
- Authorize

Policy Enforcement Point (PEP) and direction

- In (provider)
 - SOAPInput node

Configured with a security policy and binding that defines that a Web Services Security username token is present for authentication; see “Authentication” on page 478. You can use the default policy and binding WSS10Default; see “Default policy set and bindings” on page 491.

Configured with a security profile defining the Policy Decision Point (PDP); see the PDP section that follows.

Trust Store or PDP

- LDAP
 - Configured by using an LDAP security profile specifying authentication, authorization, or both; see Creating a security profile for LDAP. For authentication, both a username and password are required.
- WS-Trust v1.3 STS
 - Configured by using a WS-Trust v1.3 STS security profile specifying authentication, authorization or both; see Creating a security profile for WS-Trust V1.3 (TFIM V6.2). For authentication, both a username and password are required.
- TFIM V6.1
 - Configured by using a TFIM security profile specifying authentication, authorization or both; see Creating a security profile for TFIM V6.1. For authentication, both a username and password are required.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“Username token capabilities” on page 508

This topic describes WS-Security username token capabilities of the broker.



Authorization

Authorization is the process of verifying that an identity token has permission to access a message flow.

Related reference:

“Username token capabilities for encryption, decryption, signing, and verifying” on page 508

For Web services, you cannot complete encryption, decryption, signing, and verification by using username tokens.

“Username token capabilities for identity mapping”

For Web services, you can map an identity by using a username token.

“Username token capabilities for extraction and propagation” on page 511

This topic describes broker capability for extraction, propagation, or both using a username token in Web services.

Username token capabilities for identity mapping:

For Web services, you can map an identity by using a username token.

Identity mapping from a username identity token to a mapped username identity token is supported only in the following configurations:

Capability

- Identity mapping

Policy Enforcement Point (PEP) and direction

- In (provider)

SOAPInput node

Configured with a security policy and binding that defines that a username token is present. You can use the default policy and binding WSS10Default; see “Default policy set and bindings” on page 491.

Configured with a security profile defining the external Policy Decision Point (PDP); see the PDP section that follows.

Trust store or PDP

- WS-Trust v1.3 STS

Configured by using a WS-Trust v1.3 STS security profile that specifies identity mapping; see Creating a security profile for WS-Trust V1.3 (TFIM V6.2).

- TFIM V6.1

Configured by using a TFIM security profile that specifies identity mapping; see Creating a security profile for TFIM V6.1.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“Username token capabilities” on page 508

This topic describes WS-Security username token capabilities of the broker.



Identity mapping

Identity mapping is the transformation of a security token from one format to another format, or the federation of an identity from one realm to an equivalent identity in another realm.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.

Related reference:

“Username token capabilities for encryption, decryption, signing, and verifying” on page 508

For Web services, you cannot complete encryption, decryption, signing, and verification by using username tokens.

“Username token capabilities for authentication and authorization” on page 509

For Web services, you can complete authentication and authorization by using a Web Services Security username token.

“Username token capabilities for extraction and propagation”

This topic describes broker capability for extraction, propagation, or both using a username token in Web services.

Username token capabilities for extraction and propagation:

This topic describes broker capability for extraction, propagation, or both using a username token in Web services.

The extraction of username into the Properties folder source Identity fields, is supported in the following configurations:

Capability

- Extraction

Policy Enforcement Point (PEP) and direction

- In (provider)

SOAPInput node

Configured with a security policy and binding which defines that a username token is present. You can use the default policy and binding WSS10Default; see “Default policy set and bindings” on page 491.

Configured with a security profile that defines propagation; see Creating a security profile

The propagation of a username token into the SOAP WS-Security header, from the token present in either the mapped or the source identity fields in the properties folder, is supported in the following configuration. See Identity.

Capability

- Propagate

Policy Enforcement Point (PEP) and direction

- Out (consumer)

SOAPRequest node

SOAPAsyncRequest node

Configured with a security profile that defines propagation; for example, Default Propagation. See Security profiles

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“Username token capabilities” on page 508

This topic describes WS-Security username token capabilities of the broker.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.



Identity and security token propagation

Identity and security token propagation enables the identity and security tokens (associated with each message) to be propagated throughout a message flow, and on to target applications through output or request nodes.

Related tasks:



Configuring the extraction of an identity or security token

You can configure the SecurityPEP node or security enabled input nodes to extract the identity or security token from a message and store it in the properties tree identity fields, enabling it to be processed throughout the message flow and propagated at output or request nodes.

Related reference:

“Username token capabilities for encryption, decryption, signing, and verifying” on page 508

For Web services, you cannot complete encryption, decryption, signing, and verification by using username tokens.

“Username token capabilities for authentication and authorization” on page 509

For Web services, you can complete authentication and authorization by using a Web Services Security username token.

“Username token capabilities for identity mapping” on page 510

For Web services, you can map an identity by using a username token.

X.509 certificate token capabilities:

Various WS-Services Security X.509 certificate token profile standards are supported by IBM Integration Bus.

For details of using the X.509 certificates, see the following capabilities:

- “X.509 certificate token capabilities for encryption” on page 513
- “X.509 certificate token capabilities for decryption” on page 514
- “X.509 certificate token capabilities for signing” on page 515
- “X.509 certificate token capabilities for verifying” on page 516
- “X.509 certificate token capabilities for authentication” on page 517
- “X.509 certificate token capabilities for authorization” on page 518
- “X.509 certificate token capabilities for identity mapping” on page 519
- “X.509 certificate token capabilities for extraction and propagation” on page 520

Related tasks:



Setting up message flow security

Set up security on a message flow to control access based on the identity of a message passing through the message flow.



Integration Bus server security

You must consider several security aspects when you are setting up brokers running on Windows, Linux, z/OS, or UNIX platforms.

X.509 certificate token capabilities for encryption:

For Web services, you can complete encryption by using an X.509 certificate token.

X.509 certificate token encryption for providing message “Confidentiality” on page 477 on outgoing SOAP messages from the broker is supported in the following configurations:

Capability

- Encrypt (by using a partner public key)

Policy Enforcement Point (PEP) and direction

- Out (consumer)

SOAPRequest node

SOAPAsyncRequest node

- Out (provider)

SOAPReply node

Configured with a policy set and binding defining the message “Confidentiality” on page 479.

Trust Store or Policy Decision Point (PDP)

- Broker Truststore; for more details, see “Viewing and setting keystore and truststore runtime properties at broker level” on page 497.

Encryption is not supported with external PDPs such as TFIM or LDAP.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“X.509 certificate token capabilities” on page 512

Various WS-Services Security X.509 certificate token profile standards are supported by IBM Integration Bus.

Related tasks:

“Implementing WS-Security” on page 477

Configure authentication, XML encryption, XML signature, and message expiration by using the Policy Sets and Policy Set Bindings editor.

Related reference:

“X.509 certificate token capabilities for decryption” on page 514

For Web services, you can complete decryption by using an X.509 certificate token.

“X.509 certificate token capabilities for signing” on page 515

For Web services, you can use an X.509 certificate token for signing.

“X.509 certificate token capabilities for verifying” on page 516

For Web services, you can verify a signing by using an X.509 certificate token profile.

“X.509 certificate token capabilities for authentication” on page 517

For Web services, you can complete authentication by using an X.509 certificate token.

“X.509 certificate token capabilities for authorization” on page 518

The X.509 certificate token is not supported for authorization in any configuration or direction.

“X.509 certificate token capabilities for identity mapping” on page 519
For Web services, you can map an identity by using an X.509 certificate token.
“X.509 certificate token capabilities for extraction and propagation” on page 520
This topic describes broker Web services capability for extraction and propagation X.509 certificate token.

X.509 certificate token capabilities for decryption:

For Web services, you can complete decryption by using an X.509 certificate token.

X.509 certificate token decryption for incoming SOAP message “Confidentiality” on page 477 is supported in the following configurations:

Capability

- Decrypt (by using a broker private key)

Policy Enforcement Point (PEP) and direction.

- In (provider)
 SOAPInput node
- In (consumer)
 SOAPRequest node
 SOAPAsyncResponse node

Configured with a policy set and binding defining the message “Confidentiality” on page 479.

Trust Store or Policy Decision Point (PDP).

- Broker Truststore; for details, see “Viewing and setting keystore and truststore runtime properties at broker level” on page 497.

Decryption is not supported with external PDPs such as TFIM or LDAP.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“X.509 certificate token capabilities” on page 512

Various WS-Services Security X.509 certificate token profile standards are supported by IBM Integration Bus.

Related tasks:

“Implementing WS-Security” on page 477

Configure authentication, XML encryption, XML signature, and message expiration by using the Policy Sets and Policy Set Bindings editor.

Related reference:

“X.509 certificate token capabilities for encryption” on page 513

For Web services, you can complete encryption by using an X.509 certificate token.

“X.509 certificate token capabilities for signing” on page 515

For Web services, you can use an X.509 certificate token for signing.

“X.509 certificate token capabilities for verifying” on page 516

For Web services, you can verify a signing by using an X.509 certificate token profile.

“X.509 certificate token capabilities for authentication” on page 517
For Web services, you can complete authentication by using an X.509 certificate token.

“X.509 certificate token capabilities for authorization” on page 518
The X.509 certificate token is not supported for authorization in any configuration or direction.

“X.509 certificate token capabilities for identity mapping” on page 519
For Web services, you can map an identity by using an X.509 certificate token.

“X.509 certificate token capabilities for extraction and propagation” on page 520
This topic describes broker Web services capability for extraction and propagation X.509 certificate token.

X.509 certificate token capabilities for signing:

For Web services, you can use an X.509 certificate token for signing.

X.509 certificate token signing for outgoing SOAP message “Integrity” on page 477 is supported in the following configurations:

Capability

- Sign (by using a broker private key)

Policy Enforcement Point (PEP) and direction

- Out (consumer)
 - SOAPRequest node
 - SOAPAsyncRequest node
- Out (provider)
 - SOAPReply node

Configured with a policy set and binding defining the message “Integrity” on page 480.

Trust Store or Policy Decision Point (PDP)

- Broker Truststore; for details, see “Viewing and setting keystore and truststore runtime properties at broker level” on page 497.

Signing is not supported with an external PDP such as TFIM or LDAP.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“X.509 certificate token capabilities” on page 512

Various WS-Services Security X.509 certificate token profile standards are supported by IBM Integration Bus.

Related tasks:

“Implementing WS-Security” on page 477

Configure authentication, XML encryption, XML signature, and message expiration by using the Policy Sets and Policy Set Bindings editor.

Related reference:

“X.509 certificate token capabilities for encryption” on page 513

For Web services, you can complete encryption by using an X.509 certificate token.

“X.509 certificate token capabilities for decryption” on page 514
For Web services, you can complete decryption by using an X.509 certificate token.

“X.509 certificate token capabilities for verifying”
For Web services, you can verify a signing by using an X.509 certificate token profile.

“X.509 certificate token capabilities for authentication” on page 517
For Web services, you can complete authentication by using an X.509 certificate token.

“X.509 certificate token capabilities for authorization” on page 518
The X.509 certificate token is not supported for authorization in any configuration or direction.

“X.509 certificate token capabilities for identity mapping” on page 519
For Web services, you can map an identity by using an X.509 certificate token.

“X.509 certificate token capabilities for extraction and propagation” on page 520
This topic describes broker Web services capability for extraction and propagation X.509 certificate token.

X.509 certificate token capabilities for verifying:

For Web services, you can verify a signing by using an X.509 certificate token profile.

X.509 certificate token verification of the “Integrity” on page 477 of a signed incoming SOAP message is supported in the following configurations:

Capability

- Verify signature (by using a partner public key)

Policy Enforcement Point (PEP) and direction

- In (provider)
 SOAPInput node
- In (consumer)
 SOAPRequest node
 SOAPAsyncResponse node

Configured with a policy set and binding defining the message “Integrity” on page 480

Trust Store or Policy Decision Point (PDP)

- Broker Trust store; for details, see “Viewing and setting keystore and truststore runtime properties at broker level” on page 497.

Signature verification is not supported with an external PDP, such as TFIM or LDAP.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“X.509 certificate token capabilities” on page 512

Various WS-Services Security X.509 certificate token profile standards are supported by IBM Integration Bus.

Related tasks:

“Implementing WS-Security” on page 477

Configure authentication, XML encryption, XML signature, and message expiration by using the Policy Sets and Policy Set Bindings editor.

Related reference:

“X.509 certificate token capabilities for encryption” on page 513

For Web services, you can complete encryption by using an X.509 certificate token.

“X.509 certificate token capabilities for decryption” on page 514

For Web services, you can complete decryption by using an X.509 certificate token.

“X.509 certificate token capabilities for signing” on page 515

For Web services, you can use an X.509 certificate token for signing.

“X.509 certificate token capabilities for authentication”

For Web services, you can complete authentication by using an X.509 certificate token.

“X.509 certificate token capabilities for authorization” on page 518

The X.509 certificate token is not supported for authorization in any configuration or direction.

“X.509 certificate token capabilities for identity mapping” on page 519

For Web services, you can map an identity by using an X.509 certificate token.

“X.509 certificate token capabilities for extraction and propagation” on page 520

This topic describes broker Web services capability for extraction and propagation X.509 certificate token.

X.509 certificate token capabilities for authentication:

For Web services, you can complete authentication by using an X.509 certificate token.

The X.509 certificate token “Authentication” on page 476 of an incoming SOAP message is supported in the following configurations:

Capability

- Authenticate

Policy Enforcement Point (PEP) and direction

- In (provider)

SOAPInput node

Configured with a policy set and binding defining the certificate “Authentication” on page 478.

Optionally configured with a security profile defining an external Policy Decision Point (PDP); see the PDP section that follows.

Trust Store or PDP

- Broker Trust store; for details, see “Viewing and setting keystore and truststore runtime properties at broker level” on page 497.

- WS-Trust v1.3 STS

Configured by using a WS-Trust v1.3 STS security profile specifying authentication; see Creating a security profile for WS-Trust V1.3 (TFIM V6.2).

- TFIM V6.1

Configured by using a TFIM security profile specifying authentication; for details, see Creating a security profile for TFIM V6.1.

Certificate authentication with an external LDAP PDP is not supported.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“X.509 certificate token capabilities” on page 512

Various WS-Services Security X.509 certificate token profile standards are supported by IBM Integration Bus.

Related tasks:

“Implementing WS-Security” on page 477

Configure authentication, XML encryption, XML signature, and message expiration by using the Policy Sets and Policy Set Bindings editor.

Related reference:

“X.509 certificate token capabilities for encryption” on page 513

For Web services, you can complete encryption by using an X.509 certificate token.

“X.509 certificate token capabilities for decryption” on page 514

For Web services, you can complete decryption by using an X.509 certificate token.

“X.509 certificate token capabilities for signing” on page 515

For Web services, you can use an X.509 certificate token for signing.

“X.509 certificate token capabilities for verifying” on page 516

For Web services, you can verify a signing by using an X.509 certificate token profile.

“X.509 certificate token capabilities for authorization”

The X.509 certificate token is not supported for authorization in any configuration or direction.

“X.509 certificate token capabilities for identity mapping” on page 519

For Web services, you can map an identity by using an X.509 certificate token.

“X.509 certificate token capabilities for extraction and propagation” on page 520

This topic describes broker Web services capability for extraction and propagation X.509 certificate token.

X.509 certificate token capabilities for authorization:

The X.509 certificate token is not supported for authorization in any configuration or direction.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.



Authorization

Authorization is the process of verifying that an identity token has permission to access a message flow.

“X.509 certificate token capabilities” on page 512

Various WS-Services Security X.509 certificate token profile standards are supported by IBM Integration Bus.

Related tasks:

“Implementing WS-Security” on page 477

Configure authentication, XML encryption, XML signature, and message expiration by using the Policy Sets and Policy Set Bindings editor.

Related reference:

“X.509 certificate token capabilities for encryption” on page 513
For Web services, you can complete encryption by using an X.509 certificate token.

“X.509 certificate token capabilities for decryption” on page 514
For Web services, you can complete decryption by using an X.509 certificate token.

“X.509 certificate token capabilities for signing” on page 515
For Web services, you can use an X.509 certificate token for signing.

“X.509 certificate token capabilities for verifying” on page 516
For Web services, you can verify a signing by using an X.509 certificate token profile.

“X.509 certificate token capabilities for authentication” on page 517
For Web services, you can complete authentication by using an X.509 certificate token.

“X.509 certificate token capabilities for identity mapping”
For Web services, you can map an identity by using an X.509 certificate token.

“X.509 certificate token capabilities for extraction and propagation” on page 520
This topic describes broker Web services capability for extraction and propagation X.509 certificate token.

X.509 certificate token capabilities for identity mapping:

For Web services, you can map an identity by using an X.509 certificate token.

The broker supports Identity mapping from an X.509 certificate token in an incoming SOAP message header to username tokens in the following configurations:

Capability

- Identity mapping

Policy Enforcement Point (PEP) and direction

- In (provider)
 - SOAPInput node
 - Configured with a policy set and binding defining the certificate “Authentication” on page 478.
 - Configured with a security profile defining an external Policy Decision Point (PDP); see the PDP section that follows.

Trust Store or PDP

- TFIM V6.1
 - Configured by using a TFIM security profile specifying identity mapping; for details, see *Creating a security profile for TFIM V6.1*.
- WS-Trust v1.3 STS (TFIM V6.2)
 - Configured by using a WS-Trust v1.3 STS security profile specifying identity mapping; for details, see *Creating a security profile for WS-Trust V1.3 (TFIM V6.2)*.

Identity mapping is not supported with LDAP, or at outbound nodes.

Username tokens only can be propagated.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

**Identity mapping**

Identity mapping is the transformation of a security token from one format to another format, or the federation of an identity from one realm to an equivalent identity in another realm.

“X.509 certificate token capabilities” on page 512

Various WS-Services Security X.509 certificate token profile standards are supported by IBM Integration Bus.

Related tasks:

“Implementing WS-Security” on page 477

Configure authentication, XML encryption, XML signature, and message expiration by using the Policy Sets and Policy Set Bindings editor.

Related reference:

“X.509 certificate token capabilities for encryption” on page 513

For Web services, you can complete encryption by using an X.509 certificate token.

“X.509 certificate token capabilities for decryption” on page 514

For Web services, you can complete decryption by using an X.509 certificate token.

“X.509 certificate token capabilities for signing” on page 515

For Web services, you can use an X.509 certificate token for signing.

“X.509 certificate token capabilities for verifying” on page 516

For Web services, you can verify a signing by using an X.509 certificate token profile.

“X.509 certificate token capabilities for authentication” on page 517

For Web services, you can complete authentication by using an X.509 certificate token.

“X.509 certificate token capabilities for authorization” on page 518

The X.509 certificate token is not supported for authorization in any configuration or direction.

“X.509 certificate token capabilities for extraction and propagation”

This topic describes broker Web services capability for extraction and propagation X.509 certificate token.

X.509 certificate token capabilities for extraction and propagation:

This topic describes broker Web services capability for extraction and propagation X.509 certificate token.

The broker does not support propagation of an X.509 certificate.

The X.509 certificate token extraction is supported in the following configurations:

Capability

- Extraction

Policy Enforcement Point (PEP) and direction

- In (provider)

SOAPInput node

Configured with a policy set and binding defining the X.509 certificate is present; see “Implementing WS-Security” on page 477.

Configured with a security profile defining propagation; see the Security profiles.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.



Identity and security token propagation

Identity and security token propagation enables the identity and security tokens (associated with each message) to be propagated throughout a message flow, and on to target applications through output or request nodes.

“X.509 certificate token capabilities” on page 512

Various WS-Services Security X.509 certificate token profile standards are supported by IBM Integration Bus.

Related tasks:



Configuring the extraction of an identity or security token

You can configure the SecurityPEP node or security enabled input nodes to extract the identity or security token from a message and store it in the properties tree identity fields, enabling it to be processed throughout the message flow and propagated at output or request nodes.

Related reference:

“X.509 certificate token capabilities for encryption” on page 513

For Web services, you can complete encryption by using an X.509 certificate token.

“X.509 certificate token capabilities for decryption” on page 514

For Web services, you can complete decryption by using an X.509 certificate token.

“X.509 certificate token capabilities for signing” on page 515

For Web services, you can use an X.509 certificate token for signing.

“X.509 certificate token capabilities for verifying” on page 516

For Web services, you can verify a signing by using an X.509 certificate token profile.

“X.509 certificate token capabilities for authentication” on page 517

For Web services, you can complete authentication by using an X.509 certificate token.

“X.509 certificate token capabilities for authorization” on page 518

The X.509 certificate token is not supported for authorization in any configuration or direction.

“X.509 certificate token capabilities for identity mapping” on page 519

For Web services, you can map an identity by using an X.509 certificate token.

SAML token capabilities:

This topic describes WS-Security SAML token capabilities of the broker.

The broker provides SAML pass-through support, which means that the mechanisms for subject confirmation are not enforced. The SAML token is extracted and passed to an external security token service (STS) for validation. The STS to be used is specified on a security profile.

For information about using WS-Security SAML tokens, see the following topics:

- “SAML token capabilities for encryption, decryption, signing, and verifying”
- “SAML token capabilities for authentication and authorization” on page 523
- “SAML token capabilities for identity mapping” on page 524
- “SAML token capabilities for extraction and propagation” on page 525

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.

Related tasks:



Setting up message flow security

Set up security on a message flow to control access based on the identity of a message passing through the message flow.



Integration Bus server security

You must consider several security aspects when you are setting up brokers running on Windows, Linux, z/OS, or UNIX platforms.

SAML token capabilities for encryption, decryption, signing, and verifying:

The broker provides SAML pass-through support, which means that the encryption, decryption, signing, and verifying mechanisms for achieving SAML subject confirmation are not enforced by the broker.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“SAML token capabilities” on page 521

This topic describes WS-Security SAML token capabilities of the broker.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.

Related reference:

“SAML token capabilities for authentication and authorization” on page 523

For Web services, you can complete authentication and authorization using a SAML token.

“SAML token capabilities for identity mapping” on page 524

This topic describes the broker Web services capability for identity mapping using a SAML token.

“SAML token capabilities for extraction and propagation” on page 525

This topic describes broker capability for extraction, propagation, or both using a SAML token in Web services.

SAML token capabilities for authentication and authorization:

For Web services, you can complete authentication and authorization using a SAML token.

The SAML token “Authentication” on page 476 and Authorization are supported only in the following configuration:

Capability

- Authenticate
- Authorize

Policy Enforcement Point (PEP) and direction

- In (provider)
 - SOAPInput node

Configured with a security policy set and binding that defines that a SAML pass-through 1.1 or SAML pass-through 2.0 token is present for authentication; see “Authentication” on page 478. The broker provides only SAML pass-through support, which means that the SAML token is extracted and passed to an external security token service (STS) for validation. The STS to be used is specified in a security profile. The STS processing can be used to implement authentication based on the SAML principal, and authorization based on SAML attributes.

Configured with a security profile defining the Policy Decision Point (PDP); see the PDP section that follows.

Trust Store or PDP

- WS-Trust v1.3 STS

Configured by using a WS-Trust v1.3 STS security profile specifying authentication, authorization or both; see *Creating a security profile for WS-Trust V1.3 (TFIM V6.2)*.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“SAML token capabilities” on page 521

This topic describes WS-Security SAML token capabilities of the broker.



Authorization

Authorization is the process of verifying that an identity token has permission to access a message flow.



Authentication and validation

Authentication is the process of establishing the identity of a user or system and verifying that the identity is valid. Applying authentication to a SAML security token involves validating the assertions that it carries and confirming that it is being processed within its validity period.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.

Related reference:

“SAML token capabilities for encryption, decryption, signing, and verifying” on page 522

The broker provides SAML pass-through support, which means that the encryption, decryption, signing, and verifying mechanisms for achieving SAML subject confirmation are not enforced by the broker.

“SAML token capabilities for identity mapping”

This topic describes the broker Web services capability for identity mapping using a SAML token.

“SAML token capabilities for extraction and propagation” on page 525

This topic describes broker capability for extraction, propagation, or both using a SAML token in Web services.

SAML token capabilities for identity mapping:

This topic describes the broker Web services capability for identity mapping using a SAML token.

Identity mapping from a SAML identity token to a mapped SAML identity token is supported only in the following configurations:

Capability

- Identity mapping

Policy Enforcement Point (PEP) and direction

- In (provider)

SOAPInput node

Configured with a security policy set and bindings that specifies a SAML pass-through 1.1 or SAML pass-through 2.0 authentication token.

Configured with a security profile defining the external Policy Decision Point (PDP); see the PDP section that follows.

Trust store or PDP

- WS-Trust v1.3 STS

Configured by using a WS-Trust v1.3 STS security profile that specifies identity mapping; see Creating a security profile for WS-Trust V1.3 (TFIM V6.2).

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“SAML token capabilities” on page 521

This topic describes WS-Security SAML token capabilities of the broker.



Identity mapping

Identity mapping is the transformation of a security token from one format to another format, or the federation of an identity from one realm to an equivalent identity in another realm.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.

Related reference:

“SAML token capabilities for encryption, decryption, signing, and verifying” on page 522

The broker provides SAML pass-through support, which means that the encryption, decryption, signing, and verifying mechanisms for achieving SAML subject confirmation are not enforced by the broker.

“SAML token capabilities for authentication and authorization” on page 523

For Web services, you can complete authentication and authorization using a SAML token.

“SAML token capabilities for extraction and propagation”

This topic describes broker capability for extraction, propagation, or both using a SAML token in Web services.

SAML token capabilities for extraction and propagation:

This topic describes broker capability for extraction, propagation, or both using a SAML token in Web services.

The extraction of a SAML token into the Properties folder source Identity fields, is supported in the following configurations:

Capability

- Extraction

Policy Enforcement Point (PEP) and direction

- In (provider)

SOAPInput node

Configured with a security policy set and bindings that specifies a SAML pass-through 1.1 or SAML pass-through 2.0 authentication token.

Configured with a security profile that defines propagation; see [Creating a security profile](#)

The propagation of a SAML token into the SOAP WS-Security header, from the token present in either the mapped or the source identity fields in the properties folder, is supported in the following configuration. For more information, see [Identity](#).

Capability

- Propagate

Policy Enforcement Point (PEP) and direction

- Out (consumer)

SOAPRequest node

SOAPAsyncRequest node

Configured with a security profile that defines propagation. For more information, see [Security profiles](#).

Configured with a security policy set and bindings that specifies SAML pass-through 1.1 or SAML pass-through 2.0 authentication token.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“SAML token capabilities” on page 521

This topic describes WS-Security SAML token capabilities of the broker.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.



Identity and security token propagation

Identity and security token propagation enables the identity and security tokens (associated with each message) to be propagated throughout a message flow, and on to target applications through output or request nodes.

Related tasks:



Configuring the extraction of an identity or security token

You can configure the SecurityPEP node or security enabled input nodes to extract the identity or security token from a message and store it in the properties tree identity fields, enabling it to be processed throughout the message flow and propagated at output or request nodes.

Related reference:

“SAML token capabilities for encryption, decryption, signing, and verifying” on page 522

The broker provides SAML pass-through support, which means that the encryption, decryption, signing, and verifying mechanisms for achieving SAML subject confirmation are not enforced by the broker.

“SAML token capabilities for identity mapping” on page 524

This topic describes the broker Web services capability for identity mapping using a SAML token.

“SAML token capabilities for authentication and authorization” on page 523

For Web services, you can complete authentication and authorization using a SAML token.

Kerberos token capabilities:

This topic describes WS-Security Kerberos token capabilities of the broker.

For details of using WS-Security Kerberos tokens, see the following topics:

- “Kerberos token capabilities for encryption, decryption, signing, and verifying” on page 527
- “Kerberos token capabilities for authentication and authorization” on page 528
- “Kerberos token capabilities for identity mapping” on page 529
- “Kerberos token capabilities for extraction and propagation” on page 530

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.

Related tasks:



Setting up message flow security

Set up security on a message flow to control access based on the identity of a

message passing through the message flow.

Integration Bus server security

You must consider several security aspects when you are setting up brokers running on Windows, Linux, z/OS, or UNIX platforms.

Kerberos token capabilities for encryption, decryption, signing, and verifying:

You can use Kerberos tokens for encryption, decryption, signing, and verifying.

Kerberos token encryption for providing message “Confidentiality” on page 477 and “Integrity” on page 477 on outgoing SOAP messages from the broker is supported in the following configurations:

Capability

- Encrypt, by using a Kerberos Key Distribution Center (KDC)
- Decrypt, by using the Kerberos keytab file

Policy Enforcement Point (PEP) and direction

- In (provider)
 - SOAPInput node
- In (consumer)
 - SOAPRequest node
 - SOAPAsyncResponse node

Configured with a policy set and binding defining the message “Integrity” on page 480

- Out (consumer)
 - SOAPRequest node
 - SOAPAsyncRequest node
 - Out (provider)
 - SOAPReply node
- Configured with a Kerberos policy set and binding.

Trust Store or Policy Decision Point (PDP)

- Kerberos KDC.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.

Message flow security overview

IBM Integration Bus provides a security manager, which enables you to control access to individual messages in a message flow, using the identity of the message.

“Kerberos token capabilities” on page 526

This topic describes WS-Security Kerberos token capabilities of the broker.

Related reference:

“Kerberos token capabilities for authentication and authorization”

This topic describes the broker Web services capability for authentication, authorization, or both using a Kerberos token.

“Kerberos token capabilities for identity mapping” on page 529

This topic describes broker Web services capability for identity mapping using a Kerberos token.

“Kerberos token capabilities for extraction and propagation” on page 530

This topic describes broker capability for extraction, propagation, or both using a Kerberos token in Web services.

Kerberos token capabilities for authentication and authorization:

This topic describes the broker Web services capability for authentication, authorization, or both using a Kerberos token.

Kerberos is not applicable to authorization. Kerberos token encryption for providing message “Authentication” on page 478 on outgoing SOAP messages from the broker is supported in the following configurations:

Capability

- Authenticate using a Kerberos keytab file.

Policy Enforcement Point (PEP) and direction

- In (provider)
 SOAPInput node
- In (consumer)
 SOAPRequest node
 SOAPAsyncResponse node

Configured with a policy set and binding defining the message “Integrity” on page 480

Trust Store or Policy Decision Point (PDP)

- Kerberos keytab file.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.



Message flow security overview

IBM Integration Bus provides a security manager, which enables you to control access to individual messages in a message flow, using the identity of the message.

“Kerberos token capabilities” on page 526

This topic describes WS-Security Kerberos token capabilities of the broker.



Authorization

Authorization is the process of verifying that an identity token has permission to access a message flow.

Authentication and validation

Authentication is the process of establishing the identity of a user or system and verifying that the identity is valid. Applying authentication to a SAML security token involves validating the assertions that it carries and confirming that it is being processed within its validity period.

Related reference:

“Kerberos token capabilities for encryption, decryption, signing, and verifying” on page 527

You can use Kerberos tokens for encryption, decryption, signing, and verifying.

“Kerberos token capabilities for identity mapping”

This topic describes broker Web services capability for identity mapping using a Kerberos token.

“Kerberos token capabilities for extraction and propagation” on page 530

This topic describes broker capability for extraction, propagation, or both using a Kerberos token in Web services.

Kerberos token capabilities for identity mapping:

This topic describes broker Web services capability for identity mapping using a Kerberos token.

Kerberos tickets from SOAP nodes are not supported for token mapping with an external security token server (STS) configured in the security profile.

On the Inbound route, with SOAPInput and SOAPAsyncResponse nodes, the presence of a security profile with propagation enabled causes the Kerberos Service Principal Name (SPN) to be placed in the properties tree as a Username token.

On the Outbound route, with SOAPRequest and SOAPAsyncRequest nodes, identity propagation can be used to provide the Kerberos Key Distribution Center (KDC) credentials. Arrange for the KDC credentials to be set as a Username and password token in the properties tree and associate the SOAP node with a security profile that specifies propagation; otherwise the KDC credentials are obtained using the Kerberos resource credentials that are created using the `mqsisetdbparms` command.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“Kerberos token capabilities” on page 526

This topic describes WS-Security Kerberos token capabilities of the broker.

Identity mapping

Identity mapping is the transformation of a security token from one format to another format, or the federation of an identity from one realm to an equivalent identity in another realm.

Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.

Related reference:

“Kerberos token capabilities for encryption, decryption, signing, and verifying” on page 527

You can use Kerberos tokens for encryption, decryption, signing, and verifying. “Kerberos token capabilities for authentication and authorization” on page 528 This topic describes the broker Web services capability for authentication, authorization, or both using a Kerberos token.

“Kerberos token capabilities for extraction and propagation”

This topic describes broker capability for extraction, propagation, or both using a Kerberos token in Web services.

Kerberos token capabilities for extraction and propagation:

This topic describes broker capability for extraction, propagation, or both using a Kerberos token in Web services.

Kerberos tickets from SOAP nodes are not supported for token extraction and propagation with an external security token server (STS) configured in the security profile.

On the Inbound route, with SOAPInput and SOAPAsyncResponse nodes, the presence of a security profile with propagation enabled causes the Kerberos Service Principal Name (SPN) to be placed in the properties tree as a Username token.

On the Outbound route, with SOAPRequest and SOAPAsyncRequest nodes, identity propagation can be used to provide the Kerberos Key Distribution Center (KDC) credentials. Arrange for the KDC credentials to be set as a Username and password token in the properties tree and associate the SOAP node with a security profile that specifies propagation; otherwise the KDC credentials are obtained using the Kerberos resource credentials that are created using the `mqsisetdbparms` command.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“Kerberos token capabilities” on page 526

This topic describes WS-Security Kerberos token capabilities of the broker.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.



Identity and security token propagation

Identity and security token propagation enables the identity and security tokens (associated with each message) to be propagated throughout a message flow, and on to target applications through output or request nodes.

Related tasks:



Configuring the extraction of an identity or security token

You can configure the SecurityPEP node or security enabled input nodes to extract the identity or security token from a message and store it in the properties tree identity fields, enabling it to be processed throughout the message flow and propagated at output or request nodes.

Related reference:

“Kerberos token capabilities for encryption, decryption, signing, and verifying” on page 527

You can use Kerberos tokens for encryption, decryption, signing, and verifying.

“Kerberos token capabilities for authentication and authorization” on page 528
This topic describes the broker Web services capability for authentication, authorization, or both using a Kerberos token.

“Kerberos token capabilities for identity mapping” on page 529

This topic describes broker Web services capability for identity mapping using a Kerberos token.

LTPA token capabilities:

This topic describes WS-Security LTPA token capabilities of the broker.

The broker provides LTPA pass-through support, which means that the LTPA token is extracted and passed to an external security token service (STS) for validation. The STS to be used is specified on a security profile.

For information about using WS-Security LTPA tokens, see the following topics:

- “LTPA token capabilities for encryption, decryption, signing, and verifying”
- “LTPA token capabilities for authentication and authorization” on page 532
- “LTPA token capabilities for identity mapping” on page 533
- “LTPA token capabilities for extraction and propagation” on page 534

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.

Related tasks:



Setting up message flow security

Set up security on a message flow to control access based on the identity of a message passing through the message flow.



Integration Bus server security

You must consider several security aspects when you are setting up brokers running on Windows, Linux, z/OS, or UNIX platforms.

LTPA token capabilities for encryption, decryption, signing, and verifying:

This topic describes broker Web services capability for encryption, decryption, signing, and verifying using LTPA tokens.

The LTPA token is not applicable, or supported, for the following in any configuration or direction:

- Encryption
- Decryption
- Signing
- Verifying

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“LTPA token capabilities” on page 531

This topic describes WS-Security LTPA token capabilities of the broker.

Related reference:

“LTPA token capabilities for authentication and authorization”

For Web services, you can complete authentication and authorization using an LTPA token.

“LTPA token capabilities for identity mapping” on page 533

This topic describes the broker Web services capability for identity mapping using an LTPA token.

“LTPA token capabilities for extraction and propagation” on page 534

This topic describes broker capability for extraction, propagation, or both using an LTPA token in Web services.

LTPA token capabilities for authentication and authorization:

For Web services, you can complete authentication and authorization using an LTPA token.

The LTPA token “Authentication” on page 476 and Authorization are supported only in the following configuration:

Capability

- Authenticate
- Authorize

Policy Enforcement Point (PEP) and direction

- In (provider)
 - SOAPInput node

Configured with a security policy set and binding that defines that an LTPA token is present for authentication; see “Authentication” on page 478. The broker provides only LTPA pass-through support, which means that the LTPA token is extracted and passed to an external security token service (STS) for validation. The STS to be used is specified in a security profile. The STS processing can be used to implement authentication and authorization based on the LTPA principal and realm.

Configured with a security profile defining the Policy Decision Point (PDP); see the PDP section that follows.

Trust Store or PDP

- WS-Trust v1.3 STS
 - Configured by using a WS-Trust v1.3 STS security profile specifying authentication, authorization or both; see Creating a security profile for WS-Trust V1.3 (TFIM V6.2).

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“LTPA token capabilities” on page 531

This topic describes WS-Security LTPA token capabilities of the broker.

 Authorization

Authorization is the process of verifying that an identity token has permission to

access a message flow.



Authentication and validation

Authentication is the process of establishing the identity of a user or system and verifying that the identity is valid. Applying authentication to a SAML security token involves validating the assertions that it carries and confirming that it is being processed within its validity period.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.

Related reference:

“LTPA token capabilities for identity mapping”

This topic describes the broker Web services capability for identity mapping using an LTPA token.

“LTPA token capabilities for extraction and propagation” on page 534

This topic describes broker capability for extraction, propagation, or both using an LTPA token in Web services.

“LTPA token capabilities for encryption, decryption, signing, and verifying” on page 531

This topic describes broker Web services capability for encryption, decryption, signing, and verifying using LTPA tokens.

LTPA token capabilities for identity mapping:

This topic describes the broker Web services capability for identity mapping using an LTPA token.

Identity mapping from or to an LTPA identity token is supported only in the following configurations:

Capability

- Identity mapping

Policy Enforcement Point (PEP) and direction

- In (provider)

SOAPInput node

Configured with a security policy set and bindings that specifies an LTPA pass-through authentication token.

Configured with a security profile defining the external Policy Decision Point (PDP); see the PDP section that follows.

Trust store or PDP

- WS-Trust v1.3 STS

Configured by using a WS-Trust v1.3 STS security profile that specifies identity mapping; see *Creating a security profile for WS-Trust V1.3 (TFIM V6.2)*.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“LTPA token capabilities” on page 531

This topic describes WS-Security LTPA token capabilities of the broker.

Identity mapping

Identity mapping is the transformation of a security token from one format to another format, or the federation of an identity from one realm to an equivalent identity in another realm.

Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.

Related reference:

“LTPA token capabilities for authentication and authorization” on page 532
For Web services, you can complete authentication and authorization using an LTPA token.

“LTPA token capabilities for extraction and propagation”
This topic describes broker capability for extraction, propagation, or both using an LTPA token in Web services.

“LTPA token capabilities for encryption, decryption, signing, and verifying” on page 531

This topic describes broker Web services capability for encryption, decryption, signing, and verifying using LTPA tokens.

LTPA token capabilities for extraction and propagation:

This topic describes broker capability for extraction, propagation, or both using an LTPA token in Web services.

The extraction of an LTPA token into the Properties folder source Identity fields, is supported in the following configurations:

Capability

- Extraction

Policy Enforcement Point (PEP) and direction

- In (provider)

SOAPInput node

Configured with a security policy set and bindings that specifies an LTPA pass-through authentication token.

Configured with a security profile that defines propagation; see [Creating a security profile](#)

The propagation of an LTPA token into the SOAP WS-Security header, from the token present in either the mapped or the source identity fields in the properties folder, is supported in the following configuration. For more information, see [Identity](#).

Capability

- Propagate

Policy Enforcement Point (PEP) and direction

- Out (consumer)

SOAPRequest node

SOAPAsyncRequest node

Configured with a security profile that defines propagation. For more information, see Security profiles.

Configured with a security policy set and bindings that specifies an LTPA pass-through authentication token.

Related concepts:

“WS-Security mechanisms” on page 476

The WS-Security specification provides three mechanisms for securing Web services at the message level: authentication, integrity, and confidentiality.

“LTPA token capabilities” on page 531

This topic describes WS-Security LTPA token capabilities of the broker.



Identity

In IBM Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.



Identity and security token propagation

Identity and security token propagation enables the identity and security tokens (associated with each message) to be propagated throughout a message flow, and on to target applications through output or request nodes.

Related tasks:



Configuring the extraction of an identity or security token

You can configure the SecurityPEP node or security enabled input nodes to extract the identity or security token from a message and store it in the properties tree identity fields, enabling it to be processed throughout the message flow and propagated at output or request nodes.

Related reference:

“LTPA token capabilities for authentication and authorization” on page 532

For Web services, you can complete authentication and authorization using an LTPA token.

“LTPA token capabilities for identity mapping” on page 533

This topic describes the broker Web services capability for identity mapping using an LTPA token.

“LTPA token capabilities for encryption, decryption, signing, and verifying” on page 531

This topic describes broker Web services capability for encryption, decryption, signing, and verifying using LTPA tokens.

Moving from IBM Integration Bus on a distributed system to z/OS

Define resources for IBM Integration Bus for z/OS and move your message flows.

About this task

Read the following topics for guidance on what action you might want to take in moving part of your operations to z/OS:

- “Overview of broker creation on z/OS” on page 314
- “Customizing the z/OS environment” on page 290
- “Creating a broker on z/OS” on page 313
- Administration in z/OS

After reviewing your requirements, re-create your broker on z/OS and deploy your message flows and integration servers to the broker on z/OS. If you have extended IBM Integration Bus in a distributed environment with user-defined parsers or message processing nodes, port them to run under z/OS.

Also consider the following points:

- Floating point conversion: z/OS runs under z/OS floating point format, so floating point operations on z/OS run in a different range and accuracy from distributed systems.
- Administration commands are partially implemented as console commands and partially as JCL commands. Some commands provide both options.
- Event log messages: All address spaces have a JOBLOG where messages appear. In addition to this, all messages appear on the SYSLOG, with important operator messages being filtered to the console through MPF (Message Processing Facility).

For information about message flow transactionality, see Message flow transactions.

Moving user applications

About this task

You can write your own applications to work with IBM Integration Bus. If these applications use the common subset of functionality of all IBM Integration Bus brokers, no migration is necessary. If you are using functionality that is available on some IBM Integration Bus systems only, for example message segmentation and WebSphere MQ message groups, be aware that IBM Integration Bus for z/OS does not provide support for this migration.

Related concepts:

“Overview of broker creation on z/OS” on page 314

After you have used SMP/E to install IBM Integration Bus for z/OS, the installed executable code is located inside the file system. JCL samples are located in the PDS <h1q>.SBIPSAMP, the JCL procedures are located in the PDS <h1q>.SBIPPROC, and load module for synchronizing statistics with SMF are located in the PDS <h1q>.SBIPAUTH.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

“Creating a broker on z/OS” on page 313

Create the broker component and the other resources on which it depends.

Related reference:



Administration in z/OS

In the z/OS environment, commands are issued through the console and others in batch jobs.

Changing locales

You can change the locale for the system on which a runtime component is installed.

About this task

The way in which you change the locale depends on the operating system:

- “Changing your locale on Linux and UNIX systems”
- “Changing your locale on Windows” on page 539
- “Changing your locale on z/OS” on page 540

IBM Integration Bus uses code page converters to support character sets from different environments. “Code page converters” on page 541 describes what a code page converter is, and how to generate new converters.

Related reference:

“Supported processors” on page 620

IBM Integration Bus is supported on multiple processors.

“Operating system requirements” on page 627

IBM Integration Bus is supported on multiple operating systems.



Supported code pages

Application messages must conform to supported code pages.

“Locales” on page 664

Message support is provided in a number of locales.

Changing your locale on Linux and UNIX systems

You can change your system locale on UNIX and Linux systems.

About this task

You can set environment variables to control the system locale. You can set these variables to be system-wide, or on a per-session basis:

LC_ALL Overrides all LC_* environment variables with the given value

LC_CTYPE

Character classification and case conversion

LC_COLLATE

Collation (sort) order

LC_TIME

Date and time formats

LC_NUMERIC

Non-monetary numeric formats

LC_MONETARY

Monetary formats

LC_MESSAGES

Formats of informative and diagnostic messages, and of interactive responses

LC_PAPER

Paper size

LC_NAME

Name formats

LC_ADDRESS

Address formats and location information

LC_TELEPHONE

Telephone number formats

LC_MEASUREMENT

Measurement units (Metric or Other)

LC_IDENTIFICATION

Metadata about the locale information

LANG The default value, which is used when either LC_ALL is not set, or an applicable value for LC_* is not set**NLSPATH**

Delimited list of paths to search for message catalogs

TZ Time zone

LC_MESSAGES and NLSPATH are the most important variables to the broker. These variables define the language and location of response messages that the broker uses. The broker profile file, `mqsiprofile`, sets NLSPATH. Either you, or your system must set LC_MESSAGES. The value set in LC_MESSAGES must be a value that is installed on your machine and that the broker recognizes. LC_CTYPE is also important to the broker because it defines the character conversion that the broker performs when interacting with the local environment.

Before setting these variables, check that the language and code page are installed on your machine, and are supported by IBM Integration Bus.

You can use the command `locale` to show your current locale. The command `locale -a` displays all the locales that are currently installed on the machine. Make sure that the locale you select for LANG and LC_ALL is in the list that is returned by the command `locale -a`. The values that `locale` uses and returns are case sensitive, therefore copy them exactly when assigning them to an environment variable.

For information on languages and code pages supported by IBM Integration Bus, see "Locales" on page 664 and Supported code pages.

If you use common desktop environment (CDE), use this environment to set the locale instead of setting LANG and LC_ALL directly. The NLSPATH variable respects either method.

For example, to set IBM Integration Bus to run in a UTF-8 environment set the following values in the profile:

```
LANG=en_US.utf-8
LC_ALL=en_US.utf-8
```

where `en_US` sets the language, and `utf-8` sets the code page.

When you start a broker component, the locale of that component is inherited from the shell in which it is started. The broker component uses the LC_MESSAGES environment variable as the search path in the NLSPATH environment variable (LC_MESSAGES is set when variable LC_ALL is exported).

Messages are sent to the syslog in the code page set by this locale. If you have multiple brokers that write to this syslog, their messages are in the code page of the locale in which they were started, for example:

locale	syslog code page	ccsid
pt_BR	iso8859-1	819

locale	syslog code page	ccsid
pt_BR	ibm-850	850
pt_BR	utf-8	1208

Set the locale of the user ID that runs the syslog daemon to one that is compatible with the locales of all brokers that write to the syslog on that system, for example, utf-8. For compatibility, you can set the default locale. On Solaris, set the LANG and LC_ALL variables in /etc/default/init. On AIX and Linux, these variables are in /etc/environment. This task is not required on HP-UX.

For full-time zone support in the broker, set the TZ variable using Continent/City notation. For example set TZ to Europe/London to make London, England the time zone, or set it to America/New_York to make New York, America the time zone.

If you want to add a new locale, refer to the operating system documentation for information about how to complete that task. If the code page of the new locale is not supported by IBM Integration Bus you must add it by “Generating a new code page converter” on page 542.

Related tasks:

“Generating a new code page converter” on page 542

Generate a code page converter to handle conversions of data that belongs to a code page that is not in the default set of code pages provided by IBM Integration Bus.

Related reference:

“Supported processors” on page 620

IBM Integration Bus is supported on multiple processors.

“Operating system requirements” on page 627

IBM Integration Bus is supported on multiple operating systems.



Supported code pages

Application messages must conform to supported code pages.

Changing your locale on Windows

Change your system locale on Windows to view objects and information in a different language or code page.

About this task

Brokers are started as services on Windows, and are therefore influenced by the system locale. The command-line functions are influenced by the locale that is set for the current user. IBM Integration Bus on Windows has all locale information installed by default. However, you might have to install additional locale packages, if prompted to do so by the Windows operating system.

To change locale, use one of the following methods:

- Install a locale-specific operating system.
- Alter the system or user locale by selecting *Regional Settings* in the Control Panel.

Messages are sent to the Event Log in the code page set by the current locale.

You can use the **chcp** command to change the active console code page. Enter the command at a command prompt; if you enter **chcp** without a parameter, it displays the current setting. If you enter it with a code page, it changes the locale to that code page.

For example, to check the current code page setting:

```
C:\>chcp
Active code page: 437
```

The current page is displayed (437 represents US-ASCII). If you want to change the value to GB18030, enter:

```
C:\>chcp 54936
Active code page: 54936
```

Before you use a code page, search for *windows-number* where *number* is the active code page you want to use in the list of Supported code pages. If the code page is not in the list, either use a code page that is in the list, or generate a new code page converter.

Related tasks:

“Generating a new code page converter” on page 542

Generate a code page converter to handle conversions of data that belongs to a code page that is not in the default set of code pages provided by IBM Integration Bus.

Related reference:

“Supported processors” on page 620

IBM Integration Bus is supported on multiple processors.

“Operating system requirements” on page 627

IBM Integration Bus is supported on multiple operating systems.



Supported code pages

Application messages must conform to supported code pages.

Changing your locale on z/OS

You can change your system locale on z/OS. If you want to change your system locale on z/OS, set the LANG, LC_ALL, and NLSPATH variables.

About this task

See “Installation information - broker” on page 320 for further information.

The locale is set in the broker profile (BIPBPROF) and you must run BIPGEN to create the broker ENVFILE and any integration server specific ENVFILES.

You can use the UNIX System Services (USS) executable `locale` to show your current locale. The command `locale -a` displays all the locales currently installed on the computer. Refer to the operating system documentation for information about adding new locales. If you add a new locale after you have installed IBM Integration Bus, install that locale's message catalogs from the original install media.

You can set IBM Integration Bus to operate with a specific code page. Set the code page after a period in the LANG and LC_ALL variable. This example sets the locale to `En_us` and the code page to IBM-1140 (EBCDIC `En_us` with euro):

```
LANG=En_us.IBM-1140
LC_ALL=En_us.IBM-1140
```

Make sure that the selected code page is one of the Supported code pages. If the code page is not in the list, either use a code page that is in the list, or generate a new code page converter.

Related tasks:

“Generating a new code page converter” on page 542

Generate a code page converter to handle conversions of data that belongs to a code page that is not in the default set of code pages provided by IBM Integration Bus.

Related reference:

“Supported processors” on page 620

IBM Integration Bus is supported on multiple processors.

“Operating system requirements” on page 627

IBM Integration Bus is supported on multiple operating systems.



Supported code pages

Application messages must conform to supported code pages.

Code page converters

Brokers complete string operations in Universal Character Set coded in 2 octets (UCS-2). If incoming strings are not encoded in UCS-2, they are converted to UCS-2 on arrival.

The broker uses international components for Unicode (ICU) code page converters to convert data. The Unicode Consortium has further information on Unicode.

A code page converter is a mapping from the byte sequence in one code page to a serialized representation of UCS-2, known as UCS Transformation Format 16-bit form (UTF-16). A code page converter allows the broker to create a UCS-2 representation of an incoming string.

When you handle UTF-16 data, CCSIDs 1200, 13488 and 17584 are treated differently to others. Traditionally, in ICU usage, the endian encoding of these CSSIDs was platform-specific, and IBM Integration Bus uses an encoding parameter with these CSSIDs. You can specify the encoding parameter as `MQENC_INTEGER_REVERSED` to use these CCSIDs to explicitly produce little endian data.

Consider this example of the use of a code page converter. A message comes in on a queue from z/OS, with the WebSphere MQ CCSID field set to 1047 (LATIN-1 Open Systems without euro). The broker looks up `ibm-1047` and uses the resulting converter to create a UCS-2 representation for internal use.

If you try to convert from a Unicode to a Non-Unicode character set, the following errors might occur:

- The target buffer is too small. This error causes a recoverable exception, which you can handle; alternatively, the message is rolled back.
- A code point in the source does not have an equivalent value in the target. At first, fallback mappings are attempted (for example, if you are converting to Japanese, a backslash (\) can be mapped to a yen (¥) if the conversion supplies it as a fallback mapping). If fallback mappings are not present, a recoverable exception is thrown. You can handle the exception, or the message is rolled back.

The MRM parser substitutes invalid code points with substitution characters.

IBM Integration Bus currently supports the code pages that are listed in Supported code pages. If you need support for an additional code page, or if you require a different variant of a code page, you can extend the broker to support this code page.

Related tasks:

“Generating a new code page converter”

Generate a code page converter to handle conversions of data that belongs to a code page that is not in the default set of code pages provided by IBM Integration Bus.

Related reference:



Supported code pages

Application messages must conform to supported code pages.

Generating a new code page converter:

Generate a code page converter to handle conversions of data that belongs to a code page that is not in the default set of code pages provided by IBM Integration Bus.

Before you begin

Before you start:

- Read “Code page converters” on page 541, which provides information about what a code page converter is, and about the code pages that IBM Integration Bus supports.

About this task

To generate a new code page converter:

Procedure

1. Create or find a mapping data file with the file extension `.ucm` for the converter that you require. You can download `.ucm` files from the ICU Character set mapping files archive. These mapping data files are available and can be modified without restriction. An example mapping data file is `ibm-1284_P100-1996.ucm`. (ICU is an external open source project, not an IBM tool.)
2. Rename the `.ucm` to a file name with the format `ibm-number.ucm` where *number* is a number that you choose to identify the code page. Make sure that this number is not already used in one of the Supported code pages. For example, you could rename `ibm-1284_P100-1996.ucm` to `ibm-1284.ucm`.
3. Go to ICU downloads and download the binary distribution for your system. An exact match is not important provided that the binary files are compatible. If you have problems building the converter, see the ICU user guide.
4. Extract the files from the binary distribution archive into a temporary directory.
5. Copy the library and binary files to a directory in the environment `PATH` and `LIBPATH`. (Alternatively, copy the library and binary files to directory that is not temporary and modify the environment `PATH` and `LIBPATH` to include this directory.)

6. One of the extracted files is `makeconv.exe`; use this `makeconv` tool to convert the mapping data file (`.ucm` files) into a binary converter file (`.cnv` file), by entering the following command:

```
makeconv mapping_file.ucm
```

where `mapping_file.ucm` is the mapping data file that you are using.

The name of the binary converter file that **makeconv** produces is:

```
mapping_file.cnv
```

where `mapping_file.cnv` is the name of the mapping data file that was converted.

To make the `.cnv` file for `ibm-1284.ucm`, use the following command:

```
makeconv ibm-1284.ucm
```

7. Copy the file with the file extension `.cnv` for the code page that you need, into a directory that IBM Integration Bus can access.

The name and location of the file is of the form

```
ibm-1284.cnv
```

and is located in the `$ICU_DATA/icudt48<platform-suffix>` directory, where the `<platform-suffix>` is one of the following values:

- l for little-endian ASCII platforms
 - b for big-endian ASCII platforms
 - e for EBCDIC platforms
8. Optional: If you do not want the new code page converter to be in the same location as other ICU data, you must associate the broker with the new directory where the converter is stored (the directory added must contain the full path, not including the `icudt48x` subdirectory):
 - To create a new broker that is associated with the converter, include the `-c` parameter on the **mqsicreatebroker** command.
 - To alter an existing broker to recognize the converter, include the `-c` parameter on the **mqsichangebroker** command.
 - To affect all the products and the broker command-line tools that are using ICU, add the `directory` to the **ICU_DATA** environment variable. If you have used either the **mqsicreatebroker** command or the **mqsichangebroker** command to specify the code page converter to be used, the broker ignores the **ICU_DATA** value.

Note: To ensure consistent behavior in all components, modify the **ICU_DATA** environment variable.

Related concepts:

“Code page converters” on page 541

Brokers complete string operations in Universal Character Set coded in 2 octets (UCS-2). If incoming strings are not encoded in UCS-2, they are converted to UCS-2 on arrival.

Related tasks:

“Changing locales” on page 536

You can change the locale for the system on which a runtime component is installed.

Related reference:

 Supported code pages

Application messages must conform to supported code pages.

`mqsicreatebroker` command

Use the `mqsicreatebroker` command to create a broker and its associated resources.

`mqsichangebroker` command

Use the `mqsichangebroker` command to change one or more of the configuration parameters of the broker.

Configuring for high availability

If you want to operate your IBM Integration Bus instances in a highly available configuration, you can set up your brokers to work either with a high availability manager, such as HACMP, or with WebSphere MQ multi-instance queue managers.

About this task

- “Using multi-instance brokers”
- “Using a broker with an existing high availability manager” on page 571
- “Using a broker with an existing Windows Cluster (Windows Server)” on page 582
- “Using external web servers with IBM Integration Bus” on page 610

Related concepts:

“HTTP proxy servlet overview” on page 584

By using the HTTP proxy servlet in a servlet container, you can support high availability, load distribution, access to the broker from multiple IP addresses and ports, and a larger number of concurrent HTTP sessions.

Related tasks:

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

Using multi-instance brokers

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

Before you begin

Before you start:

- Read the overview of multi-instance queue managers on WebSphere MQ; see “Multi-instance queue managers” on page 546.
- For further information about multi-instance queue managers, see the WebSphere MQ information center.

About this task

The set of tasks that describe how to configure a multi-instance broker assumes the following criteria:

- A network server is configured to host both the shared workpath for the multi-instance broker and the shared directories for the multi-instance queue manager.

- The network server is shared between two client machines, each of which has a licensed copy of the IBM Integration Bus and WebSphere MQ products installed.

The steps are true for all operating systems that are supported by IBM Integration Bus and WebSphere MQ except for z/OS. Multi-instance brokers and multi-instance queue managers are not supported on z/OS.

To configure a broker to run in multi-instance mode, complete the following steps.

Procedure

1. Create a shared work path directory on an NFS or NAS server. On Windows you can use a shared UNC path.
2. Create a WebSphere MQ multi-instance queue manager.
3. Create a multi-instance broker, by choosing one of the following methods:
 - Create a multi-instance broker, without domain restrictions. This method is recommended.
 - Create a multi-instance broker and multi-instance queue manager together.
 - Create a multi-instance broker, with domain restrictions.

What to do next

After you have created a multi-instance broker, you can complete the following actions.

- “Deleting a multi-instance broker” on page 562
- “Deleting a multi-instance queue manager” on page 563
- “Listing a multi-instance broker” on page 564
- “Backing up and restoring a multi-instance broker” on page 565
- “Configuring the global cache for multi-instance brokers” on page 566

Related concepts:

“Multi-instance queue managers” on page 546

A multi-instance queue manager restarts automatically on a standby server.

Related tasks:

“Creating the shared directories” on page 548

How you create the shared directories that you need for your multi-instance IBM Integration Bus.

“Creating the WebSphere MQ multi-instance queue manager” on page 549

How you create the WebSphere MQ multi-instance queue manager that you need for your multi-instance IBM Integration Bus.

“Creating a multi-instance broker” on page 556

How you create a multi-instance broker using an existing multi-instance queue manager for a high availability configuration.

“Creating a multi-instance broker with Windows on the Domain Controller” on page 558

How you create a multi-instance broker for a high availability configuration, when Windows is installed on the Domain Controller.

“Creating a multi-instance broker and a multi-instance queue manager” on page 560

How you create a multi-instance broker and a multi-instance queue manager together for a high availability configuration.

“Configuring the global cache for multi-instance brokers” on page 566
You can configure the global cache to withstand software or hardware failures so that it is available for as much time as possible. Configure a multi-instance broker to host container servers by using an XML policy file.

Related information:

[WebSphere MQ Version 7 product documentation](#)

Multi-instance queue managers

A multi-instance queue manager restarts automatically on a standby server.

Figure 12 shows a multi-instance configuration for QM1. WebSphere MQ is installed on two servers, one of which is a spare. One queue manager, QM1, has been created. One instance of QM1 is active, and is running on one server. The other instance of QM1 is running in standby on the other server, doing no active processing, but ready to take over from the active instance of QM1, if the active instance fails.

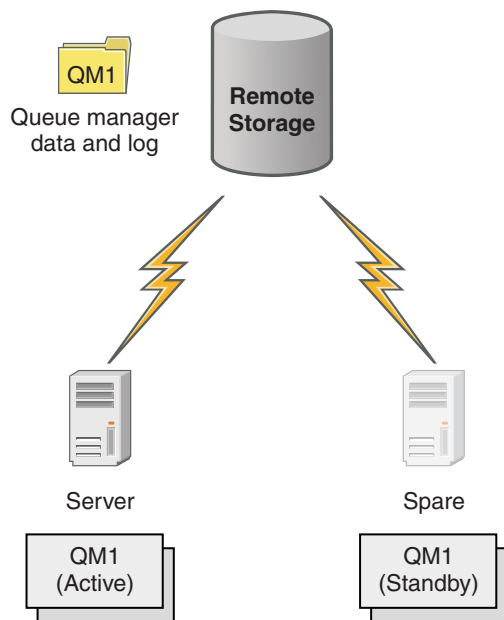


Figure 12. Multi-instance queue manager

When you intend to use a queue manager as a multi-instance queue manager, create a single queue manager on one of the servers using the WebSphere MQ **crtmqm** command, placing its queue manager data and logs in shared network storage. On the other server, rather than create the queue manager again, use the WebSphere MQ **addmqinf** command to create a reference to the queue manager data and logs on the network storage.

You can now run the queue manager from either of the servers. Each of the servers references the same queue manager data and logs; there is only one queue manager, and it is active on only one server at a time.

The queue manager can run either as a single instance queue manager, or as a multi-instance queue manager. In both cases only one instance of the queue manager is running, processing requests. The difference is that when running as a multi-instance queue manager, the server that is not running the active instance of

the queue manager runs as a standby instance, ready to take over from the active instance automatically if the active server fails.

The only control you have over which instance becomes active first is the order in which you start the queue manager on the two servers. The first instance to acquire read/write locks to the queue manager data becomes the active instance.

You can swap the active instance to the other server, once it has started, by stopping the active instance using the switchover option to transfer control to the standby.

The active instance of QM1 has exclusive access to the shared queue manager data and logs folders when it is running. The standby instance of QM1 detects when the active instance has failed, and becomes the active instance. It takes over the QM1 data and logs in the state they were left by the active instance, and accepts reconnections from clients and channels.

The active instance might fail for various reasons that result in the standby taking over:

- Failure of the server hosting the active queue manager instance.
- Failure of connectivity between the server hosting the active queue manager instance and the file system.
- Unresponsiveness of queue manager processes, detected by WebSphere MQ, which then shuts down the queue manager.

You can add the queue manager configuration information to multiple servers, and choose any two servers to run as the active/standby pair.

A multi-instance queue manager is one part of a high availability solution. You need some additional components to build a useful high availability solution.

- Client and channel reconnection to transfer WebSphere MQ connections to the computer that takes over running the active queue manager instance.
- A high performance shared network file system that manages locks correctly and provides protection against media and file server failure.
- Resilient networks and power supplies to eliminate single points of failure in the basic infrastructure.
- Applications that tolerate failover. In particular you need to pay close attention to the behavior of transactional applications, and to applications that browse WebSphere MQ queues.
- Monitoring and management of the active and standby instances to ensure that they are running, and to restart active instances that have failed. Although multi-instance queue managers restart automatically, you need to be sure that your standby instances are running, ready to take over, and that failed instances are brought back online as new standby instances.

You can use a multi-instance queue manager to provide high availability for your IBM Integration Bus environment. For more information, see “Configuring for high availability” on page 544.

Related tasks:

“Creating the WebSphere MQ multi-instance queue manager” on page 549

How you create the WebSphere MQ multi-instance queue manager that you need for your multi-instance IBM Integration Bus.

“Creating a multi-instance broker” on page 556

How you create a multi-instance broker using an existing multi-instance queue

manager for a high availability configuration.

“Creating a multi-instance broker and a multi-instance queue manager” on page 560

How you create a multi-instance broker and a multi-instance queue manager together for a high availability configuration.

Creating the shared directories

How you create the shared directories that you need for your multi-instance IBM Integration Bus.

Before you begin

Before you start:

Before you create the shared directory, read the documentation supplied with your NFS or NAS product used at your enterprise (or on Windows a shared UNC path). Note that, if you are intending to use an NFS server, this server must use NFSv4.

About this task

The following procedure gives a generalized set of instructions for a NFS shared path, because the specific instructions vary by product:

Procedure

1. Create an NFS Share on the server for the multi-instance WebSphere MQ queue manager.
2. Mount the NFS Share on both client nodes A and B, for the WebSphere MQ multi-instance queue manager, using a suitable location; for example, /Shared/Location/WMQ.
3. Ensure that the WMQ directories are owned by user and group mqm, and that the access permissions are set to rwx for user and group. For example `ls -al` displays:

```
drwxrwxr-x mqm mqm 4096 Jun 14 14:38 WMQ
```
4. Create an NFS share on the server for the multi-instance broker.
5. Mount the NFS Share on both client nodes A and B, for the multi-instance broker, using a suitable location; for example, /Shared/Location/IIB.
6. Ensure that the IIB directories are owned by user and group mqbrkrs, and that the access permissions are set to rwx for user and group. For example `ls -al` displays:

```
drwxrwxr-x mqbrkrs mqbrkrs 4096 Jun 14 14:39 IIB
```
7. If you are working on AIX you must turn off attribute caching so that you use the NFS server solely for multi-instance tasks.

Use the following command, on both client nodes A and B, to turn off attribute caching for the example share locations used in the preceding instructions:

```
nfs4cl setfsoptions /Shared/Location/WMQ noac
nfs4cl setfsoptions /Shared/Location/IIB noac
```

For further information, see AIX NFS commands

What to do next

Create a WebSphere MQ multi-instance queue manager.

Related tasks:

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

“Creating the WebSphere MQ multi-instance queue manager”

How you create the WebSphere MQ multi-instance queue manager that you need for your multi-instance IBM Integration Bus.

“Creating a multi-instance broker with Windows on the Domain Controller” on page 558

How you create a multi-instance broker for a high availability configuration, when Windows is installed on the Domain Controller.

Creating the WebSphere MQ multi-instance queue manager

How you create the WebSphere MQ multi-instance queue manager that you need for your multi-instance IBM Integration Bus.

Before you begin

Before you start:

Create the shared directories that you require for the multi-instance queue manager; see “Creating the shared directories” on page 548.

About this task

The following procedure gives an overview of how you create a multi-instance WebSphere MQ queue manager. See “Creating a multi-instance queue manager” on page 551 for more information.

Procedure

1. Create a multi-instance WebSphere MQ queue manager called QM1 on client node A. You do this by using the following command:

```
-crtmqm -md /SharedLocation/WMQ/data  
        -ld /SharedLocation/WMQ/logs QM1
```

where:

md Is the name of the directory used to hold data files for a queue manager.

ld Is the name of the directory used to hold log files.

Note that it is important that the name of the queue manager goes at the end of the syntax. See the WebSphere MQ documentation for further information on the **crtmqm** command.

2. Add the details of WebSphere MQ queue manager QM1 onto client node B. You do this by using the following command:

```
-addmqinf -v Name=QM1 -v Directory=WMQ -v Prefix=/var/mqm  
          -v DataPath=/SharedLocation/WMQ/data/QM1
```

where:

Name Is the name of the queue manager.

Directory

Is the name of the queue manager data directory.

Prefix Is the directory path *under* which this queue manager data directory is stored by default.

Data Path

Is the data path where the queue manager data files are placed. The value of Directory is not appended automatically to this path; you must provide the transformed queue manager name as part of DataPath.

The parameters listed above are all required parameters on Windows and UNIX platforms, with the exception of DataPath, which is optional on UNIX platforms only.

See the WebSphere MQ documentation for further information on the **addmqinf** command.

3. Start queue manager QM1 on client node A in multi-instance mode. You do this by using the following command:

```
strmqm -x QM1
```

See the WebSphere MQ documentation for further information on the **strmqm** command.

4. Observe the queue manager running in active mode. You do this by using the following command:

```
dspmq -x
```

See the WebSphere MQ documentation for further information on the **dspmq** command.

5. Start queue manager QM1 on client node B. Observe the queue manager running in standby mode.

6. Ensure that queue manager QM1 works as follows when simulating a failover from node A to node B:

- a. Stop queue manager QM1 on client node A. You do this by using the following command:

```
endmqm -s QM1
```

Observe on client node B, queue manager QM1 running in active mode, and on client node A that queue manager QM1 is now stopped.

- b. Restart queue manager QM1 on client node A. You do this by using the following command:

```
strmqm -x QM1
```

Observe on client node A, queue manager QM1 running in standby mode, and on client node B, queue manager QM1 running in active mode.

What to do next

Create a multi-instance broker.

Related concepts:

“Creating a multi-instance queue manager” on page 551

Create a multi-instance queue manager by creating the queue manager on one server, and configuring WebSphere MQ on another server to use the shared queue manager data and logs.

Related tasks:

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

“Creating the shared directories” on page 548

How you create the shared directories that you need for your multi-instance IBM Integration Bus.

“Creating a multi-instance broker with Windows on the Domain Controller” on page 558

How you create a multi-instance broker for a high availability configuration, when Windows is installed on the Domain Controller.

Creating a multi-instance queue manager:

Create a multi-instance queue manager by creating the queue manager on one server, and configuring WebSphere MQ on another server to use the shared queue manager data and logs.

Most of the work of setting up a multi-instance queue manager involves setting up the network storage that holds the queue manager data and log files, and making the files available to other servers using network shares. These tasks need to be performed by someone with administrative authority, such as *root* on UNIX systems. Once the shares are set up, and a normal queue manager has been created using the shares for its queue manager data and logs, you only need to configure WebSphere MQ on the other servers. You do not create the queue manager again on the other servers.

File access control

You need to take care that the user and group `mqm` on all other servers have permission to access the shares.

On UNIX platforms, you need to make the `uid` and `gid` of `mqm` the same on all the systems. You might need to edit `/etc/passwd` on each system to set a common `uid` and `gid` for `mqm`, then reboot your UNIX systems.

On Microsoft Windows, you must install WebSphere MQ on a domain server, and create a user to own WebSphere MQ resources. The user must either be a member of domain group `mqm` or a member of another global domain group which is directly or indirectly a member of `mqm`. Make this user the owner of the shared queue manager and log files. The `sid` of the user who owns the queue manager and log files is then the same as the `sid` of the user that runs instances of the queue manager.

Configuration information

Configure as many queue manager instances as you need by modifying the WebSphere MQ configuration information about each server that has WebSphere MQ installed and is to run a queue manager instance. The commands, `dspmqinf` and `addmqinf` help you to configure the additional queue manager instances, or you can edit the `mq5.ini` file on UNIX servers directly. The topics, “Create a multi-instance queue manager on Linux” on page 552 and “Create a multi-instance queue manager on Windows Server” on page 554 are examples showing how to configure a multi-instance queue manager.

On UNIX systems, you can share a single `mq5.ini` file by placing it on the network share and setting the `AMQ_MQS_INI_LOCATION` environment variable to point to it.

On Microsoft Windows, WebSphere MQ has for a number of releases kept its configuration information in the Windows registry. As from Version 7.0.1, some configuration information has moved from the registry into configuration files.

Multi-instance queue manager configuration is moved out of the registry into `qm.ini` and `qmstatus.ini` files, which are located in the queue manager data directory. The WebSphere MQ configuration information (`mqqs.ini` on UNIX platforms) remains in the registry on Windows as does all the configuration information for existing queue managers and new queue managers that are configured to use the default data directory.

Restrictions

1. Configure multiple instances of the same queue manager only on servers having the same operating system, architecture (both machines having 32-bit or 64-bit word size, for example), and endian settings.
2. All WebSphere MQ installations must be at Version 7.5.0.1 for distributed platforms, and at Version 7.1 for z/OS.
3. Typically active and standby installations are maintained at the same maintenance level. Consult the maintenance instructions for each upgrade to check if you must upgrade all installations together.
4. The network share that contains queue manager data and logs can only be shared between queue managers that are configured with the same WebSphere MQ user, group, and access control mechanism. For example, the network share set up on a Linux server could contain separate queue manager data and logs for AIX, Solaris, HP-UX, and Linux queue managers.
5. On UNIX systems, configure the shared file system on networked storage with a hard, interruptible, mount rather than a soft mount. A hard interruptible mount forces the queue manager to hang until it is interrupted by a system call. Soft mounts do not guarantee data consistency after a server crash.
6. On Microsoft Windows, the domain group `mqm` must have full access to shared WebSphere MQ log and data directories.
7. The shared log and data directories cannot be stored on a FAT, or an NFSv3 file system.
8. z/OS does not support multi-instance queue managers; use queue sharing groups.
9. Multi-instance queue managers are supported only on the platform on which they are created.

Related concepts:

“Create a multi-instance queue manager on Linux”

An example shows how to set up a multi-instance queue manager on Linux. The setup is small to illustrate the concepts involved. The example is based on Linux Red Hat Enterprise 5. The steps differ on other UNIX platforms.

“Create a multi-instance queue manager on Windows Server” on page 554

An example shows how to set up an example multi-instance queue manager on Windows Server. The setup is small and simple, to demonstrate the concepts involved, rather than being production scale. The steps might differ, depending on the version of Windows Server.

Related tasks:

“Creating the shared directories” on page 548

How you create the shared directories that you need for your multi-instance IBM Integration Bus.

Create a multi-instance queue manager on Linux:

An example shows how to set up a multi-instance queue manager on Linux. The setup is small to illustrate the concepts involved. The example is based on Linux Red Hat Enterprise 5. The steps differ on other UNIX platforms.

The example is set up on a 2 GHz notebook computer with 3 GB RAM running Windows. Two VMware virtual machines run Linux Red Hat Enterprise 5 in 640 MB images. A WebSphere MQ client is installed on an additional 400 MB VMWare image that runs Windows and runs the sample high availability applications. All the virtual machines are configured as part of a VMware host-only network for security reasons. Although it is generally recommended not to use an automatically generated IP address, it is acceptable to use one for a short demonstration, because the IP address is unlikely to change.

Example

Table 10. Illustrative multi-instance queue manager configuration on Linux

Server 1	Server 2
Log in as <i>root</i>	
Follow the instructions in WebSphere MQ Quick Beginnings for Linux to install WebSphere MQ, create the mqm user and group, and define /var/mqm.	
Carry out the task, "Verifying shared file system locking" on page 555 to check the file system supports multi-instance queue managers.	
Check what uid and gid /etc/passwd displays for mqm, for example, mqm:x:501:100:MQ User:/var/mqm:/bin/bash	Match the uid and gid for mqm in /etc/passwd and reboot if you have had to change the values.
Create log and data directories in a common folder, /MQHA, that is to be shared. For example, 1. mkdir /MQHA 2. mkdir /MQHA/logs 3. mkdir /MQHA/qmgrs	Create the folder, /MQHA, to mount the shared file system. Keep the path the same as on server 1; for example: 1. mkdir /MQHA
Ensure that the MQHA directories are owned by user and group mqm, and the access permissions are set to rwx for user and group; for example ls -al displays, drwxrwxr-x mqm mqm 4096 Nov 27 14:38 MQDATA 1. chown -R mqm:mqm /MQHA 2. chmod -R ug+rwx /MQHA	
Create the queue manager: crtmqm -ld /MQHA/logs -md /MQHA/qmgrs -q QM1	
Add(The '*' allows all machines that can reach this one mount /MQHA for read/write. Restrict access on a production machine.) /MQHA *(rw, sync, no_wdelay, fsid=0) to /etc/exports	
Start the NFS daemon: <i>/etc/init.d/nfs start</i>	
Discover the host name or IP address of server 1: ifconfig (Try the command /sbin/ifconfig if the simple ifconfig command does not work.)	Mount the exported file system /MQHA: mount -hard -intr -t nfs4 192.168.217.130:/ /MQHA
Copy the queue manager configuration details from Server 1, dspmqinf -o command QM1 and copy the result to the clip board, addmqinf -s QueueManager -v Name=QM1 -v Directory=QM1 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM1	Paste the queue manager configuration command into Server 2, addmqinf -s QueueManager -v Name=QM1 -v Directory=QM1 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM1
Start the queue manager instances, in either order, with the -x parameter: strmqm -x QM1	

Related concepts:

“Creating a multi-instance queue manager” on page 551

Create a multi-instance queue manager by creating the queue manager on one server, and configuring WebSphere MQ on another server to use the shared queue manager data and logs.

“Create a multi-instance queue manager on Windows Server”

An example shows how to set up an example multi-instance queue manager on Windows Server. The setup is small and simple, to demonstrate the concepts involved, rather than being production scale. The steps might differ, depending on the version of Windows Server.

Create a multi-instance queue manager on Windows Server:

An example shows how to set up an example multi-instance queue manager on Windows Server. The setup is small and simple, to demonstrate the concepts involved, rather than being production scale. The steps might differ, depending on the version of Windows Server.

Before you begin

On Windows, multi-instance queue managers must run on a domain controller. For this reason, and for this example, install two Windows servers, and configure them as domain controllers in the same domain.

The example was set up on a 2 GHz notebook computer with 3 GB RAM running Windows. Two VMware virtual machines ran Windows Server in 500 MB images. A WebSphere MQ client was installed on an additional 400 MB VMware image that ran Windows to run the sample high availability applications. All the virtual machines were configured as part of a VMware host-only network for security reasons. Although it is recommended not to use automatically generated IP address, for a short demonstration it is satisfactory as the IP addresses ought not to change. As part of the configuration you need to update the TCP/IP properties of the virtual machines with the address of the domain server you have chosen to be the DNS server.

If you configure a queue manager to run as a service, you need to check that the user the service logs on as is a member of the local group mqm.

Example

Table 11. Illustrative multi-instance queue manager configuration on Windows Server

Server 1	Server 2
Log in with user who is a member of the local group mqm.	
Create log and data directories in a folder on an NTFS drive, c:\MQHA, making sure that the owner is a member of mqm, and mqm has full-control authority to the folders. C:\MQHA\data C:\MQHA\log	
Create a share MQHA for C:\MQHA.	Connect to \\hostname\MQHA
Using WebSphere MQ Explorer, create a queue manager, QM1, overriding the default paths to the data and log folders and use UNC names to refer to the data and log folders. \\hostname\MQHA\data and \\hostname\MQHA\log.	

Table 11. Illustrative multi-instance queue manager configuration on Windows Server (continued)

Server 1	Server 2
<p>On a command-line run,</p> <pre>dspmqlinf -o command QM1</pre> <p>and copy the result to the clip board,</p> <pre>addmqinf -s QueueManager -v Name=QM1 -v Directory=QM1 -v Prefix="C:\IBM\MQ\Data" -v DataPath="\\hostname\MQHA\data\QM1"</pre>	<p>Paste and run the command,</p> <pre>addmqinf -s QueueManager -v Name=QM1 -v Directory=QM1 -v Prefix="C:\IBM\MQ\Data" -v DataPath="\\hostname\MQHA\data\QM1"</pre>
<p>Using WebSphere MQ Explorer, start the manager instances with the Permit a standby instance check box selected.</p>	

Related concepts:

“Creating a multi-instance queue manager” on page 551

Create a multi-instance queue manager by creating the queue manager on one server, and configuring WebSphere MQ on another server to use the shared queue manager data and logs.

“Create a multi-instance queue manager on Linux” on page 552

An example shows how to set up a multi-instance queue manager on Linux. The setup is small to illustrate the concepts involved. The example is based on Linux Red Hat Enterprise 5. The steps differ on other UNIX platforms.

Verifying shared file system locking:

Run the WebSphere MQ **amqmfsc** command to check obtaining and releasing file locks, and concurrent writing on UNIX systems.

Before you begin

Before you start:

You need a machine with networked storage, and two other machines connected to it that have WebSphere MQ installed. You need to have administrator (root) authority to configure the file system, and be a WebSphere MQ administrator to run **amqmfsc**.

About this task

Create the directory or directories on the networked storage that you are going to use to share queue manager data and logs. The directory owner needs to be a WebSphere MQ administrator, that is, a member of the mqm group on UNIX. The user who runs the tests must have WebSphere MQ administrator authority.

Use the example of exporting and mounting a file system in “Create a multi-instance queue manager on Linux” on page 552 to help you through configuring the file system. Different file systems require different configuration steps. Refer to the file system documentation.

Procedure

1. Export the shared directory on the networked storage system and start the NFS daemon.
2. Mount the exported directory on the two WebSphere MQ servers.
3. Run **amqmfsc**, without any options, on each system to check basic locking

4. Run **amqmfscck** on both WebSphere MQ systems simultaneously, using the **-c** option, to test writing to the directory concurrently.
5. Run **amqmfscck** on both WebSphere MQ systems at the same time, using the **-w** option, to test waiting for and releasing a lock on the directory concurrently.

Creating a multi-instance broker

How you create a multi-instance broker using an existing multi-instance queue manager for a high availability configuration.

Before you begin

Note: The set of instructions that follows assumes that the multi-instance queue manager exists. To create a multi-instance broker and multi-instance queue manager together, see “Creating a multi-instance broker and a multi-instance queue manager” on page 560.

Before you start:

Create the WebSphere MQ multi-instance queue manager.

The queue manager must be created with the **"-a"** or **"-ar"** flag on **crtmqm**, specifying a domain group that WMQ can use for securing shared files. If you have the option, use the **"-ar"** flag. The **"-ar"** flag is available when using WebSphere MQ Version 7.1.

For more information, see “Creating the WebSphere MQ multi-instance queue manager” on page 549.

About this task

You do not require a Windows installation on the domain controllers to use these instructions. To use an existing queue manager to create a multi-instance broker:

Procedure

1. Create the shared directories that you require for the multi-instance broker. For more information about shared directories, see “Creating the shared directories” on page 548
2. Define the following user and groups:
 - a. A domain group which is a member of the local mqbrkrs on both systems. For example, WMB\Domain mqbrkrs
 - b. A domain user which is a member of the Domain mqbrkrs group. This ID is used for running the broker. For example, WMB\mqsiuser.
 - c. A domain user which is a member of the Domain mqbrkrs group and a member of the local Administrators group on both machines. This ID is used for creating the broker. It can be the same as the previous ID, but it is not necessary to run the broker as an Administrator. For example, WMB\mqsiuser-admin.
3. Create a folder for the broker shared files on the file server, c:\mqsi share. Update the security permissions of the folder, using the **change permissions** section of the **Advanced Security Settings** panel:
 - a. Clear "include inheritable permissions from this objects parent".
 - b. Remove all of the existing entries.
 - c. Add "Domain mqbrkrs" with "Full Control". If this folder is also being used for multi-instance queue manager, then the domain group used to

secure the queue manager must also be added with "Full Control" as well. Use the advanced sharing options to share this folder.

- d. Remove the default "Everyone" user from the list.
 - e. Add "Domain mqbrkrs" with "Full Control".
 - f. Optional: If this share is also being used for the multi-instance queue manager, then the domain group used to secure the queue manager must also be added with "Full Control" as well.
4. As user mqsiuser-admin, open a command console with elevated privileges. To open a command console with elevated privileges, use the **mqsi command console** command. For more information, see **mqsi command console** command

5. Create a multi-instance broker called MB1 on client node A. Do this by using the following command:

```
mqsi createbroker MIMB1 -i "WMB\mqsiuser" -a <password> -q MIQM1 -e \\MyServer\mqsi share -B "WMB\Domain mqbrkrs"
```

Where MIQM1 is the name of the existing multi-instance queue manager created with the -a or -ar flags as mentioned in the above code. Replace <password> with the mqsiuser-admin password. If you want to start the multi-instance broker as a WebSphere MQ service, you can specify -d defined on the **mqsi createbroker** command.

You must ensure that the SharedLocation exists, and that your user ID has access to the shared location before you run this command.

See **mqsi createbroker** command for further information.

6. Add the details of broker MB1 onto client node B. Do this by using the following command:

```
mqsi addbrokerinstance MIMB1 -i "WMB\mqsiuser" -a <password> -e \\MyServer\mqsi share
```

See **mqsi addbrokerinstance** command for further information.

Note that the preceding example is for a UNIX system.

7. Start queue manager MIQM1 so that it is active on client node A.
8. Start broker MB1 on client node A. Do this by using the following command:

```
mqsi start MB1
```
9. Start broker MB1 on client node B. Observe that broker MB1 is running in standby mode against the standby queue manager QM1. Do this by using the following command:

```
mqsi list
```
10. Optional: Ensure that broker MB1 works as follows:
 - a. Stop broker MB1 and queue manager QM1 on client node A. Observe on client node B that broker MB1 and queue manager QM1 change from standby to active mode.
 - b. Restart queue manager QM1 and broker MB1 on client node A. Observe on client node B that queue manager QM1 and broker MB1 return to standby mode.

Results

You have created a multi-instance broker using an existing multi-instance queue manager. When broker MB1 and queue manager QM1 stop on client node A, the same broker and queue manager on client node B become active, and return to standby when client node A becomes active again.

Related tasks:

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

“Creating the shared directories” on page 548

How you create the shared directories that you need for your multi-instance IBM Integration Bus.

“Creating the WebSphere MQ multi-instance queue manager” on page 549

How you create the WebSphere MQ multi-instance queue manager that you need for your multi-instance IBM Integration Bus.

“Configuring a broker as a WebSphere MQ service” on page 614

Use these topics to make changes when your broker is operating as a WebSphere MQ service.

Related reference:



mqsicommandconsole command

Use the **mqsicommandconsole** command to launch an elevated command console from which commands that require elevation on Windows can be run.



mqsicreatebroker command

Use the **mqsicreatebroker** command to create a broker and its associated resources.



mqsiaaddbrokerinstance command

Use the **mqsiaaddbrokerinstance** command to create a multi-instance broker on a server where IBM Integration Bus has been installed.

Creating a multi-instance broker with Windows on the Domain Controller

How you create a multi-instance broker for a high availability configuration, when Windows is installed on the Domain Controller.

Before you begin

Note: The set of instructions that follows assumes that all machines with brokers are Windows Domain Controllers. To create a multi-instance broker that does not require the host to be a Windows Domain Controller, see “Creating a multi-instance broker” on page 556.

Before you start:

1. Create the WebSphere MQ multi-instance queue manager; see “Creating the WebSphere MQ multi-instance queue manager” on page 549.
2. Create the shared directories that you require for the multi-instance broker; see “Creating the shared directories” on page 548.

About this task

A multi-instance broker can be created only by using the **mqsicreatebroker** command. It is not possible to convert an existing broker to a multi-instance broker by using the **mqsichangebroker** command, nor is it possible to migrate a broker from a previous release to a Version 9.0 multi-instance broker.

Similarly, a broker instance can be created only by using the **mqsiaaddbrokerinstance** command.

You can configure a multi-instance broker to start as a WebSphere MQ service.

To create a multi-instance broker:

Procedure

1. Create a multi-instance broker called MB1 on client node A, by entering the following command:

```
mqsicreatebroker MB1 -q QM1 -e /SharedLocation/WMB
```

On Windows systems, to run the **mqsicreatebroker** command you must open a command console with elevated privileges. To open a command console with elevated privileges, use the **mqsicommandconsole** command. For more information, see **mqsicommandconsole** command.

Specify **-d** defined on the **mqsicreatebroker** command to start the multi-instance broker as a WebSphere MQ service.

You must ensure that the SharedLocation exists, and that your user ID has access to the shared location before you run this command.

For more information, see the **mqsicreatebroker** command.

2. Add the details of broker MB1 onto client node B. Do this by using the following command:

```
mqsiaddbrokeinstance MB1 -e /SharedLocation/WMB
```

For more information, see **mqsiaddbrokeinstance** command.

Note that the preceding example is for a UNIX system.

3. Start queue manager MIQM1 so that it is active on client node A.
4. Start broker MB1 on client node A. Do this by using the following command:

```
mqsistart MB1
```

5. Start broker MB1 on client node B. Observe that broker MB1 runs in standby mode against the standby queue managerQM1. Do this by using the following command:

```
mqsilist
```

6. Optional: Ensure that broker MB1 works as follows:
 - a. Stop broker MB1 and queue manager QM1 on client node A. Observe on client node B that broker MB1 and queue manager QM1 change from standby to active mode.
 - b. Restart queue manager QM1 and broker MB1 on client node A. Observe on client node B that queue manager QM1 and broker MB1 return to standby mode.

Results

You have created a multi-instance broker using an existing multi-instance queue manager. When broker MB1 and queue manager QM1 stop on client node A, the same broker and queue manager on client node B become active, and return to standby when client node A becomes active again.

Related tasks:

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

“Creating the shared directories” on page 548

How you create the shared directories that you need for your multi-instance IBM Integration Bus.

“Creating the WebSphere MQ multi-instance queue manager” on page 549

How you create the WebSphere MQ multi-instance queue manager that you need

for your multi-instance IBM Integration Bus.

“Configuring a broker as a WebSphere MQ service” on page 614

Use these topics to make changes when your broker is operating as a WebSphere MQ service.

Related reference:



mqsicommandconsole command

Use the **mqsicommandconsole** command to launch an elevated command console from which commands that require elevation on Windows can be run.



mqsicreatebroker command

Use the **mqsicreatebroker** command to create a broker and its associated resources.



mqsiaaddbrokerinstance command

Use the **mqsiaaddbrokerinstance** command to create a multi-instance broker on a server where IBM Integration Bus has been installed.

Creating a multi-instance broker and a multi-instance queue manager

How you create a multi-instance broker and a multi-instance queue manager together for a high availability configuration.

Before you begin

Before you start:

When you create the multi-instance queue manager, use the `-ar crtmqm` option. Before you can create a multi-instance broker on a single domain, you must configure user and file permissions.

About this task

As user `mqsuser-admin`, open a command console with elevated privileges by using the **mqsicommandconsole** command. To create a multi-instance broker and a multi-instance queue manager together, complete the following steps:

Procedure

1. Create the shared directories that you require for the multi-instance broker. For more information about shared directories, see “Creating the shared directories” on page 548
2. Define the following user and groups:
 - a. A domain group which is a member of the local `mqbrkrs` on both systems. For example, `WMB\Domain mqbrkrs`
 - b. A domain group which is a member of the local `mqm` group on both systems. For example, `WMB\Domain mqm`
 - c. A domain user which is a member of the Domain `mqbrkrs` and `mqm` groups. This ID is used for running the broker. For example, `WMB\mqsuser`.
 - d. A domain user which is a member of the Domain `mqbrkrs` group and a member of the local Administrators group on both machines. This ID is used for creating the broker. It can be the same as the previous ID, but it is not necessary to run the broker as an Administrator. For example, `WMB\mqsuser-admin`. The listed user and groups are using the example domain name `WMB`.

3. Create a folder for the broker shared files on the file server, c:\mqshare. Update the security permissions of the folder, using the **change permissions** section of the **Advanced Security Settings** panel:
 - a. Clear "include inheritable permissions from this objects parent".
 - b. Remove all of the existing entries.
 - c. Add "Domain mqbrkr" with "Full Control".
 - d. Add "Domain mqm" with "Full Control".
 - e. Remove the default "Everyone" user from the list.
 - f. Add "Domain mqbrkr" with "Full Control".
4. Create a multi-instance broker called MB1 on client node A. Do this by using the following command:


```
mqsicreatebroker MIMB1 -i "WMB\mqsiuser" -a <password> -q MIQM1 -e \\MyServer\mqshare -B "WMB\Domain mqbrkr"
```

Where MIQM1 is the name of the existing multi-instance queue manager created with the -a or -ar flags as mentioned in the above code. Replace <password> with the mqsiuser-admin password. If you want to start the multi-instance broker as a WebSphere MQ service, you can specify -d defined on the **mqsicreatebroker** command.

Specify -d defined on the **mqsicreatebroker** command to start the multi-instance broker as a WebSphere MQ service.

You must ensure that the SharedLocation exists, and that your user ID has access to the shared location before you run this command.

See the **mqsicreatebroker** command for further information.
5. Add the details of broker MB1 onto client node B. Do this by using the following command:


```
mqsiaadbrokerinstance MIMB1 -i "WMB\mqsiuser" -a <password> -e \\MyServer\mqshare
```

See **mqsiaadbrokerinstance** command for further information.

Note that the preceding example is for a UNIX system.
6. Start queue manager MIQM1 so that it is active on client node A.
7. Start broker MB1 on client node A. Do this by using the following command:


```
mqsistart MB1
```
8. Start broker MB1 on client node B. Observe that broker MB1 is running in standby mode against the standby queue managerQM1. Do this by using the following command:


```
mqsilist
```
9. Optional: Ensure that broker MB1 works as follows:
 - a. Stop broker MB1 and queue manager QM1 on client node A. Observe on client node B that broker MB1 and queue manager QM1 change from standby to active mode.
 - b. Restart queue manager QM1 and broker MB1 on client node A. Observe on client node B that queue manager QM1 and broker MB1 return to standby mode.

Results

You have created both a multi-instance broker and multi-instance queue manager. When broker MB1 and queue manager QM1 stop on client node A, the same broker and queue manager on client node B become active. When node A becomes active again, it takes on the role of standby node.

Related tasks:

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

“Creating the shared directories” on page 548

How you create the shared directories that you need for your multi-instance IBM Integration Bus.

“Creating the WebSphere MQ multi-instance queue manager” on page 549

How you create the WebSphere MQ multi-instance queue manager that you need for your multi-instance IBM Integration Bus.

“Configuring a broker as a WebSphere MQ service” on page 614

Use these topics to make changes when your broker is operating as a WebSphere MQ service.

Related reference:



mqsicommandconsole command

Use the **mqsicommandconsole** command to launch an elevated command console from which commands that require elevation on Windows can be run.



mqsicreatebroker command

Use the **mqsicreatebroker** command to create a broker and its associated resources.



mqsiaaddbrokerinstance command

Use the **mqsiaaddbrokerinstance** command to create a multi-instance broker on a server where IBM Integration Bus has been installed.

Deleting a multi-instance broker

How you delete a multi-instance broker.

About this task

The order of deletion of a multi-instance message broker and its associated instances is important. You should take care to use the correct command for each step in the process.

Ideally, you should remove all of the broker instances using the **mqsiremovebrokerinstance** command before you attempt to remove the broker itself. This command removes all local references to the broker instance, but does not affect the shared configuration for the multi-instance broker on the shared work path (which was specified with the **-e** option on the **mqsicreatebroker** and **mqsiaaddbrokerinstance** commands).

Note that the **mqsiremovebrokerinstance** command cannot be issued against a standby broker instance. Stop the active broker instance before you run this command.

Now, remove the multi-instance broker using the **mqsdeletebroker** command. This process removes all references to the broker on both the local and shared work paths.

If you make a mistake, and unintentionally remove a broker instance, it can be re-created using the **mqsiaaddbrokerinstance** command, providing that the **mqsdeletebroker** command has not been used to delete the broker.

If the multi-instance broker has been removed using the **mqsideletebroker** command before removing any associated broker instances, it will not be possible to start the broker instances.

To recover from this situation, re-create the multi-instance broker using the **mqsicreatebroker** command with the **-e** option, specifying the original shared work path location.

The following procedure gives an overview of how you delete a multi-instance broker:

Procedure

1. Stop the multi-instance broker MB1 on client node A.
2. Stop the multi-instance broker MB1 on client node B.
3. Remove the broker instance MB1 on client node B. Do this by using the following command:

```
-mqsiremovebrokerinstance MB1
```

See **mqsiremovebrokerinstance** command for further information.

4. Remove the broker MB1 on client node A. Do this by using the following command:

```
-mqsideletebroker MB1
```

See **mqsideletebroker** command for further information.

Note, that you must remove broker instances created by using the **mqsiaddbrokerinstance** command before you remove the main broker created using the **mqsicreatebroker** command.

Related tasks:

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

“Creating the shared directories” on page 548

How you create the shared directories that you need for your multi-instance IBM Integration Bus.

“Creating the WebSphere MQ multi-instance queue manager” on page 549

How you create the WebSphere MQ multi-instance queue manager that you need for your multi-instance IBM Integration Bus.

Related reference:

 **mqsideletebroker** command

Use the **mqsideletebroker** command to delete a named broker. The command also deletes the queues on the associated queue manager (created when the broker was created). You can also specify that the queue manager is to be deleted.

 **mqsiremovebrokerinstance** command

Use the **mqsiremovebrokerinstance** command to remove a multi-instance broker from a server where IBM Integration Bus has been installed.

Deleting a multi-instance queue manager

To delete a multi-instance queue manager completely, you need to use the WebSphere MQ **dltmqm** command to delete the queue manager, then remove instances from other servers using either the WebSphere MQ **rmvmqinf** or WebSphere MQ **dltmqm** commands.

Run the WebSphere MQ **dltmqm** command to delete a queue manager that has instances defined on other servers, on any server where that queue manager is defined. You do not need to run the WebSphere MQ **dltmqm** command on the same server that you created it on. Then run the WebSphere MQ **rmvmqinf** or WebSphere MQ **dltmqm** command on all the other servers which have a definition of the queue manager.

You can only delete a queue manager when it is stopped. At the time you delete it no instances are running, and the queue manager, strictly speaking, is neither a single or a multi-instance queue manager; it is simply a queue manager that has its queue manager data and logs on a remote share. When you delete a queue manager, its queue manager data and logs are deleted, and the queue manager stanza is removed from the `mq5.ini` file on the server on which you issued the WebSphere MQ **dltmqm** command. You need to have access to the network share containing the queue manager data and logs when you delete the queue manager. On Windows there is no `mq5.ini` file; instead, the queue manager stanza is removed from the registry.

On other UNIX platform servers where you have previously created instances of the queue manager there are also entries in the `mq5.ini` files on those servers. You need to visit each server in turn, and remove the queue manager stanza by running the WebSphere MQ command **rmvmqinf** *Queue manager stanza name*. On Windows, run the same command, and the queue manager stanza is removed from the registry.

On UNIX platforms, if you have placed a common `mq5.ini` file in network storage and referenced it from all the servers by setting the `AMQ_MQS_INI_LOCATION` environment variable on each server, you need to delete the queue manager from only one of its servers as there is only one `mq5.ini` file to update.

Example

First server

```
dltmqm QM1
```

Other servers where instances are defined

```
rmvmqinf QM1, or
```

```
dltmqm QM1
```

Related concepts:

“Creating a multi-instance queue manager” on page 551

Create a multi-instance queue manager by creating the queue manager on one server, and configuring WebSphere MQ on another server to use the shared queue manager data and logs.

Related tasks:

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

“Creating the shared directories” on page 548

How you create the shared directories that you need for your multi-instance IBM Integration Bus.

Listing a multi-instance broker

How you list a multi-instance broker.

Procedure

Use the **mqsilist** command to view multi-instance brokers. Do this by using the following command:

```
mqsilist
```

The output from the **mqsilist** command is of the following format:

```
BIP1284I: Broker 'BK1' on queue manager 'QM1' is running.  
BIP1292I: The multi-instance Broker 'BKHA1' on multi-instance queue manager  
'QMHA1' has stopped.  
BIP1294I  Broker 'BKHA2' is a multi-instance broker running in standby mode  
on multi-instance queue manager 'QMHA2'.  
BIP1295I Broker 'BKHA3' is a multi-instance broker running in active mode on  
multi-instance queue manager 'QMHA3'  
BIP8071I: Successful command completion.
```

Results

The preceding results show:

- A standard broker BK1 running
- A multi-instance broker BKHA1 stopped
- A multi-instance broker BKHA2 started in Standby mode
- A multi-instance broker BKHA3 running in Active mode.

Related tasks:

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

“Creating the shared directories” on page 548

How you create the shared directories that you need for your multi-instance IBM Integration Bus.

“Creating the WebSphere MQ multi-instance queue manager” on page 549

How you create the WebSphere MQ multi-instance queue manager that you need for your multi-instance IBM Integration Bus.

Related reference:



mqsilist command

Use the **mqsilist** command to list installed brokers and their associated resources.

Backing up and restoring a multi-instance broker

How you back up and restore a multi-instance broker.

Before you begin

Before you start:

To ensure that the backup is complete and correct, back up the broker when the broker is stopped, or when it is not processing a configuration change (such as a deployment or property change).

About this task

When you back up a multi-instance broker, you back up the registry and configuration from the shared work path of the broker and its associated instances.

Hence you perform the backup procedure described in this topic for the multi-instance broker only. There are no additional steps for backing up the associated instances of the broker.

When you restore a multi-instance broker, you re-create associated instances by using the **mqsiaaddbrokerinstance** command.

Procedure

1. Back up a multi-instance broker in the standard way by using the **mqsibackupbroker** command. Do this by using the following command:

```
mqsibackupbroker MB1 -d /BackupDirectory/WMB
```

where:
 - MB1 is the name of the broker.
 - /BackupDirectory/WMB is the directory in which the backup file is created.The command detects a multi-instance broker and backs up the registry and configuration from the shared work path of the broker.
There are no additional backup steps for the associated instances of the broker.
2. Restore a multi-instance broker in the standard way by using the **mqsirestorebroker** command. Do this by using the following command:

```
mqsirestorebroker MB1 -d /BackupDirectory/WMB -a 20091009.zip
```

where:
 - MB1 is the name of the broker.
 - /BackupDirectory/WMB is the directory in which the backup file is stored.
 - 20091009.zip is the name of the backup (archive) file.The command detects a multi-instance broker and restores the registry and configuration to the shared work path of the broker.
3. Re-create associated instances of the multi-instance broker by using the **mqsiaaddbrokerinstance** command. See the **mqsiaaddbrokerinstance** command for further information.

Related tasks:

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

Related reference:



mqsibackupbroker command

Use the **mqsibackupbroker** command to back up the current configuration of a broker.



mqsirestorebroker command

Use the **mqsirestorebroker** command to restore the broker configuration from a backup file.

Configuring the global cache for multi-instance brokers

You can configure the global cache to withstand software or hardware failures so that it is available for as much time as possible. Configure a multi-instance broker to host container servers by using an XML policy file.

Before you begin

Before you start:

For more information about the default global cache topology, see Data caching overview.

About this task

You can configure the global cache so that a multi-instance broker hosts up to 4 container servers. If the active broker instance fails, the global cache switches automatically to use the container servers in the standby broker instance.

Consider the following example. Your global cache consists of 2 brokers that host catalog servers and container servers, and a multi-instance broker. The active instance of the multi-instance broker hosts up to 4 container servers. If the active instance of the multi-instance broker fails, the cache will remain operational as long as at least one of the catalog servers is still available. Data is temporarily rebalanced across the remaining container servers in the brokers that host the catalog servers. When the standby instance of the multi-instance broker starts, the container servers rejoin the global cache, and cached data is rebalanced automatically.

A multi-instance broker cannot host a catalog server. Therefore, you cannot configure an integration server to host a catalog server if that integration server is defined with multiple listener hosts.

A sample XML policy file is provided as a starting point for your configuration. The `policy_multi_instance.xml` file configures three brokers in a high availability scenario. Two brokers each host a catalog server, and a multi-instance broker hosts two container servers.

To configure a multi-instance broker, a **listenerHost** element has been introduced as an alternative to the `listenerHost` attribute of the **broker** element. You can use the **listenerHost** element to specify a list of listener hosts. Alternatively, you can set the **listenerHost** property on the integration server to a comma-separated list of listener hosts.

The following steps describe how to configure the global cache for a multi-instance broker.

Procedure

1. Copy the sample policy file, `policy_multi_instance.xml`, from `install_dir/sample/globalcache` to another location on your file system.
Do not edit the sample policy file in its original location; copy it to your own file system first. The original sample policy file might be replaced when you apply maintenance to IBM Integration Bus. You can place a copy of the same policy file on each computer where a broker is running, or you can provide a single copy of the policy file in a shared file system for all brokers to access. No matter how the file is shared between two computers, the policy file must be placed on the same file path on each computer or shared system.
2. Modify the policy file for your system, specifying the appropriate broker names and listener hosts, the port range that the broker is to use, and how many catalog servers the broker hosts. Optionally, you can also specify a domain name for all catalog servers in the embedded cache. If you do not set a domain name, the broker creates one.

Ensure that the policy meets the following criteria:

- You can define 0, 1, or 2 catalog servers for an individual broker, but at least one catalog server must be defined in the policy.

- A multi-instance broker cannot host a catalog server.
- You cannot specify the **listenerHost** element more than once for a broker that is configured to host one or more catalog servers.
- If two brokers share a host name, you must set a distinct port range for each broker.
- Ensure that the port range for each broker includes at least 20 ports.
- The broker names and listener hosts specified in the policy must match the values defined for the brokers.
- You must specify either the **listenerHost** element or the `listenerHost` attribute for each broker.
- You can define only one domain name in the policy file.
- If specified, the domain name must precede the broker elements in the policy file.
- The policy file must be encoded in UTF-8.
- The policy file must contain valid XML. The policy file is validated against an XML schema when you set the broker-level property. You can also validate the policy file against the copy of the schema (`policy.xsd`) that is provided in `install_dir/cachesupport/schema`.

When you use an XML policy file, the broker-level **portRange** property is ignored. The port range specified in the XML file overrides the property specified for the broker.

3. Save the policy file.

4. Set the cache policy to the fully qualified name of the policy file.

The path that you specify must be absolute, not relative. If you use a shared drive on Windows, you must use the `\\hostname\directory` path syntax to the shared drive, instead of a mapped drive letter. The IBM Integration Bus user ID that is used to access the `\\hostname\directory` path must have read access to the file system and must use the same password.

You can set the cache policy by using commands (see *Configuring the embedded global cache by using commands*) or IBM Integration Explorer (see *Configuring the embedded global cache by using IBM Integration Explorer*).

5. Restart each broker.

Results

When each broker restarts, it uses the values in the policy file to determine its cache properties. If multiple listener hosts are specified, the global cache tries to bind to each one in turn until it finds one that is available on the system. If the global cache does not find a listener host that is available on the system, it uses the first listener host in the list.

Each broker contains up to 4 container servers. To find out where container servers are placed, use the **mqsicacheadmin** command to run the **showPlacement** command, as shown in the following example:

```
mqsicacheadmin brokerName -c showPlacement
```

You can also use the **mqsicacheadmin** command to show cache components in a multi-broker cache. For example, the **listHosts** command shows the host names, number of hosts, and number of catalogs in the cache:

```
mqsicacheadmin brokerName -c listHosts
```

Related concepts:

Data caching overview

WebSphere eXtreme Scale provides IBM Integration Bus with data caching capability.

Data caching terminology

The global cache is embedded in the broker. You can also connect to an external WebSphere eXtreme Scale grid.

Global cache scenario: Storing state for integrations

You can use a global cache to store state for integrations. With a global cache, each broker can handle replies, even when the request was processed by another broker.

Global cache scenario: Caching static data

You can use a global cache to store static data. The use of a global cache facilitates horizontal scaling in situations where a cache is used to minimize network interactions to a back end system. With a global cache, you can increase the number of clients while maintaining a predictable response time for each client.

Related tasks:

Configuring the embedded global cache

Configure properties of the embedded global cache by using commands, IBM Integration Explorer, an XML policy file, or the IBM Integration API.

Managing data caching

Store data that you want to reuse by using the embedded global cache or an external WebSphere eXtreme Scale grid.

Accessing the global cache with a JavaCompute node

You can use a JavaCompute node to interact with a map in a global cache or an external WebSphere eXtreme Scale grid.

Monitoring the global cache

You can use the `mqsicacheadmin` command, resource statistics, and the activity log to monitor the global cache. You can use resource statistics and the activity log to monitor external grids.

Related reference:

Parameter values for the cachemanager component

Select the objects and properties that are associated with the global cache that you want to change.

JavaCompute node

Use the JavaCompute node to work with messages by using the Java language.

[WebSphere eXtreme Scale product documentation](#)

[WebSphere eXtreme Scale product web page](#)

Integration Registry configuration for multi-instance integration nodes

You can configure integration nodes to access a shared Integration Registry in a multi-instance integration node environment to withstand software and hardware failures.

In a multi-instance integration node configuration, the Integration Registry is stored in the file system that is shared between the two integration node instances.

Compared to using an Integration Registry in a single integration node configuration, there is some further configuring to use the Integration Registry in a multi-instance integration node environment.

The Integration Registry is hosted inside a runtime integration node, and artifacts that are created in the Integration Registry are uniquely identified by a URL. The Integration Registry assigns URLs as shown in the following example:

```
<protocol>://<Integration Registry hostname>:<Portnumber>/isr/oslc/resource/isr_core/<UUID>
```

As seen in the URL example, in a multi-instance integration node environment, you must make the Integration Registry host name component of the URL known to each integration node, then map the **integrationRegistryHostName** to the IP address of the system that hosts the integration node. You can designate a shared Integration Registry host name for a multi-instance integration node environment by setting the **-4 integrationRegistryHostName** parameter of the **mqsicreatebroker** and **mqsichangebroker** commands.

If the **integrationRegistryHostName** is not explicitly set to the **-4** parameter, the integration node by default uses the host name of the system on which the integration node is running. In a multi-instance integration node environment, the default host name does not work because different integration node instances run on different hosts. For this reason, the **integrationRegistryHostName** must be explicitly set to a host name, which must be mapped on each system to the IP address that hosts the integration node instance.

The systems that host the integration node instances must then be configured to map **integrationRegistryHostName** to the IP address of the system on which the instance is running. You can configure this mapping by adding entries to your operating system's Internet Protocol network configuration hosts file. For more information about modifying these configuration files, refer to your operating system documentation.

See the following diagram for a visual representation of an Integration Registry configuration for multi-instance integration nodes:

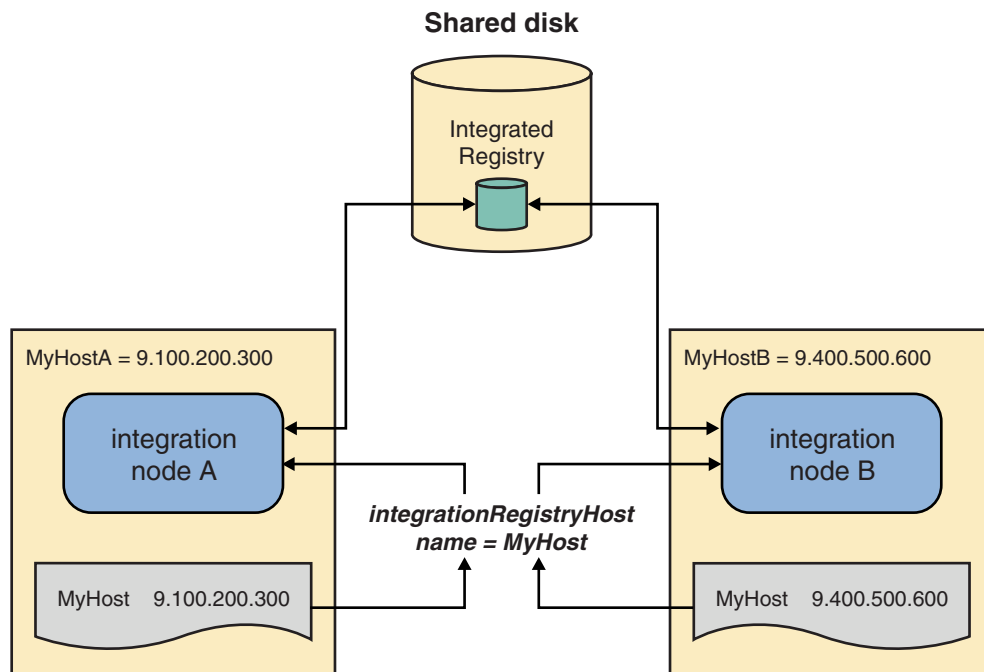


Figure 13. An Integration Registry configuration for multi-instance integration nodes

In the diagram:

1. Integration node A and integration node B are created with a common **integrationRegistryHostname** called *MyHost*.
2. *MyHost* is then mapped to IP address of the system where a integration node instance is running.

Related concepts:



Integration Registry

The IBM Integration Bus provides an Integration Registry, which can store the following items:

Related tasks:

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

Using a broker with an existing high availability manager

You can use WebSphere Message Broker Version 8.0 with an existing high availability manager, for example HACMP, HA/XD, VCS, or HP-UX Serviceguard.

About this task

With the introduction of multi-instance message brokers in WebSphere Message Broker Version 7.0, configuring WebSphere Message Broker Version 8.0 with a high availability (HA) manager is much easier. Previously, support pack IC91 was provided to assist in configuring this requirement.

A number of scripts were provided, but most of these are no longer required because WebSphere Message Broker Version 8.0 has built in many of the functions as a result of the multi-instance work.

This topic summarizes how to complete the following tasks:

1. Create a broker.
2. Add a broker instance.
3. Start a broker.
4. Stop a broker.
5. Monitor a broker.
6. Delete a broker.

Procedure

1. To create a broker, mount the shared resource onto your primary node and use the following **mqsicreatebroker** command with the **-e** parameter to specify your shared resource location.

```
mqsicreatebroker MyBroker -q MQ1 -e /MQHA/MyBroker/
```

where:

- MyBroker is the name of the broker.
- MQ1 is the name of the queue manager.
- /MQHA/MyBroker/ is the directory for your shared resource.

2. To add another broker instance, mount the shared resource onto your secondary nodes and use the following **mqsiaaddbrokerinstance** command.

```
mqsiaaddbrokerinstance MyBroker -e /MQHA/MyBroker/
```

where:

- MyBroker is the name of the broker.
- /MQHA/MyBroker/ is the directory for your shared resource.

3. To start a broker, you can use one of the following script files:

hamqsi_start_broker

```
#!/bin/ksh
# Module:
#   hamqsi_start_broker
#
# Args:
#   BROKER = name of broker to start
#
# Description:
#   This script attempts to start the MQSI Broker
#
#   Runs as the userid which runs broker, and must have
#   the user's environment (i.e. invoke from "su - $MQUSER ..")
#
```

```
BROKER=$1
```

```
if [ -z "$BROKER" ]
then
  echo "hamqsi_start_broker: ERROR! No Broker name supplied"
  echo "   Usage: hamqsi_start_broker <BROKER>"
  exit 1
fi
```

```
# Ensure that the broker is not already running. In this test
# we look for any broker-related processes, which might have
# been left around after a previous failure. Any that remain
# must now be terminated. This is a brutal means of mopping
# up broker processes.
```



```

# We stop the processes in the following order:
# bipservice      - first so it cannot issue restarts
# bipbroker       - next for same reason
# biphttplistener
# startDataFlowEngine
# DataFlowEngine - last
#
echo "hamqsi_start_broker: Ensure $BROKER not already running"
for process in bipservice bipbroker biphttplistener startDataFlowEngine
DataFlowEngine
do
    # Output of kill redirected to /dev/null in case no processes
    ps -ef | grep "$process $BROKER" | grep -v grep | \
        awk '{print $2}' | xargs kill -9 > /dev/null 2>&1
done

# Start the Broker
echo "hamqsi_start_broker: Start Broker " $BROKER
mqsisstart $BROKER > /dev/null 2>&1
if [ $? -ne "0" ]
then
    echo "hamqsi_start_broker: Bad result from mqsisstart for $BROKER"
    exit 1
fi

# Check to see if the broker service has started. This loop
# uses a fixed online timeout of approx. 10 seconds.
TIMED_OUT=yes
i=0
while [ $i -lt 10 ]
do
    # Check for Broker start. We look for bipservice and
    # bipbroker to be running; there might be no message flows
    # deployed.
    # Look to see whether bipservice is running
    cnt=`ps -ef | grep "bipservice $BROKER" | grep -v grep | wc -l`
    if [ $cnt -gt 0 ]
    then
        # Look to see whether bipbroker is running
        cnt=`ps -ef | grep "bipbroker $BROKER" | grep -v grep | wc -l`
        if [ $cnt -gt 0 ]
        then
            # Broker is online
            echo "hamqsi_start_broker: ${BROKER} is running"
            TIMED_OUT=no
            break # out of timing loop
        fi
    fi
    # Manage the loop counter
    i=`expr $i + 1`
    sleep 1
done

# Report error if broker failed to start in time
if [ ${TIMED_OUT} = "yes" ]
then
    echo "hamqsi_start_broker: Broker service failed to start: " $BROKER

```

```

    exit 1
fi

exit 0
hamqsi_start_broker_as
#
#!/bin/ksh
# Module:
#   hamqsi_start_broker_as
#
# Args:
#   broker = broker name
#   qm      = name of broker queue manager
#   mquser = user account under which QM and Broker are run
#
# Description:
#   Starting an MQSI Broker requires the following services:
#   (1) The MQSeries Queue Manager which supports the Broker
#   (2) The MQSI Broker service
#   This script provides a single source to initiate the required
#   services in sequence.
#
# Queue Manager:
#   This script uses the strmqm script supplied by MQSeries V7
#
# Broker:
#   This script then invokes the hamqsi_start_broker script which
#   checks that the broker is fully stopped and then starts it.
#
#   The hamqsi_start_broker_as script should be run as root.

# Check running as root
if [ `id -u` -ne 0 ]
then
    echo "Must be running as root"
    exit 1
fi

BROKER=$1
QM=$2
MQUSER=$3

# Check all parameters exist

if [ -z "$BROKER" ]
then
    echo "hamqsi_start_broker_as: ERROR! No Broker name supplied"
    echo "  Usage: hamqsi_start_broker_as <BROKER> <QM> <MQUSER>"
    exit 1
fi

if [ -z "$QM" ]
then
    echo "hamqsi_start_broker_as: ERROR! No queue manager name supplied"
    echo "  Usage: hamqsi_start_broker_as <BROKER> <QM> <MQUSER>"
    exit 1
fi

```

```

if [ -z "$MQUSER" ]
then
    echo "hamqsi_start_broker_as: ERROR! No Userid supplied"
    echo " Usage: hamqsi_start_broker_as <BROKER> <QM> <MQUSER>"
    exit 1
fi

# -----
# Start the Queue Manager
#
echo "hamqsi_start_broker_as: Start Queue manager " $QM
su $MQUSER -c "/opt/mqm/bin/strmqm $QM"
rc=$?
if [ $rc -ne 0 ]
then
    echo "hamqsi_start_broker_as: Could not start the queue manager"
    exit $rc
fi

# -----
# Start the Broker
#
# Ensure that the Broker is not already running and start the Broker
su - $MQUSER -c "/MQHA/bin/hamqsi_start_broker $BROKER"
rc=$?
if [ $rc -ne 0 ]
then
    echo "hamqsi_start_broker_as: Could not start the broker"
    exit $rc
fi

exit $rc

```

4. To stop a broker, you can use one of the following script files:

hamqsi_stop_broker

```

#!/bin/ksh
# Module:
# hamqsi_stop_broker
#
# Args:
# broker = name of broker
# timeout = max time to allow for each phase of termination
#
# Description:
# This script stops the broker, forcibly if necessary.
# The script should be run by the user account under which
# the broker is run, including environment.

BROKER=$1
TIMEOUT=$2

if [ -z "$BROKER" ]
then
    echo "hamqsi_stop_broker: ERROR! No broker name supplied"
    echo " Usage: hamqsi_stop_broker <BROKER> <TIMEOUT>"
    exit 1
fi

if [ -z "$TIMEOUT" ]
then
    echo "hamqsi_stop_broker: ERROR! No timeout supplied"
    echo " Usage: hamqsi_stop_broker <BROKER> <TIMEOUT>"
    exit 1

```

```

fi

for severity in normal immediate terminate
do
# Issue the stop method in the background - we don't
# want to risk having it hang us up, indefinitely. We
# want to be able to concurrently run a TIMEOUT timer
# to give up on the attempt, and try a more forceful
# stop. If the kill version fails then there is nothing
# more we can do here anyway.

echo "hamqsi_stop_broker: Attempting ${severity} stop of ${BROKER}"
case $severity in

normal)
# Minimum severity of stop is to issue mqsisstop
mqsisstop $BROKER > /dev/null 2>&1 &
;;

immediate)
# This is an immediate stop.
mqsisstop $BROKER -i > /dev/null 2>&1 &
;;

terminate)
# This is a brutal means of mopping up Broker processes.
# We stop the processes in the following order:
#  bipservice      - first so it cannot issue restarts
#  bipbroker       - next for same reason
#  biphttplistener
#  startDataFlowEngine
#  DataFlowEngine - last
for process in bipservice bipbroker biphttplistener startDataFlowEngine
DataFlowEngine
do
# Output of kill redirected to /dev/null in case no processes
ps -ef | grep "$process $BROKER" | grep -v grep | \
awk '{print $2}' | xargs kill -9 > /dev/null 2>&1
done
;;

esac

echo "hamqsi_stop_broker: Waiting for ${severity} stop of ${BROKER} to complete"
TIMED_OUT=yes
SECONDS=0
while (( $SECONDS < $TIMEOUT ))
do
# See whether there are any broker processes still running
cnt=`ps -ef | \
grep -E "bipservice $BROKER|bipbroker $BROKER|startDataFlowEngine
$BROKER|DataFlowEngine $BROKER|biphttplistener $BROKER" | \
grep -v grep | wc -l`
if [ $cnt -gt 0 ]
then
# It's still running...wait for timeout
sleep 1 # loop granularity
else
# It's stopped, as desired
echo "${BROKER} has stopped"
TIMED_OUT=no
break # out of while ..offline timeout loop
fi
done # timeout loop

if [ ${TIMED_OUT} = "yes" ]
then
continue # to next level of urgency

```

```

else
    break          # instance is stopped, job is done
fi

done # next level of urgency

if [ ${TIMED_OUT} = "no" ]
then
    echo "hamqsi_stop_broker: Completed"
    exit 0
else
    echo "hamqsi_stop_broker: Completed with errors"
    exit 1
fi

hamqsi_stop_broker_as

#!/bin/ksh
# Module:
#   hamqsi_stop_broker_as
#
# Arguments are:
#   broker = name of broker
#   qm     = name of broker queue manager
#   mquser = user account under which QM and broker run
#   timeout = max time to allow each phase of stop processing
#
# Description:
#   This script stops the Broker, Queue Manager in that sequence.
#
#   Broker:
#   The script invokes the hamqsi_stop_broker script to stop the
#   broker, which checks that the broker is fully stopped.
#
#   Queue Manager:
#   This script uses the strmqm script supplied by MQSeries V7
#
#   The hamqsi_stop_broker_as script should be run as root.

# Check running as root
if [ `id -u` -ne 0 ]
then
    echo "Must be running as root"
    exit 1
fi

BROKER=$1
QM=$2
MQUSER=$3
TIMEOUT=$4

# Check all parameters

if [ -z "$BROKER" ]
then
    echo "hamqsi_stop_broker_as: ERROR! No Broker name supplied"
    echo "   Usage: hamqsi_stop_broker_as <BROKER> <QM> <MQUSER> <TIMEOUT>"
    exit 1
fi

if [ -z "$QM" ]
then

```

```

    echo "hamqsi_stop_broker_as: ERROR! No queue manager name supplied"
    echo "    Usage: hamqsi_stop_broker_as <BROKER> <QM> <MQUSER> <TIMEOUT>"
    exit 1
fi

if [ -z "$MQUSER" ]
then
    echo "hamqsi_stop_broker_as: ERROR! No userid supplied"
    echo "    Usage: hamqsi_stop_broker_as <BROKER> <QM> <MQUSER> <TIMEOUT>"
    exit 1
fi

if [ -z "$TIMEOUT" ]
then
    echo "hamqsi_stop_broker_as: ERROR! No Timeout value supplied"
    echo "    Usage: hamqsi_stop_broker_as <BROKER> <QM> <MQUSER> <TIMEOUT>"
    exit 1
fi

METHOD_STATUS="OK"

# -----
# Stop the BROKER
#
echo "hamqsi_stop_broker_as: Stop Broker " $BROKER
su - $MQUSER -c "/MQHA/bin/hamqsi_stop_broker $BROKER $TIMEOUT"
if [ $? -ne "0" ]
then
    # Even if the above operation failed, just report and then continue by
    # stopping other components
    echo "hamqsi_stop_broker_as: Attempt to stop broker $BROKER failed"
    METHOD_STATUS="Error"
fi

# -----
# Stop the Queue Manager, using script from MQ V7
#
echo "hamqsi_stop_broker_as: Stop Queue Manager $QM"
su $MQUSER -c "/opt/mqm/bin/endmqm -i $QM"
if [ $? -ne "0" ]
then
    # Even if the above operation failed, just report and then continue by
    # stopping other components
    echo "hamqsi_stop_broker_as: Attempt to stop queue manager $QM failed"
    METHOD_STATUS="Error"
fi

if [ ${METHOD_STATUS} = "OK" ]
then
    exit 0
else
    echo "hamqsi_stop_broker_as: Completed with errors"
    exit 1
fi

```

5. To monitor a broker, you can use the following script file that checks only for the existence of the main broker processes and provides a successful return code if they are found:

hamqsi_monitor_broker_as

```

#!/bin/ksh
# Module:
#   hamqsi_monitor_broker_as
#
# Args:
#   BROKER = name of broker in AppServer
#   QM      = name of queue manager in AppServer
#   MQUSER  = userid under which queue manager and broker run
#
# Description:
#   This is the application monitor script used with HACMP/ES. It
#   needs to be invoked by a parameter-less wrapper script because
#   HACMP does not allow parameters to be passed to application
#   monitor scripts.
#
#   This hamqsi_monitor_broker_as script is run as root, and uses
#   su as needed to monitor the 3 components of the application server.
#
#   This script is tolerant of a queue manager that is still in
#   startup. If the queue manager is still starting this application
#   monitor script will exit with 0 - which indicates
#   to HACMP that there's nothing wrong. This is to allow for
#   startup time for the queue manager which might exceed the
#   Stabilisation Interval set for the Application Monitor in HACMP/ES.
#
# Exit codes:
#   0 => Broker & QM are all running OK or starting
#   >0 => One or more components are not responding.
#
# Check running as root
if [ `id -u` -ne 0 ]
then
    echo "Must be running as root"
    exit 1
fi

BROKER=$1
QM=$2
MQUSER=$3

# Check the parameters

if [ -z "$BROKER" ]
then
    echo "hamqsi_monitor_broker_as: ERROR! No broker name supplied"
    exit 1
fi

if [ -z "$QM" ]
then
    echo "hamqsi_monitor_broker_as: ERROR! No queue manager name supplied"
    exit 1
fi

if [ -z "$MQUSER" ]
then
    echo "hamqsi_monitor_broker_as: ERROR! No mquser supplied"
    exit 1
fi

# Use a state variable to reflect the state of components as they
# are tested. Valid values are "stopped", "starting" and "started"
# Initialise it to "stopped" for safety.
STATE="stopped"

# -----
# Check that the queue manager is running or starting.
#
su - $MQUSER -c "echo 'ping qmgr' | runmqsc ${QM}" > /dev/null 2>&1
pingresult=$?
# pingresult will be 0 on success; non-zero on error (man runmqsc)
if [ $pingresult -eq 0 ]
then

```



```

# ping succeeded
echo "hamqsi_monitor_broker_as: Queue Manager ${QM} is responsive"
STATE="started"
else
# ping failed
# Don't condemn the QM immediately, it might be in startup.
# The following regexp includes a space and a tab, so use tab-friendly
# editors.
srchstr=" $QM[ ]*$"
cnt=`ps -ef | grep strmqm | grep "$srchstr" | grep -v grep |
    | awk '{print $2}' | wc -l`
if [ $cnt -gt 0 ]
then
# It appears that QM is still starting up, tolerate
echo "hamqsi_monitor_broker_as: Queue Manager ${QM} is starting"
STATE="starting"
else
# There is no sign of QM start process
echo "hamqsi_monitor_broker_as: Queue Manager ${QM} is not responsive"
STATE="stopped"
fi
fi

# Decide whether to continue or to exit
case $STATE in
stopped)
echo "hamqsi_monitor_broker_as: Queue manager ($QM) is not running correctly"
exit 1
;;
starting)
echo "hamqsi_monitor_broker_as: Queue manager ($QM) is starting"
echo "hamqsi_monitor_broker_as: WARNING - Stabilisation Interval might be too short"
echo "hamqsi_monitor_broker_as: WARNING - No test of broker $BROKER will be conducted"
exit 0
;;
started)
echo "hamqsi_monitor_broker_as: Queue manager ($QM) is running"
continue
;;
esac

# -----
# Check the MQSI Broker is running
#
# Re-initialise STATE for safety
STATE="stopped"
#
# The broker runs as a process called bipservice which is responsible
# for starting and re-starting the admin agent process (bipbroker).
# The bipbroker is responsible for starting any DataFlowEngines. The
# bipbroker starts the DataFlowEngines using the wrapper script
# startDataFlowEngine. If no integration servers have been assigned to
# the broker there will be no DataFlowEngine processes. There should
# always be a bipservice and bipbroker process pair. This monitor
# script only tests for bipservice, because bipservice should restart
# bipbroker if necessary - the monitor script should not attempt to
# restart bipbroker and it might be premature to report an absence
# of a bipbroker as a failure.
#
cnt=`ps -ef | grep "bipservice $BROKER" | grep -v grep | wc -l`
if [ $cnt -eq 0 ]
then
echo "hamqsi_monitor_broker_as: MQSI Broker $BROKER is not running"
STATE="stopped"
else
echo "hamqsi_monitor_broker_as: MQSI Broker $BROKER is running"
STATE="started"
fi

# Decide how to exit
case $STATE in
stopped)
echo "hamqsi_monitor_broker_as: Broker ($BROKER) is not running correctly"
exit 1

```

```

;;
started)
  echo "hamqsi_monitor_broker_as: Broker ($BROKER) is running"
  exit 0
;;
esac

```

If you require more information than that supplied by the preceding example, you can code your monitor to do the following actions:

- Subscribe to IBM Integration Bus accounting and statistics, and analyze the results.
 - Put a dummy message through the broker and analyze the results.
6. Before you delete the broker on the primary node, you must delete any brokers on the standby nodes. To delete a broker, use the **mqsiremovebrokerinstance** command on the secondary nodes, and the **mqsdeletebroker** command on the primary node.

Results

For further information about configuring WebSphere MQ with a high availability manager, refer to the WebSphere MQ Version 7 product documentation online.

See the:

- **addmqinf** command
- **dspmqinf** command
- **rmvmqinf.** command

and the **-md** option on the **crtmqm** command.

Related tasks:

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

Related reference:

 **mqscreatebroker** command

Use the **mqscreatebroker** command to create a broker and its associated resources.

 **mqsaddbrokerinstance** command

Use the **mqsaddbrokerinstance** command to create a multi-instance broker on a server where IBM Integration Bus has been installed.

 **mqsiremovebrokerinstance** command

Use the **mqsiremovebrokerinstance** command to remove a multi-instance broker from a server where IBM Integration Bus has been installed.

 **mqsdeletebroker** command

Use the **mqsdeletebroker** command to delete a named broker. The command also deletes the queues on the associated queue manager (created when the broker was created). You can also specify that the queue manager is to be deleted.

Related information:

 [WebSphere MQ Version 7 product documentation](#)

Using a broker with an existing Windows Cluster (Windows Server)

You can use IBM Integration Bus with the existing high availability manager for Windows Server (Failover Cluster Manager).

About this task

For more information about the versions of Windows Server that are supported by IBM Integration Bus, see the IBM Integration Bus Requirements web page.

This topic summarizes how to complete the following tasks:

1. Complete the prerequisite setup.
2. Configure a local group.
3. Create a broker.
4. Add broker instances to the additional nodes.
5. Add the broker service to the cluster configuration
6. Start and stop a broker.
7. Delete a broker.

Procedure

1. To complete the prerequisite setup, complete the following steps:
 - a. Validate your Failover Cluster Manager configuration.
 - b. Complete all of the steps documented in 'Supporting the Microsoft Cluster Service' of the WebSphere MQ documentation to create a cluster configuration that contains the queue manager on which you want the broker to run.
 - c. Note the `haregtyp.exe` command as a prerequisite for creating WebSphere MQ resources in the cluster.
2. To configure a local group, ensure that the domain user under which you want the broker to run exists in the local `mqrkrs` group on all nodes on which you want the broker to run.
3. To create a broker, use the following `mqsicreatebroker` command on the node on which your cluster is currently running.

```
mqsicreatebroker MyBroker -q MQ1 -e E:\Broker\Workspace
```

where:
 - MyBroker is the name of the broker.
 - MQ1 is the name of the queue manager.
 - E:\Broker\Workspace is the directory of a shared disk (nonquorum) in your cluster configuration.
4. To add this broker instance to the other nodes in your cluster, switch your cluster to each node in turn. When a node is active, use the following `mqsiaaddbrokerinstance` command.

```
mqsiaaddbrokerinstance MyBroker -e E:\Broker\Workspace
```

where:
 - MyBroker is the name of the broker.
 - E:\Broker\Workspace is the directory of a shared disk (nonquorum) in your cluster configuration.
5. To add a broker generic service resource to the cluster which contains the broker queue manager, complete the following steps:

- a. Select the IBM IBM Integration Bus component MyBroker service when asked. All other settings can be left unchanged.
 - b. Add a dependency on the WebSphere MQ resource, to ensure that the queue manager is started before the broker.
6. To start and stop the broker resource, use Failover Cluster Manager.
 7. To delete a broker resource from the Failover Cluster Manager, take the following steps:
 - a. Delete the broker generic service from the cluster configuration.
 - b. Use the **mqsiremovebrokerinstance** command on all nodes but one, moving the cluster between nodes before running each command.
 - c. On the final node, use the **mqsdeletebroker** command to completely remove the broker configuration.

Results

For further information about configuring WebSphere MQ with a high availability manager for Windows Server, see the WebSphere MQ Version 7 product documentation online.

See the following topics:

- Introducing MSCS clusters
- Supporting the Microsoft Cluster Service (MSCS)
- Setting up WebSphere MQ for MSCS clustering

Related tasks:

“Using a broker with an existing high availability manager” on page 571
 You can use WebSphere Message Broker Version 8.0 with an existing high availability manager, for example HACMP, HA/XD, VCS, or HP-UX Serviceguard.

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

Related reference:

 **mqscreatebroker** command

Use the **mqscreatebroker** command to create a broker and its associated resources.

 **mqsaddbrokerinstance** command

Use the **mqsaddbrokerinstance** command to create a multi-instance broker on a server where IBM Integration Bus has been installed.

 **mqsremovebrokerinstance** command

Use the **mqsremovebrokerinstance** command to remove a multi-instance broker from a server where IBM Integration Bus has been installed.

 **mqsdeletebroker** command

Use the **mqsdeletebroker** command to delete a named broker. The command also deletes the queues on the associated queue manager (created when the broker was created). You can also specify that the queue manager is to be deleted.

Related information:

 [WebSphere MQ Version 7 product documentation](#)

HTTP proxy servlet overview

By using the HTTP proxy servlet in a servlet container, you can support high availability, load distribution, access to the broker from multiple IP addresses and ports, and a larger number of concurrent HTTP sessions.

The HTTP proxy servlet is a Java servlet that you can use in a servlet container, such as WebSphere Application Server Liberty Profile or Apache Tomcat, or in a web server such as IBM WebSphere Application Server. The HTTP proxy servlet receives HTTP requests from web services client applications and replaces the support that is provided by the broker and embedded (integration server) HTTP listeners.

The HTTP proxy servlet supports SSL (HTTPS) secure protocol when it is deployed in a servlet container that is configured to support SSL.

If you are deploying a web browser-based JavaScript application that uses the JavaScript client API to call an IBM Integration Bus integration service, then you must deploy the proxy servlet on the same web application server as your JavaScript application. For more information about the JavaScript client API, see Integration service JavaScript client API .

You cannot use the HTTP proxy servlet if you configure your broker environment to use multi-instance WebSphere MQ queue managers because the HTTP proxy servlet cannot connect to the standby queue manager when it becomes active.

For a detailed description of the HTTP proxy servlet, see “HTTP proxy servlet operation.”

Before you install and test the HTTP proxy servlet, ensure that you understand the following concepts:

- Working with HTTP flows
- HTTP listeners
- “HTTP proxy servlet operation”
- “Importance of the context root when you configure the HTTP proxy servlet” on page 590

When you have gained an understanding of the HTTP proxy servlet concepts, read the following topics to help you configure, install, and test the HTTP proxy servlet:

- “Configuring the HTTP proxy servlet” on page 592
- “Deploying the HTTP proxy servlet on a servlet container” on page 602
- “Testing the HTTP proxy servlet” on page 608

Related tasks:

“Using multi-instance brokers” on page 544

Configure a broker to run in multi-instance mode when you require high availability using WebSphere MQ.

HTTP proxy servlet operation

Before you deploy the HTTP proxy servlet, ensure that you understand how the HTTP proxy servlet manages the communication between IBM Integration Bus applications and web services clients in a number of supported configurations.

The HTTP proxy servlet is a Java Web Application Archive (.war) file that is part of the runtime environment in WebSphere Message Broker Version 6.1 Fix Pack 3 (6.1.0.3) and above, and can be found in the following directory:

```
IIB_runtime_install_path/tools
```

where *IIB_runtime_install_path* specifies the name of your runtime installation directory.

The HTTP proxy servlet is a Java servlet that receives HTTP requests. The HTTP proxy servlet matches the received web address with the web address that the HTTP or SOAP input nodes are monitoring, then passes the HTTP request message to the correct HTTP or SOAP input node flow by using WebSphere MQ.

The HTTP proxy servlet receives response messages from the HTTP or SOAP reply nodes and sends them back to the client applications over HTTP or HTTPS. The broker has several internal WebSphere MQ queues (SYSTEM.BROKER.WS.* queues), that are used for the communication between the HTTP proxy servlet and the HTTP or SOAP input and reply nodes.

The HTTP proxy servlet is deployed in a servlet container.

A *servlet container* (or web container) is the runtime environment for servlets and Java Server Pages (JSP). Apache Tomcat is an example of a servlet container. WebSphere Application Server is an example of a web server, which includes the functionality of a servlet container. The HTTP proxy servlet can be deployed in a local servlet container that is running on the same machine as IBM Integration Bus or on a remote servlet container that is running on a different machine to IBM Integration Bus. The servlet container must allow the HTTP proxy servlet to configure and call the WebSphere MQ classes for Java.

After the HTTP proxy servlet is deployed and running on the servlet container, it uses the HTTP listener of the servlet container to receive HTTP requests. If the servlet container is configured to support SSL (HTTPS), web services requests are received by the message flows by using SSL.

The following topics describe some common scenarios that are based on typical IBM Integration Bus deployments, and the HTTP proxy servlet configuration parameters that are required.

- The HTTP proxy servlet is on the same machine as IBM Integration Bus
- The HTTP proxy servlet is on a different machine to IBM Integration Bus
- The HTTP proxy servlet is deployed in an environment with a network local-balancer

HTTP proxy servlet is deployed to a servlet container on the same machine as IBM Integration Bus:

You can configure the HTTP proxy servlet for a deployment where the HTTP proxy servlet is deployed to a servlet container on the same machine as IBM Integration Bus.

In the following figure, the HTTP proxy servlet is running on the same machine as the broker. The HTTP proxy servlet connects to the broker queue manager by using bindings mode (local connection).

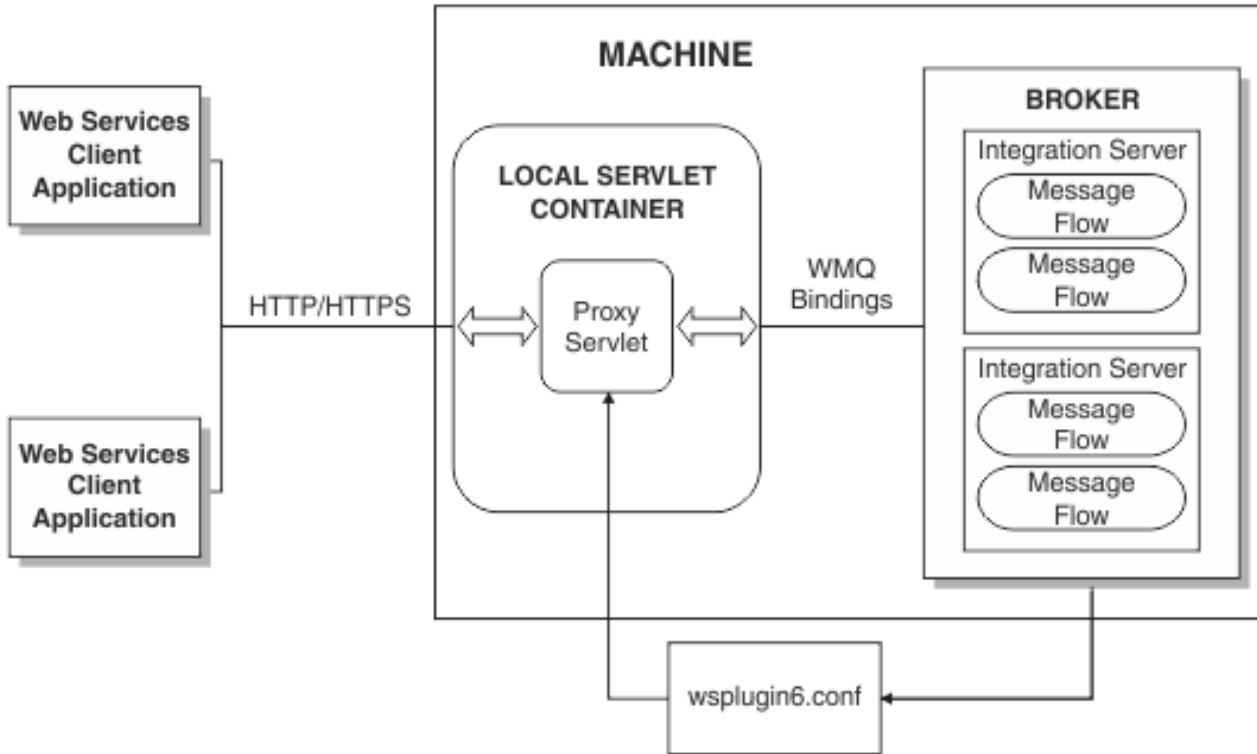


Figure 14. HTTP proxy servlet deployed to a servlet container on the same machine as IBM Integration Bus

In this configuration example, you must configure the following HTTP proxy servlet configuration parameters. For information about configuring the HTTP proxy servlet, see “Configuring the HTTP proxy servlet” on page 592.

- Set **useClientMode** to false.
- Set **useQueueManagerDataInsteadOfConfigFile** to "".
- Set **configFile** to the location of the configuration file, for example, C:\ProgramData\IBM\MQSI\components\IB9NODE\config\wsplugin6.conf.

Before you configure, deploy, and test the HTTP proxy servlet, ensure that you understand the following concepts:

- Working with HTTP flows
- HTTP listeners
- “HTTP proxy servlet operation” on page 584
- “Importance of the context root when you configure the HTTP proxy servlet” on page 590

When you have gained an understanding of the HTTP proxy servlet concepts, read the following topics to help you configure, deploy, and test the HTTP proxy servlet:

- “Configuring the HTTP proxy servlet” on page 592
- “Deploying the HTTP proxy servlet on a servlet container” on page 602
- “Testing the HTTP proxy servlet” on page 608

HTTP proxy servlet is deployed to a servlet container that is on a different machine to IBM Integration Bus:

You can configure the HTTP proxy servlet for a deployment where the HTTP proxy servlet is deployed to a servlet container on a different machine to IBM Integration Bus, with a WebSphere MQ client link to the broker queue manager.

In the following figure, the HTTP proxy servlet is running on a remote machine to the broker. The HTTP proxy servlet connects to the broker queue manager by using a WebSphere MQ client connection, the HTTP proxy servlet can be configured to access HTTP or SOAP nodes, and the HTTP or SOAP node configuration is retrieved from the SYSTEM.BROKER.WS.ACK queue.

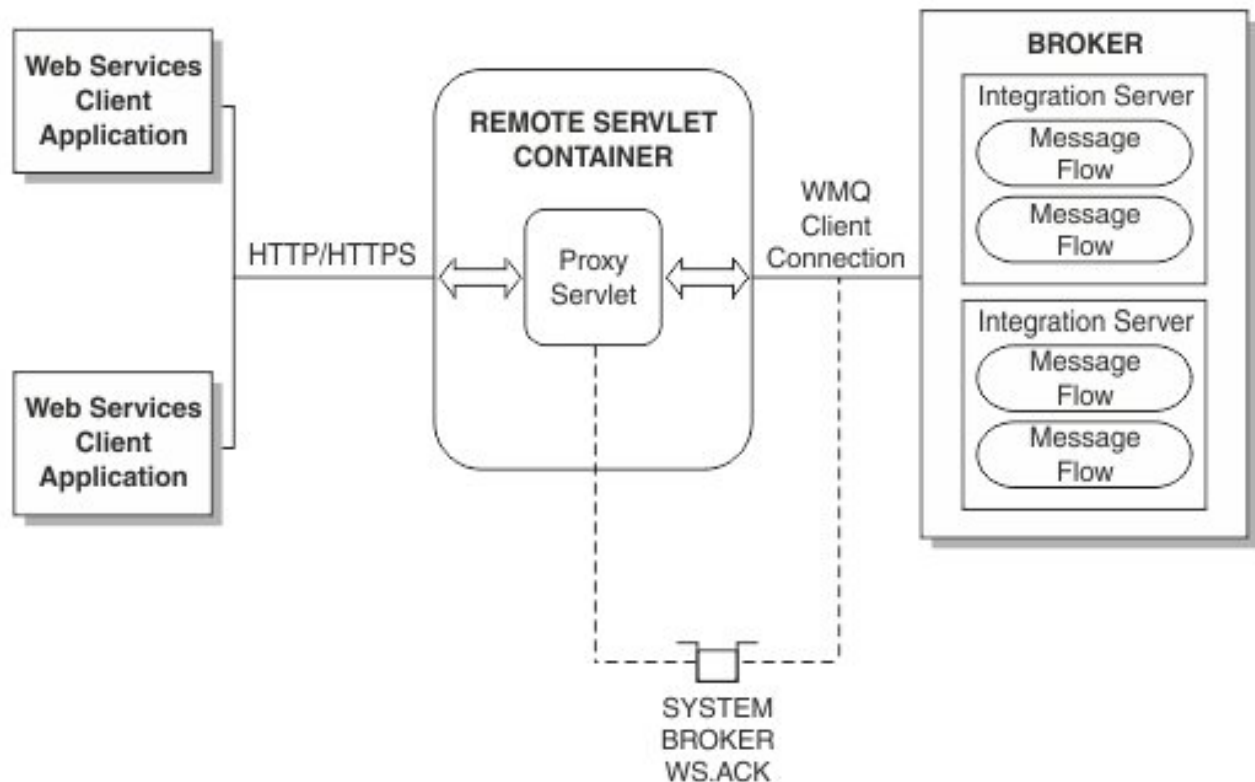


Figure 15. HTTP proxy servlet deployed to a servlet container that is on a different machine to IBM Integration Bus

In this configuration example, you must configure the following HTTP proxy servlet configuration parameters. For information about configuring the HTTP proxy servlet, see “Configuring the HTTP proxy servlet” on page 592.

- Set **useClientMode** to true.
- Set **useQueueManagerDataInsteadOfConfigFile** to * or the broker queue manager name.
- Set **clientModeHostname**, **clientModeChannelName**, and **clientModePortNumber** to the correct values as detailed in “HTTP proxy servlet configuration parameters” on page 594.

The HTTP proxy servlet attempts to connect to the remote queue manager for the broker by reading the required configuration data from the WebSphere MQ client connection. For the connection to succeed, you must start the

SYSTEM.DEFAULT.LISTENER.TCP listener in the remote queue manager. For more information, search for *START LISTENER* in the WebSphere MQ information center.

Before you configure, deploy, and test the HTTP proxy servlet, ensure that you understand the following concepts:

- Working with HTTP flows
- HTTP listeners
- “HTTP proxy servlet operation” on page 584
- “Importance of the context root when you configure the HTTP proxy servlet” on page 590

When you have gained an understanding of the HTTP proxy servlet concepts, read the following topics to help you configure, deploy, and test the HTTP proxy servlet:

- “Configuring the HTTP proxy servlet” on page 592
- “Deploying the HTTP proxy servlet on a servlet container” on page 602
- “Testing the HTTP proxy servlet” on page 608

Related information:

 [WebSphere MQ Version 7.5 product documentation](#)

HTTP proxy servlet is deployed to a servlet container that is on a different machine to IBM Integration Bus with a network load-balancer for distributing work to several brokers:

You can configure the HTTP proxy servlet for a deployment where the HTTP proxy servlet is deployed to a servlet container that is on a different machine to IBM Integration Bus and has a WebSphere MQ client link to the broker queue manager. In this configuration a network load-balancer is used to distribute work to several brokers.

In the following figure the HTTP proxy servlet is configured to load balance WebSphere MQ connections across multiple brokers. A network load balancer is required for this configuration.

When the HTTP proxy servlet is configured to connect to multiple brokers, the brokers must be identical clones of each other, which means that the same HTTP and SOAP flows are deployed with the same web addresses.

The HTTP proxy servlet sends the HTTP requests over the WebSphere MQ client connection to distribute the load between the active broker connections.

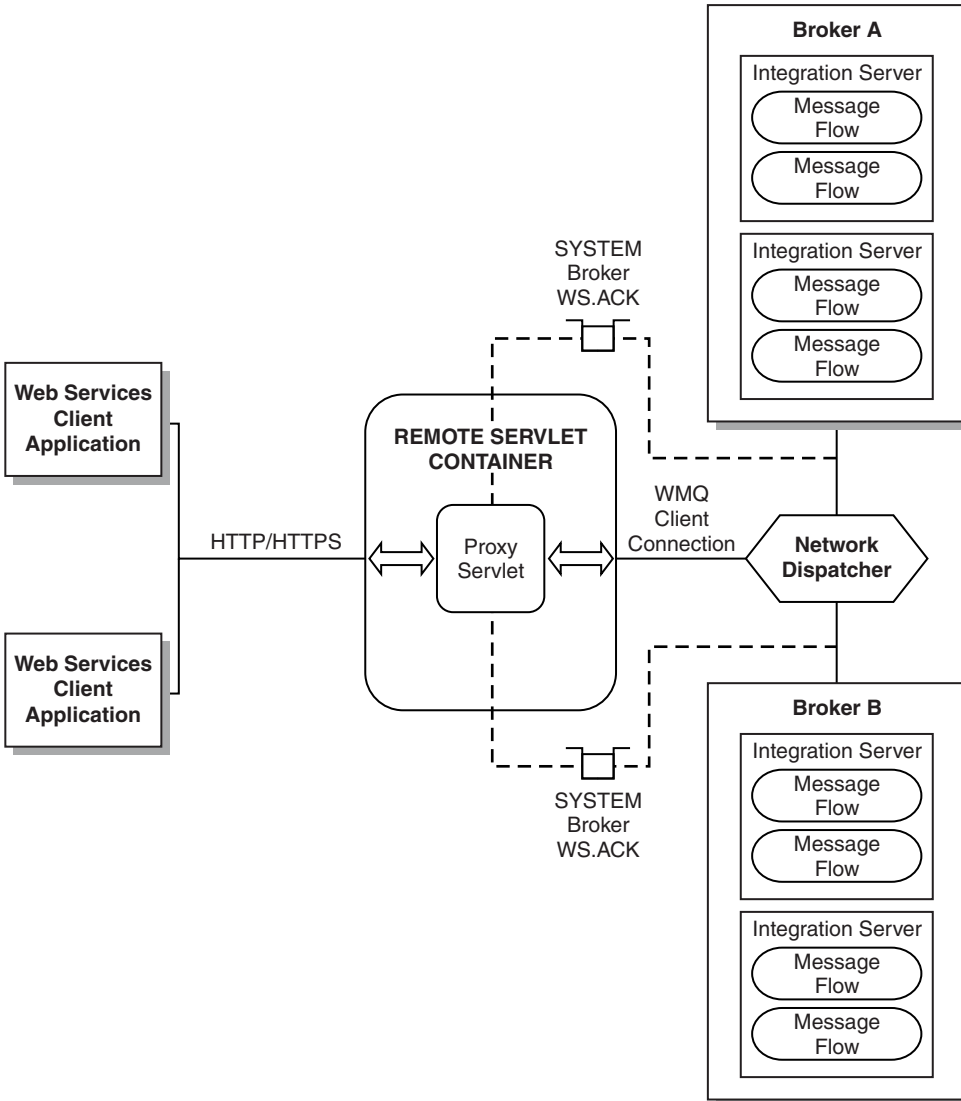


Figure 16. HTTP proxy servlet deployed to a servlet container that is on a different machine to IBM Integration Bus with a network load-balancer for distributing work to several brokers

The configuration is the same as that described in “HTTP proxy servlet is deployed to a servlet container that is on a different machine to IBM Integration Bus” on page 587, but the network load-balancer replaces the broker's machine. Configuration files cannot be used because there are several brokers behind one virtual IP address, and each one has a different configuration file. The HTTP proxy servlet loads information for each connection, and uses the relevant configuration information for each broker.

In this configuration example, you must configure the following HTTP proxy servlet configuration parameters. For information about configuring the HTTP proxy servlet, see “Configuring the HTTP proxy servlet” on page 592.

- Set **useClientMode** to true.
- Set **useQueueManagerDataInsteadOfConfigFile** to * or the broker queue manager name.

- Set **clientModeHostname**, **clientModeChannelName**, and **clientModePortNumber** to the correct values as detailed in “HTTP proxy servlet configuration parameters” on page 594.

If failover is one of the reasons for deploying this configuration, it is recommended that you configure the following additional HTTP proxy servlet configuration parameters.

- Set **clientModeConnectRetryCount** to a value equal or higher than the number of broker. This setting ensures that a single failed server does not cause intermittent errors, even if the load-balancer does simple round-robin scheduling. The HTTP proxy servlet uses the first available broker.
- Set **reconnectActiveLinksAge** to a value less than the firewall timeout. This setting prevents the reuse of old connections that might have been discarded by firewalls between the HTTP proxy servlet and the load-balancer (or between the load-balancer and the broker).

You can set **testConnectionBeforeReuse** to true as an alternative way to manage dropped WebSphere MQ connections between the HTTP proxy servlet and broker queue managers. However, this option causes an MQINQ call to be performed before any attempt to send data to the broker. If the MQINQ call fails, a new connection is established, and the data is sent over the new connection. Because the configuration adds another operation to the MQPUT and MQGET calls, it results in significant processing for every message; use this option only if no alternative options are available. For information about MQINQ, MQPUT, and MQGET calls, search for *Function calls* in the WebSphere MQ information center.


Before you configure, deploy, and test the HTTP proxy servlet, ensure that you understand the following concepts:

- Working with HTTP flows
- HTTP listeners
- “HTTP proxy servlet operation” on page 584
- “Importance of the context root when you configure the HTTP proxy servlet”

When you have gained an understanding of the HTTP proxy servlet concepts, read the following topics to help you configure, deploy, and test the HTTP proxy servlet:

- “Configuring the HTTP proxy servlet” on page 592
- “Deploying the HTTP proxy servlet on a servlet container” on page 602
- “Testing the HTTP proxy servlet” on page 608

Related information:

 [WebSphere MQ Version 7.5 product documentation](#)

Importance of the context root when you configure the HTTP proxy servlet

You must assign a context root to the HTTP proxy servlet when you deploy the HTTP proxy servlet to a servlet container. The choice of context root determines the message flow nodes or integration service URLs with which the HTTP proxy servlet can communicate.

Web addresses, or Universal Resource Locators (URLs), have an important role when HTTP or SSL (HTTPS) protocols are used.

The HTTP proxy servlet passes the requests from the servlet container to the broker and vice versa.

Each HTTP or SOAP input node expects to receive requests from a specific web address (or web addresses when wildcard characters are used). The servlet container also uses the web address to locate the servlets that are going to process the HTTP or HTTPS requests that the servlet container receives on behalf of the broker.

The web address is structured in the following way:

schema://host_name:port/url_path

where:

schema

Specifies the protocol (HTTP or HTTPS).

host_name

Specifies the host name or IP address of the server where the servlet container is running.

port

Specifies the port number on which the servlet container is listening.

url_path

Specifies the URL path; a series of tokens (separated by slashes /) that identify both the HTTP proxy servlet, and the HTTP or SOAP input nodes.

When you use the HTTP proxy servlet, the URL path is partitioned as follows:

context_root/node_url_path

where:

context_root

Specifies the part of the URL path (context root) that is allocated to the proxy servlet by the servlet container when the proxy servlet is deployed.

node_url_path

Specifies the part of the URL path that makes the web address unique to a specific HTTP or SOAP input node.

Note: You can set *context_root* to / in which case *url_path* and *node_url_path* are the same.

The entire URL path must be configured in the properties of the HTTPInput or SOAPInput node. For more information, see HTTPInput node and SOAPInput node.

For example, you might have HTTPInput nodes that are configured to receive requests for the following web addresses:

- Node1: http://myhost.com/app1
- Node2: http://myhost.com/public/app2
- Node3: http://myhost.com/public/app3
- Node4: http://myhost.com/private/app4

If you set the context root for your HTTP proxy servlet to /public, then you can communicate with Node2 and Node3 via the HTTP proxy servlet.

If you set the context root for your HTTP proxy servlet to `/private`, then you can communicate with Node4 via the HTTP proxy servlet.

If you set the context root for your HTTP proxy servlet to `/`, then you can communicate with all of the HTTPInput nodes via the HTTP proxy servlet.

Choosing a context root for the HTTP proxy servlet

- If you have many existing applications that you want to access with the HTTP proxy servlet, and the input nodes or integration services do not use URLs with a common root, you might want to set the context root for the HTTP proxy servlet to `/`. Then you do not need to modify the URLs used by your input nodes or integration services. However, if you set the context root for the HTTP proxy servlet to `/`, then the HTTP proxy servlet becomes the default application for the servlet container, and the HTTP proxy servlet attempts to process all requests that do not match the context root of any other applications that are hosted by the servlet container.
- If you want to access a subset of your existing applications with the HTTP proxy servlet, you might configure the URLs for the input nodes or integration services to start with a consistent value, for example `/public`. Then you must set the context root of the HTTP proxy servlet to the same value.
- If you are building applications from scratch, then consider the requirements of the HTTP proxy servlet when you plan the structure of the URLs you configure for your input nodes or integration services.

Before you install and test the HTTP proxy servlet, ensure that you understand the following concepts:

- Working with HTTP flows
- HTTP listeners
- “HTTP proxy servlet operation” on page 584

When you have gained an understanding of the HTTP proxy servlet concepts, read the following topics to help you configure, install, and test the HTTP proxy servlet:

- “Configuring the HTTP proxy servlet”
- “Deploying the HTTP proxy servlet on a servlet container” on page 602
- “Testing the HTTP proxy servlet” on page 608

Configuring the HTTP proxy servlet

Configure the HTTP proxy servlet with the details of the broker environment to which the HTTP proxy servlet connects. The HTTP proxy servlet must be configured before you deploy the HTTP proxy servlet to a servlet container.

Before you begin

Before you start:

In order that the HTTP proxy servlet can access the SOAPInput and SOAPReply nodes, you must enable the broker listener for each integration server where message flows with SOAP nodes are deployed. See HTTP listeners and Switching from embedded listeners to a broker-wide listener.

About this task

The HTTP proxy servlet proxyservlet.war file is part of the runtime environment in WebSphere Message Broker Version 6.1 Fix Pack 3 (6.1.0.3) and above, and can be found in the following directory:

IIB_installation_path/tools

where *IIB_installation_path* specifies the name of your runtime installation directory.

Complete the following steps to configure the web deployment descriptor (web.xml) for the HTTP proxy servlet by using the IBM Integration Toolkit.

Procedure

1. From the IBM Integration Toolkit menu, switch to the Java EE perspective by clicking **Window > Open Perspective > Other** and clicking **Java EE**.
2. Click **File > Import**, expand the **Web** section, select **WAR file** in the list, and click **Next**.
3. Click **Browse** to find the proxyservlet.war file in *IIB_runtime_install_path/tools*, where *IIB_runtime_install_path* specifies the name of your IBM Integration Bus runtime installation path (for example, C:\Program Files\IBM\MQSI\9.0.0.0\tools\proxyservlet.war), and click **Open**.
4. Set the name of the Web project to proxyservlet and click **Finish**. The HTTP proxy servlet is now ready for configuring by using the Java EE perspective.
5. In the Enterprise Explorer view, expand **proxyservlet** and double-click **Deployment Descriptor** to view the web deployment descriptor.
6. Find the **Servlets and JSPs** section in the Web Deployment Descriptor, and click the servlet link called **WBIMBServlet** to display the servlet web address mappings and initialization parameters. The same parameters in the web.xml file can be configured through JNDI in WebSphere Application Server. This alternative method means that you set up at the application server side only once for any future deployment of the proxy servlet. This operation is possible because the JNDI configuration parameters take precedence over the initialization parameters in the web.xml file. For more information about setting up the JNDI interface for the proxy servlet, see “Setting up the JNDI interface for the proxy servlet” on page 599
7. Click the **Source** tab, which is found at the bottom of the Deployment Descriptor view. The source of the web deployment descriptor (web.xml) displays the HTTP proxy servlet parameters.
8. Edit the HTTP proxy servlet parameters as required; see “HTTP proxy servlet configuration parameters” on page 594.
9. When the configuration is complete, save the changes to the web.xml file by pressing **Ctrl+S**.
10. Export the configured HTTP proxy servlet ready for deployment to a servlet container by completing the following steps:
 - a. In the Enterprise Explorer view, right-click the **Deployment Descriptor** for the proxyservlet Web project, and click **Export > WAR file**. The WAR Export dialog window is displayed.
 - b. Click **Browse** and specify a location and a name for the configured WAR file. For example, myproxyservlet.war.
 - c. Click **Save** and then click **Finish**.

Results

Results:

You have now configured the HTTP proxy servlet with the initialization parameters.

What to do next

Next steps:

Deploy and configure the HTTP proxy servlet on the web application server that will host the JavaScript application, see “Deploying the HTTP proxy servlet on a servlet container” on page 602.

Related concepts:

“HTTP proxy servlet overview” on page 584

By using the HTTP proxy servlet in a servlet container, you can support high availability, load distribution, access to the broker from multiple IP addresses and ports, and a larger number of concurrent HTTP sessions.

“HTTP proxy servlet operation” on page 584

Before you deploy the HTTP proxy servlet, ensure that you understand how the HTTP proxy servlet manages the communication between IBM Integration Bus applications and web services clients in a number of supported configurations.

Related tasks:

“Deploying the HTTP proxy servlet on a servlet container” on page 602

Load and install the HTTP proxy servlet file on a servlet container, such as Apache Tomcat or WebSphere Application Server Liberty Profile, or a web server, such as WebSphere Application Server.

“Testing the HTTP proxy servlet” on page 608

Test the HTTP proxy servlet with a broker that receives HTTP requests and passes them to a message flow.

“Setting up the JNDI interface for the proxy servlet” on page 599

The JNDI interface for the proxy servlet requires a one time setup of the WebSphere Application Server full profile.

HTTP proxy servlet configuration parameters:

Before you can deploy the HTTP proxy servlet to the servlet container, you must configure it with the following initialization parameters for the broker environment to which the HTTP proxy servlet connects.

The parameters that you can configure for the HTTP proxy servlet are detailed in the following sections. For information about how to configure the HTTP proxy servlet, see “Configuring the HTTP proxy servlet” on page 592.

- “General options” on page 595
- “Information options” on page 596
- “ReplyToQ and QMgr options” on page 596
- “SSL connection options” on page 596
- “MQ connection options” on page 597

The following topics describe some common scenarios and the HTTP proxy servlet configuration parameters that are required.

- “HTTP proxy servlet is deployed to a servlet container on the same machine as IBM Integration Bus” on page 585

- “HTTP proxy servlet is deployed to a servlet container that is on a different machine to IBM Integration Bus” on page 587
- “HTTP proxy servlet is deployed to a servlet container that is on a different machine to IBM Integration Bus with a network load-balancer for distributing work to several brokers” on page 588

General options:

Parameter name	Default value	Description
brokerName	*	<p>Set to * or the name of your broker.</p> <p>Use this parameter to set the name that is used for error messages; the value is auto-detected if set to "".</p> <p>Set a value if several brokers are being used by the HTTP proxy servlet, and a single name is required for error messages.</p>
configFilePath	/var/mqsi/components/IB9NODE/config/wsplugin6.conf	<p>Set to the full path of the configuration file.</p> <p>If the broker that is used by the HTTP proxy servlet is local, set this parameter to the wsplugin6.conf file.</p> <p>This value is used only when the parameter useQueueManagerDataInsteadOfConfigFile is set to blank. The configuration file is used only when the HTTP proxy servlet is running on the same server as the broker, and it has access to the file.</p> <p>In Windows, the file is stored in C:\install_dir\config/wsplugin.conf or C:\Documents and Settings\All Users\IBM\MQSI\components\broker_name\config/wsplugin6.conf where <i>broker_name</i> is the name of your broker.</p> <p>On Linux and UNIX, the file is stored in /var/mqsi/config/wsplugin.conf or in /var/mqsi/components/broker_name/config/wsplugin6.conf where <i>broker_name</i> is the name of your broker.</p>
useFastpathBindingsConnection	false	<p>Set to true or false.</p> <p>Set the value to true to make the HTTP proxy servlet connect in fastpath mode if the HTTP proxy servlet is using a local queue manager.</p>
traceFileName		<p>Set to the full path of the trace file.</p> <p>If this parameter is not specified the trace is sent to stdout.</p>
turnTraceOn	0	<p>Set to 0, 1, or 2.</p> <p>Set 0 for no trace, 1 for normal trace, or 2 for debug trace.</p>

Information options:

Parameter name	Default value	Description
enableStatusPage	false	Set to true or false. Set the value to true, to make the page accessible at <code>http://HostName:Port/ContextRoot/messagebroker/httpproxy/statuspage</code> where <i>HostName</i> is the name of your servlet container, <i>Port</i> is the port for your servlet container, and <i>ContextRoot</i> is the context root you have set for your HTTP proxy servlet. For more information about the context root, see "Importance of the context root when you configure the HTTP proxy servlet" on page 590.
enableInfoHeaders	false	Set to true or false. Set the value to true to make the HTTP proxy servlet add extra headers in the response message. These headers are: <ul style="list-style-type: none"> • X-WMB-Broker-Name • X-WMB-QM-Name • X-WMB-MQ-URL-CorrelId The headers also contain details of the configuration that is used for the response message.

ReplyToQ and QMgr options:

Parameter name	Default value	Description
useClusterMode	false	Set to true or false. Set to true if the HTTP proxy servlet is required to put reply-to queue and queue manager information in the MQMD of sent messages to enable the broker to respond to the correct queue manager in a cluster.
clusterModeQueueManagerName	SOME_OTHER_QUEUE_MANAGER	Set to the queue manager name for initial MQCONN and ReplyToQMgr.
clusterModeReplyToQ	OUR.REPLYTO.QUEUE	Set to the reply queue name on which to listen.

SSL connection options:

Parameter name	Default value	Description
useSecuredChannel	false	Set to true or false. Set to true if SSL is configured on the MQ channel. If set to true, the proxy servlet attempts to establish a secured connection to the MQ channel by using the <code>keyStore</code> , <code>keyStorePassword</code> , <code>trustStore</code> , <code>trustStorePassword</code> , and <code>cipherSuite</code> parameter values.
keyStore		Set to the full path of the keystore file. The fully qualified path to the keystore file, which is of type "JKS". For example, in Windows: <code>C:\\Program Files\\IBM\\MQSI\\keystore.jks</code> On Linux and UNIX: <code>/var/mqsi/keystore.jks</code>

Parameter name	Default value	Description
keyStorePassword	changeit	Set to the password of the keystore file.
trustStore		Set to the full path of the truststore file. The fully qualified path to the truststore file, which is of type "JKS". For example, on Windows: C:\Program Files\IBM\MQSI\truststore.jks On Linux and UNIX: /var/mqsi/truststore.jks This field is mandatory if useSecuredChannel is set to true.
trustStorePassword	changeit	Set to the password of the truststore file.
cipherSuite		Set to the encryption type that is configured in the MQ channel. For example: SSL_RSA_WITH_NULL_MD5 This field is mandatory if useSecuredChannel is set to true.

MQ connection options:

Parameter name	Default value	Description
useClientMode	false	Set to true or false. Set the value to true to use the WebSphere MQ client or set the value to false to use the bindings connection. Normally, useQueueManagerDataInsteadOfConfigFile would also be set to the broker queue manager if this parameter is set to true.
clientModeHostname	localhost	Set to the host name or the IP address for the queue manager.
clientModeChannelName	SYSTEM.DEF.SVRCONN	Set to the WebSphere MQ SVRCONN channel name
clientModePortNumber	1414	Set to the port number of the WebSphere MQ listener.
clientModeConnectRetryCount	1	Set to the number of times to retry the WebSphere MQ connect call. Use this parameter in cases where a network dispatcher or load balancer is used to distribute work to a set of queue managers and one of the queue managers fails. A new connect call might fail the first time, but succeed the second time. The retry count must be set to a high number to provide the greatest chance of success.
useQueueManagerDataInsteadOfConfigFile		Set to the queue manager name, * (remote proxy), or leave blank (local proxy). Set this value to the queue manager name to make the HTTP proxy servlet retrieve web address data from a queue, and avoid the need for a configuration file to be accessible by the HTTP proxy servlet.
sleepBeforeGet	0	Set to the time (in seconds) to wait before the HTTP proxy servlet issues an MQGET call for a response message from the broker.
disconnectBeforeSleep	true	Set to true or false. Set the value to true to release the WebSphere MQ handle while sleeping and minimize the number of simultaneous WebSphere MQ connections.

Parameter name	Default value	Description
<code>reconnectActiveLinksAge</code>	-1	<p>Set to a time (in seconds), 0, or -1</p> <p>Set the value to a number greater than zero to force WebSphere MQ connections to be disconnected and reconnected if they have been inactive, because of low traffic volumes, for more than the specified number of seconds.</p> <p>Set the value to -1 to prevent any reconnection.</p> <p>Set the value to 0 to force all connections to be used only one time.</p> <p>This parameter is useful if the connection to WebSphere MQ goes through a firewall that closes connections after a period of inactivity.</p> <p>Set the value to a value less than the firewall timeout to prevent clients from receiving WebSphere MQ 2009 (connection broken) error code responses.</p>
<code>testConnectionBeforeReuse</code>	false	<p>Set to true or false.</p> <p>Set the value to true, to make the HTTP proxy servlet attempt an MQINQ before it does the MQPUT of the HTTP data message. All problems with a cached WebSphere MQ client connection are detected at that point, and a new connection is established for the MQPUT of the actual data (and MQGET of the response).</p> <p>This parameter causes significant extra network traffic, and must be used only if there are problems with dropped connections, which are usually seen as WebSphere MQ 2009 errors, indicating that the connection is broken.</p>
<code>maximumConnectionAge</code>	-1	<p>Set to a time (in seconds), 0, or -1</p> <p>Set the value to a number greater than zero to disconnect and reconnect WebSphere MQ connections if they are older than the specified number of seconds.</p> <p>Set the value to -1 to prevents any reconnections.</p> <p>Set the value to 0 to use all connections only one time.</p> <p>This parameter is of most use if the frequent changes to the WebSphere MQ connection parameters are expected due to redeployment of the IBM Integration Bus flows, and you require the HTTP proxy servlet to reflect these changes within the specified number of seconds.</p>

Related concepts:

“HTTP proxy servlet overview” on page 584

By using the HTTP proxy servlet in a servlet container, you can support high availability, load distribution, access to the broker from multiple IP addresses and ports, and a larger number of concurrent HTTP sessions.

“HTTP proxy servlet operation” on page 584

Before you deploy the HTTP proxy servlet, ensure that you understand how the HTTP proxy servlet manages the communication between IBM Integration Bus applications and web services clients in a number of supported configurations.

Related tasks:

“Deploying the HTTP proxy servlet on a servlet container” on page 602

Load and install the HTTP proxy servlet file on a servlet container, such as Apache Tomcat or WebSphere Application Server Liberty Profile, or a web server, such as WebSphere Application Server.

“Testing the HTTP proxy servlet” on page 608

Test the HTTP proxy servlet with a broker that receives HTTP requests and passes them to a message flow.

Setting up the JNDI interface for the proxy servlet:

The JNDI interface for the proxy servlet requires a one time setup of the WebSphere Application Server full profile.

About this task

The proxy servlet initialization parameters must be configured for the broker environment that the proxy servlet is connecting to each time the proxy servlet is deployed to the servlet container. It is now possible to configure the `web.xml` parameters only once through the JNDI in WebSphere Application Server, regardless of how many future deployments there might be of the proxy servlet. Because the JNDI configuration parameters take precedence over the initialization parameters in the `web.xml` file, using this method means that you need to set up at the application server side only once for any future deployments of the proxy servlet.

These setup tasks must all be completed in the WebSphere Application Server administrative console.

Creating a resource environment provider:

About this task

Configure a resource environment provider, which encapsulates the referenceables that convert resource environment entry data into resource objects. These resource objects can then be accessed by applications.

Procedure

1. Select **Resources > Resource Environment > Resource Environment Providers**. The Resource environment providers wizard opens.
2. Click **New**. The Configuration panel opens so that you can configure a new resource environment provider.
3. Type a name for the resource environment provider in the **Name** field. For example, `MyResourceEnvironmentProvider`. It is recommended that you enter a meaningful description in the **Description** field, but it is not required. Click **OK** to continue and then save the changes.

Results

The new resource environment provider is listed in the wizard.

Creating a referenceable object:

About this task

Configure a new referenceable, which specifies the factory class that converts data in the Java Naming and Directory Interface (JNDI) name space into an object that represents your resource to WebSphere Application Server.

Procedure

1. Select **Resources > Resource Environment > Resource Environment Providers**. The Resource environment providers wizard opens.
2. From the Resource environment providers panel, select the provider that you created in the previous task: In this example, it is called *MyResourceEnvironmentProvider*.
3. Click **Referenceables**. The Referenceables panel opens.
4. Click **New**. The Configuration panel opens.
5. In the Configuration panel, type the following values:
 - a. In the **Factory class name** field, type:
`com.ibm.broker.httpproxy.MQResourceConnectionFactory`
 - b. In the **Class name** field, type:
`com.ibm.broker.httpproxy.MQResourceConnection`
6. Click **OK** and save the changes.

Creating resource environment entries:

About this task

Configure resource environment entries, which are objects that contain information about a resource, and represent it in the JNDI name space.

Procedure

1. Select **Resources > Resource Environment > Resource Environment Providers > MyResourceEnvironmentProvider**. The Resource environment providers wizard opens at the configuration panel for your provider.
2. Click **Resource environment entries**. The Resource environment entries panel opens.
3. Click **New**. The Configuration panel opens.
4. In the Configuration panel, type your values. In this example, the values are:
 - a. In the **Name** field, type for example: `MQResourceReference`
 - b. In the **JNDI name** field, type for example: `proxyservlet/reference/MQResourceReference`. This JNDI name is used during application deployment resource mapping.
 - c. In the **Referenceables** drop-down list, ensure that `com.ibm.broker.httpproxy.MQResourceConnectionFactory` is selected.
5. Click **OK** and save the changes.

Creating custom properties or define the parameters to be configured:

About this task

Specify custom properties that your enterprise information system (IES) requires for the resource providers and resource factories that you configure. For example, most database vendors require extra custom properties for data sources that access the database.

Procedure

1. Select **Resources > Resource Environment > Resource Environment Providers > MyResourceEnvironmentProvider > ResourceEnvironment Entries > MQResourceReference**. where *MyResourceEnvironmentProvider* and *MQResourceReference* are your own specified values. The Resource environment providers wizard opens at the configuration panel for your entry.
2. Click **Custom properties**. The Configuration panel opens.

3. Click **New**.
4. In the Configuration panel, you must type your values for all the resources that are administered through the administrative console. For example: The message broker name, the configuration file path, the client mode channel name, and the client mode port number. For each resource, you must provide values for **Name**, **Value**, **Description**, and **Type**. The following example is for the message broker name

Option	Description
Name	This field refers to the value in the "<param-name> </param-name>" tag in the web.xml. For example: brokerName
Value	This field refers to the value in the "<param-value> </param-value>" tag in the web.xml. For example: RRB
Description	Optional. It is recommended that you provide a meaningful description.
Type	Specify the type. For example: java.lang.String

5. Click **OK** to save the configuration.
6. Click **New** to create a new configuration for the remaining resources.

Results

You have a list of configured resources that is similar to the following table, but with the values that you specified:

Table 12. .

Name	Value	Description	Required
brokerName	RRB	Message Broker Name	false
configFilePath	C:\Documents and Settings\All Users\WINDOWS\Application Data\IBM\MQSI\components\RRB\config\soapplugin6.conf	Configuration File Path	false
clientModeChannelName	SYSTEM.DEF.SVR.CONN	Client Mode Channel Name	false
clientModePortNumber	2908	Client Mode Port Number	false

The WebSphere Application Server wizard does not provide an option to specify the **Required** attribute and so the default value is set to false. This attribute can be ignored.

What to do next

After you complete these tasks, you must deploy the proxyservlet.war file. During the deployment, you must provide the JNDI reference name for the **Resource Environment** reference that is defined in the web.xml file. For example: proxyservlet/reference/MQResourceReference.

After the deployment, proxyservlet gives precedence to the values configures in the Resource Environment Entries. If there is an environment change, the values must be modified in the custom properties of the Resource Environment Entries. After you modify a value, restart the Proxyservlet application in WebSphere

Application Server for the new values to take effect.

Related concepts:

“HTTP proxy servlet overview” on page 584

By using the HTTP proxy servlet in a servlet container, you can support high availability, load distribution, access to the broker from multiple IP addresses and ports, and a larger number of concurrent HTTP sessions.

“HTTP proxy servlet operation” on page 584

Before you deploy the HTTP proxy servlet, ensure that you understand how the HTTP proxy servlet manages the communication between IBM Integration Bus applications and web services clients in a number of supported configurations.

Related tasks:

“Deploying the HTTP proxy servlet on a servlet container”

Load and install the HTTP proxy servlet file on a servlet container, such as Apache Tomcat or WebSphere Application Server Liberty Profile, or a web server, such as WebSphere Application Server.

“Configuring the HTTP proxy servlet” on page 592

Configure the HTTP proxy servlet with the details of the broker environment to which the HTTP proxy servlet connects. The HTTP proxy servlet must be configured before you deploy the HTTP proxy servlet to a servlet container.

“Testing the HTTP proxy servlet” on page 608

Test the HTTP proxy servlet with a broker that receives HTTP requests and passes them to a message flow.

Deploying the HTTP proxy servlet on a servlet container

Load and install the HTTP proxy servlet file on a servlet container, such as Apache Tomcat or WebSphere Application Server Liberty Profile, or a web server, such as WebSphere Application Server.

Before you begin

Before you start:

Before you deploy the HTTP proxy servlet, you must complete the following tasks:

- Enable the broker listener for each integration server where message flows with SOAP nodes are deployed. See HTTP listeners and Switching from embedded listeners to a broker-wide listener.
- “Configuring the HTTP proxy servlet” on page 592

About this task

Supported operating systems

The documentation describes the installation on Windows, but the HTTP proxy servlet can be installed on any operating system that is supported by the servlet container.

Prerequisite components

The HTTP proxy servlet requires the following components:

- WebSphere Message Broker Version 7.0 or later.
- A supported version of WebSphere MQ. For the latest details of all supported levels of hardware and software, visit the IBM Integration Bus Requirements website.
- A servlet container, such as WebSphere Application Server Liberty Profile or Apache Tomcat, or a web server, such as WebSphere Application Server.

The following topics describe how to deploy the HTTP proxy servlet onto some specific servlet containers. However, the process should be similar for other servlet containers or web servers.

- “Deploying the HTTP proxy servlet on WebSphere Application Server Liberty Profile”
- “Deploying the HTTP proxy servlet on Apache Tomcat” on page 605
- “Deploying the HTTP proxy servlet on WebSphere Application Server” on page 607

Results

Results:

You have deployed the HTTP proxy servlet on a servlet container or web server

What to do next

Next steps:

Test the HTTP proxy servlet. For information about how to complete this task, see “Testing the HTTP proxy servlet” on page 608.

Related concepts:

“HTTP proxy servlet overview” on page 584

By using the HTTP proxy servlet in a servlet container, you can support high availability, load distribution, access to the broker from multiple IP addresses and ports, and a larger number of concurrent HTTP sessions.

Deploying the HTTP proxy servlet on WebSphere Application Server Liberty Profile:

Load and install the HTTP proxy servlet file on WebSphere Application Server Liberty Profile.

Before you begin

Before you start:

Before you deploy the HTTP proxy servlet, you must complete the following tasks:

- “Configuring the HTTP proxy servlet” on page 592
- Enabling the broker listener for SOAP nodes

About this task

Install and customize a WebSphere Application Server Liberty Profile (Liberty) server, and then deploy the HTTP proxy servlet by completing the following steps. The instructions assume that IBM Integration Bus is installed and running on the same machine as the Liberty server.

Procedure

To deploy the HTTP proxy servlet on the Liberty server, complete the following steps:

1. Follow the instructions on the WebSphere Application Server V8.5.5 Liberty Profile download page to download and install a Liberty server.

2. Create the following sub directory in your Liberty server installation:
Liberty_installation_path\wlp\usr\shared\resources\wmq.
3. Copy the following WebSphere MQ JAR files from *WebSphere_MQ_installation_path\Java\lib* to *Liberty_installation_path\wlp\usr\shared\resources\wmq*:
 - com.ibm.mq.commonservices.jar
 - com.ibm.mq.headers.jar
 - com.ibm.mq.jar
 - com.ibm.mq.jmqi.jar
 - com.ibm.mq.pcf.jar
 - connector.jar

The HTTP proxy servlet uses WebSphere MQ to communicate with IBM Integration Bus.

4. Copy the HTTP proxy servlet .war file that you exported from IBM Integration Bus (for example, *myproxyservlet.war*) to *Liberty_installation_path\wlp\usr\shared\apps*.
5. In the Eclipse workspace for the Liberty server, click the **Servers** tab and open the *server.xml* file by double-clicking the **Server Configuration** entry.
6. Add the following lines to *server.xml* to define the shared library that contains the WebSphere MQ .jar files:

```
<library id="wmq" name="wmq">
  <fileset dir="{shared.resource.dir}/wmq" includes="*.jar"/>
</library>
```

7. Add the following lines to *server.xml* to set the context root for the HTTP proxy servlet and link the HTTP proxy servlet with the WebSphere MQ shared library. For information about the context root, see “Importance of the context root when you configure the HTTP proxy servlet” on page 590.

```
<application id="proxyservlet" location="servlet_file_name.war"
  name="proxyservlet" type="war" context-root="context_root">
  <classloader privateLibraryRef="wmq"/>
</application>
```

where

servlet_file_name

Specifies the name of the HTTP proxy servlet file that you exported from IBM Integration Bus.

context_root

Specifies the context root you want to assign to the HTTP proxy servlet.

Note: If you want the HTTP proxy servlet to use an empty context root set `context-root="/"`.

8. Save and close *server.xml*, and start the Liberty server.
9. To test if the HTTP proxy servlet is active, enter the following URL in a web browser:

http://host_name:port/context_root

where

host_name

Specifies the host name of your Liberty server.

port

Specifies the port number of your Liberty server.

context_root

Specifies the context root that you assigned to the HTTP proxy servlet.

For example: `http://localhost:9080/`.

If the HTTP proxy servlet can reach the broker, then you should get an HTTP Status 404 response with the following error message:

URI / does not map to any message flow in broker *integration_node*

where *integration_node* is the name of your broker.

Results**Results:**

The HTTP proxy servlet is deployed on the WebSphere Application Server Liberty Profile server.

What to do next**Next steps:**

The HTTP proxy servlet is now ready to be tested with an integration node that receives HTTP (not HTTPS) requests and passes them to a message flow. For information about how to complete this task, see “Testing the HTTP proxy servlet” on page 608.

Related tasks:

“Deploying the HTTP proxy servlet on a servlet container” on page 602

Load and install the HTTP proxy servlet file on a servlet container, such as Apache Tomcat or WebSphere Application Server Liberty Profile, or a web server, such as WebSphere Application Server.

Deploying the HTTP proxy servlet on Apache Tomcat:

Load and install the HTTP proxy servlet file on Apache Tomcat.

Before you begin**Before you start:**

Before you deploy the HTTP proxy servlet, you must complete the following tasks:

- “Configuring the HTTP proxy servlet” on page 592
- Enabling the broker listener for SOAP nodes

About this task

Install and customize an Apache Tomcat 7 installation, and then deploy the HTTP proxy servlet by completing the following steps. The instructions assume that IBM Integration Bus is installed and running on the same machine as Apache Tomcat.

Procedure

1. Download Apache Tomcat from the Apache Tomcat 7 download page.
2. Run the Apache Tomcat installer and follow the instructions to complete the installation.

Note: When prompted for the path to a Java virtual machine (JVM) you can use the JVM that is deployed with IBM Integration Bus, for example C:\Program Files\IBM\MQSI\9.0.0.0\jre17.

3. Find the file `catalina.properties`.
This can be found at the following location:
`Tomcat_installation_path/conf/catalina.properties`
where `Tomcat_installation_path` is the name of your Apache Tomcat installation directory.
4. Edit `catalina.properties` and locate the following line:
`shared.loader=`
5. Edit the line so that it includes the following text:
`shared.loader=${catalina.home}/shared/lib,${catalina.home}/shared/lib/*.jar`
6. Create the directory `Tomcat_installation_path/shared/lib`.
7. Copy the following WebSphere MQ JAR files from `WebSphere_MQ_installation_path\Java\lib` to `Tomcat_installation_path\shared\lib`:
 - `com.ibm.mq.commonservices.jar`
 - `com.ibm.mq.headers.jar`
 - `com.ibm.mq.jar`
 - `com.ibm.mq.jmqi.jar`
 - `com.ibm.mq.pcf.jar`
 - `connector.jar`

The HTTP proxy servlet uses WebSphere MQ to communicate with IBM Integration Bus.

8. Start the Apache Tomcat Windows service.
9. Open a web browser and enter the web address `http://localhost:port` where `port` is the HTTP port number you specified in the Apache Tomcat installation. The default value for the port is 8080. The Apache Tomcat home page is displayed.
10. Click **Tomcat Manager** and enter the admin user ID and password.
11. Scroll down to the section **WAR file to deploy** and click **Browse**.
12. Navigate to the HTTP proxy servlet file that you exported from IBM Integration Bus. For example, `myproxyservlet.war`.

Note: By default in Apache TomCat, the context root that is used by the HTTP proxy servlet is the same as the `.war` file name. You can rename the `.war` file before you deploy the HTTP proxy servlet in order to assign a different context root to the HTTP proxy servlet. For information about the context root, see “Importance of the context root when you configure the HTTP proxy servlet” on page 590. If you want the HTTP proxy servlet to use an empty context root (`/`), then you must rename the `.war` file to `ROOT.war` and uninstall the current default web application (the Tomcat Welcome page).

13. Click **Deploy**.
14. To test if the HTTP proxy servlet is active, enter the following URL in a web browser:

`http://host_name:port/context_root`

where

host_name

Specifies the host name of your Apache Tomcat server.

port

Specifies the port number of your Apache Tomcat server.

context_root

Specifies the context root that you assigned to the HTTP proxy servlet.

For example: `http://localhost:8080/`.

If the HTTP proxy servlet can reach the broker, then you should get an HTTP Status 404 response with the following error message:

URI / does not map to any message flow in broker *broker*

where *broker* is the name of your broker.

Results**Results:**

The proxy servlet is deployed on your Apache Tomcat server.

What to do next**Next steps:**

The HTTP proxy servlet is now ready to be tested with an integration node that receives HTTP (not HTTPS) requests and passes them to a message flow. For information about how to complete this task, see “Testing the HTTP proxy servlet” on page 608.

Related tasks:

“Deploying the HTTP proxy servlet on a servlet container” on page 602

Load and install the HTTP proxy servlet file on a servlet container, such as Apache Tomcat or WebSphere Application Server Liberty Profile, or a web server, such as WebSphere Application Server.

Deploying the HTTP proxy servlet on WebSphere Application Server:

Load and install the HTTP proxy servlet file on WebSphere Application Server.

Before you begin**Before you start:**

Before you deploy the HTTP proxy servlet, you must complete the following tasks:

- “Configuring the HTTP proxy servlet” on page 592
- Enabling the broker listener for SOAP nodes
- Identify or install a WebSphere Application Server installation that can host the HTTP proxy servlet.

About this task**Procedure**

To deploy the HTTP proxy servlet on WebSphere Application Server, complete the following steps:

1. Deploy the HTTP proxy servlet file (that you exported from IBM Integration Bus) by following the instructions to deploy a web application in the WebSphere Application Server Version 8.5 product documentation.

Note: Make sure that you select the correct context root for your HTTP proxy servlet, see “Importance of the context root when you configure the HTTP proxy servlet” on page 590.

2. To test if the HTTP proxy servlet is active, enter the following URL in a web browser:

```
http://host_name:port/context_root
```

where

host_name

Specifies the host name of your WebSphere Application Server.

port

Specifies the port number of your WebSphere Application Server.

context_root

Specifies the context root that you assigned to the HTTP proxy servlet.

For example: `http://localhost:9080/`.

If the HTTP proxy servlet can reach the broker, then you should get an HTTP Status 404 response with the following error message:

```
URI / does not map to any message flow in broker integration_node
```

where *integration_node* is the name of your broker.

Results

Results:

The HTTP proxy servlet is deployed on WebSphere Application Server.

What to do next

Next steps:

The HTTP proxy servlet is now ready to be tested with an integration node that receives HTTP (not HTTPS) requests and passes them to a message flow. For information about how to complete this task, see “Testing the HTTP proxy servlet.”

Related tasks:

“Deploying the HTTP proxy servlet on a servlet container” on page 602
Load and install the HTTP proxy servlet file on a servlet container, such as Apache Tomcat or WebSphere Application Server Liberty Profile, or a web server, such as WebSphere Application Server.

Testing the HTTP proxy servlet

Test the HTTP proxy servlet with a broker that receives HTTP requests and passes them to a message flow.

Before you begin

To test the HTTP proxy servlet, use an existing web services client application or write your own SSL test client application by using Java.

Before you test the HTTP proxy servlet, you must have completed the following tasks:

- “Configuring the HTTP proxy servlet” on page 592
- Enabling the broker listener for SOAP nodes for the proxy servlet to access
- “Deploying the HTTP proxy servlet on a servlet container” on page 602

Procedure

To test the HTTP proxy servlet, complete the following steps:

1. Install a client application that can send HTTP or SSL (HTTPS) requests. For example,:
 - OpenSSL: An open source tool available from the Internet at: [OpenSSL downloads website](#) that can be used to send SSL (HTTPS) requests to the servlet container.
 - A broker message flow that has an HTTPRequest or SOAPRequest node that can generate and send HTTP requests to an HTTP listener.
 - A web browser by using web pages or Java Server Pages (JSP) that can send HTTP POST requests. Most web browsers support HTTP and HTTPS.
 - A client application that sends requests by using HTTP, HTTPS, or both HTTP and HTTPS.
2. Configure a message flow with HTTP and SOAP input and reply nodes. The HTTPInput and SOAPInput nodes must be configured with a URL that matches the context root of the HTTP proxy servlet; see “Importance of the context root when you configure the HTTP proxy servlet” on page 590. The message flow receives the messages from the HTTP proxy servlet. If an HTTP or SOAP reply node is configured, responses are sent back to the HTTP proxy servlet.

Results

Results:

You have tested the HTTP proxy servlet.

Related concepts:

“HTTP proxy servlet overview” on page 584

By using the HTTP proxy servlet in a servlet container, you can support high availability, load distribution, access to the broker from multiple IP addresses and ports, and a larger number of concurrent HTTP sessions.



HTTP listeners

You can choose between broker-wide listeners and integration server (embedded) listeners to manage HTTP messages in your HTTP or SOAP flows. Learn about the two types of listener, how ports are assigned to them, and how you can switch from one to the other for individual integration servers.

Related tasks:

“Deploying the HTTP proxy servlet on a servlet container” on page 602

Load and install the HTTP proxy servlet file on a servlet container, such as Apache Tomcat or WebSphere Application Server Liberty Profile, or a web server, such as WebSphere Application Server.

“Configuring the HTTP proxy servlet” on page 592

Configure the HTTP proxy servlet with the details of the broker environment to which the HTTP proxy servlet connects. The HTTP proxy servlet must be configured before you deploy the HTTP proxy servlet to a servlet container.



Resolving problems when developing message flows

Use the advice given here to help you to resolve common problems that can arise when developing message flows.

Using external web servers with IBM Integration Bus

IBM Integration Bus is frequently a component in Service Oriented Architecture (SOA) patterns that include external web servers and other HTTP solutions. Learn about the facilities that IBM Integration Bus provides to help you to configure your web server infrastructure, and which configuration option best fits your existing topology.

The following diagram shows a web server topology that uses a web server plug-in generated in WebSphere Application Server to handle HTTP and HTTPS connections with an application server. The web clients use HTTPS to communicate with the web server, and the WebSphere Application Server plug-in forwards traffic over HTTP to one of two application servers. This topology provides simplified administration: the clients connect to a single endpoint, and all client security is handled in a single place. The plug-in can be used to handle load-balancing and failover between the two application servers.

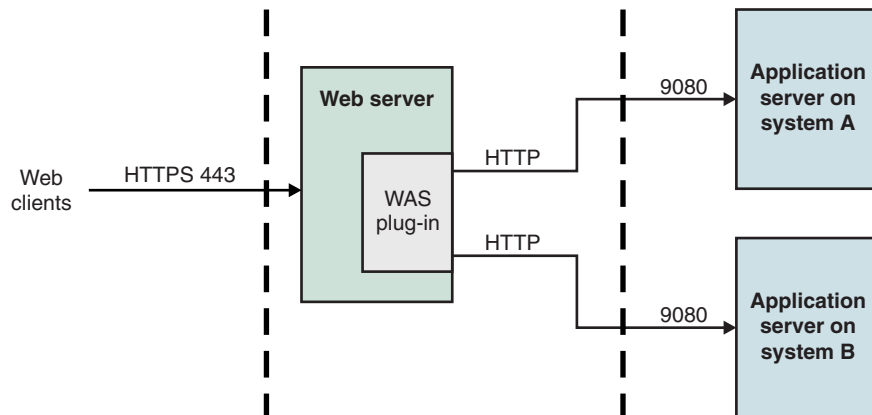


Figure 17. WebSphere Application Server plug-in with application servers

You can configure a WebSphere Application Server plug-in so that it can be used, without any loss of functionality, with IBM Integration Bus in the place of the application server. You can then take advantage of the load-balancing and failover capabilities of the plug-in to:

- Support many HTTP sessions
- Enable access to the broker from multiple IP addresses and ports
- Simplify the administration of your SOA topology

In the following diagram, IBM Integration Bus takes the place of the application servers. HTTP traffic is handled by broker-wide listeners on both systems.

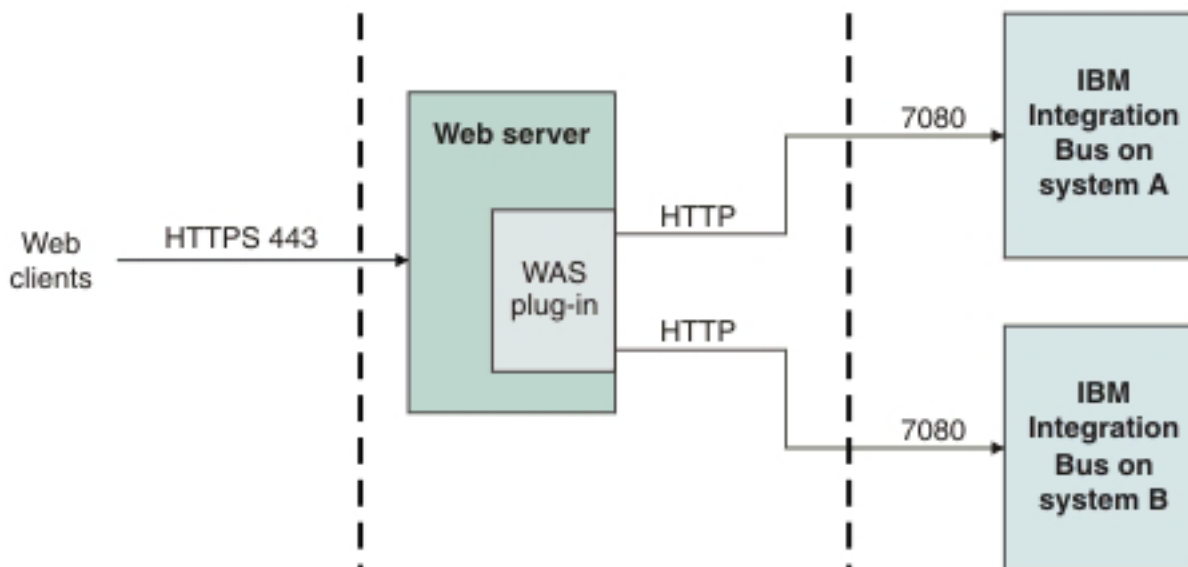


Figure 18. WebSphere Application Server plug-in with IBM Integration Bus

The WebSphere Application Server plug-in can be used in the preceding cases, and in more complex scenarios, to:

- Implement load-balancing
- Implement clustering for failover
- Expose application server and web server services and files through a single front end
- Manage security for all URLs in a single place and in a consistent manner
- Set up a network dispatcher to multiple web service endpoints

You can use IBM Integration Bus to achieve this type of configuration. Depending on your existing topology, use IBM Integration Bus to generate the most appropriate type of configuration data:

- If your existing topology incorporates WebSphere Application Server, generate a configuration file that can be read by a WebSphere Application Server web server plug-in
- If your existing topology does not contain WebSphere Application Server, but uses an Apache web server, such as IBM HTTP Server, generate a configuration file that can be merged into your Apache `mod_proxy` configuration
- If your topology is not described by the previous options, you can use the IBM Integration API to generate a per-broker list of port and URL data in CSV format, which you can use in your web server configuration

Refer to the following topics to help you to generate the appropriate configuration data for your web server topology:

- “Generating an HTTP server configuration by using the IBM Integration Explorer” on page 612
- “Generating an HTTP server configuration by using the IBM Integration API” on page 613
- The documentation on `BrokerProxy.getPortConfigAsCSV()` in the CMP reference documentation

Related concepts:



Processing HTTP messages

Hypertext Transfer Protocol (HTTP) is an Internet protocol that is used to transfer and display hypertext and XML documents on the Web.

Related tasks:



Developing applications that use the IBM Integration API

Develop Java applications that use the IBM Integration API (also known as the CMP) to communicate with, deploy to, and manage brokers and their associated resources.

Generating an HTTP server configuration by using the IBM Integration Explorer

You can generate a configuration file for either the WebSphere Application Server plug-in or the Apache mod_proxy module to use for connecting to a web server such as IBM HTTP Server.

Before you begin

Read the concept topic “Using external web servers with IBM Integration Bus” on page 610.

About this task

Complete the following steps to generate a configuration file:

Procedure

1. In the Navigator view, select the brokers for which you want to generate a configuration file. If you want to work with multiple brokers, hold down the Ctrl key while selecting the brokers.
2. Right-click the selected brokers and select **Export Port Configuration**.
3. Select the configuration that you want to export.
 - If you want to use the WebSphere Application Server plug-in, select **WebSphere Application Server Plugin Configuration**.
 - If you want to use the mod_proxy module, select **mod_proxy Configuration**.
 - In the Save As dialog, navigate to a target location for the configuration file, and enter the file name of your choice.
 - Click **Save** to save the file.

What to do next

Now that you have generated a configuration file, follow the relevant product instructions for configuring the web server:

- For the WebSphere Application Server plug-in, see the documentation on web server plug-ins at WebSphere Application Server documentation.
- For the Apache mod_proxy module, see the documentation on the mod_proxy module at Apache HTTP Server documentation.

Related concepts:

“Using external web servers with IBM Integration Bus” on page 610
IBM Integration Bus is frequently a component in Service Oriented Architecture (SOA) patterns that include external web servers and other HTTP solutions. Learn about the facilities that IBM Integration Bus provides to help you to configure your web server infrastructure, and which configuration option best fits your existing topology.

Generating an HTTP server configuration by using the IBM Integration API

Generate a configuration file for either the WebSphere Application Server plug-in or the Apache `mod_proxy` module to use for connecting to a web server such as IBM HTTP Server (IHS).

Before you begin

Read the concept topic, “Using external web servers with IBM Integration Bus” on page 610.

About this task

Use the IBM Integration API to generate the file contents for the `mod_proxy` module or the WebSphere Application Server plug-in configuration files. These files contain port and URL data for multiple brokers on multiple servers.

Procedure

Call the appropriate static method on the `BrokerProxy` class.

- If you are using the WebSphere Application Server plug-in, use the following method, passing in as parameters a `java.util.Map` mapping of `BrokerProxy` objects to host names and a value of data type `long` that represents a timeout interval in milliseconds:

```
public static String generateWASPlugin(Map <BrokerProxy,String> brokerHostMap, long timeoutInMilliseconds)
    throws ConfigManagerProxyLoggedException
```

- If you are using the Apache `mod_proxy` module, use the following method, passing in as parameters a `java.util.Map` mapping of `BrokerProxy` objects to host names and a value of data type `long` that represents a timeout interval in milliseconds:

```
public static String generateModProxyPlugin(Map <BrokerProxy,String> brokerHostMap, long timeoutInMilliseconds)
    throws ConfigManagerProxyLoggedException
```

- If you want to use an alternative to the plug-ins provided by WebSphere Application Server or Apache, you can use the following static `CMP` method on the `BrokerProxy` class to retrieve port and URL data for an individual broker in CSV format.

```
public static String getPortConfigAsCSV()
    throws ConfigManagerProxyLoggedException
```

For more information, see the `CMP` reference documentation at `CMP`.

What to do next

Now that you have generated a configuration file, complete the relevant product instructions for configuring the web server:

- For the WebSphere Application Server IHS plug-in, see the documentation on web server plug-ins at WebSphere Application Server documentation
- For the `mod_proxy` module, see the documentation on the `mod_proxy` module at Apache HTTP Server documentation

Related concepts:

“Using external web servers with IBM Integration Bus” on page 610
IBM Integration Bus is frequently a component in Service Oriented Architecture (SOA) patterns that include external web servers and other HTTP solutions. Learn about the facilities that IBM Integration Bus provides to help you to configure your web server infrastructure, and which configuration option best fits your existing topology.

Related tasks:

Developing applications that use the IBM Integration API
Develop Java applications that use the IBM Integration API (also known as the CMP) to communicate with, deploy to, and manage brokers and their associated resources.

Configuring a broker as a WebSphere MQ service

Use these topics to make changes when your broker is operating as a WebSphere MQ service.

About this task

- “Starting and stopping a broker as a WebSphere MQ service”
- “Modifying the WebSphere MQ service for a broker” on page 616
- “Deleting the WebSphere MQ service for a broker” on page 617
- “Reporting and displaying the status of a broker that runs as a WebSphere MQ service” on page 618

Related tasks:

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

“Creating a multi-instance broker with Windows on the Domain Controller” on page 558

How you create a multi-instance broker for a high availability configuration, when Windows is installed on the Domain Controller.

Starting and stopping a broker as a WebSphere MQ service

Configuring a broker to start and stop as a WebSphere MQ service.

Before you begin**Before you start:**

Ensure that you make the mqm user ID a member of the mqbrkr group. On Windows, you must reboot your workstation for the change to take effect.

About this task

To configure a broker to run as a WebSphere MQ service, use one of the following options:

Procedure

- Create a broker to start as a WebSphere MQ service by using the **mqscreatebroker** command with the **-d** parameter. For example,
mqscreatebroker MyBroker -q MyQMGR -d defined

where

MyBroker

Is the name of the broker that you want to start and stop as a WebSphere MQ service.

MyQMGR

Is the name of the queue manager associated with the broker.

- Modify an existing broker to start as a WebSphere MQ service by using the **mqschangebroker** command with the **-d** parameter. For example,
mqschangebroker MyBroker -d defined

where

MyBroker

Is the name of the broker that you want to start and stop as a WebSphere MQ service.

Results

When you have configured a broker to start and stop as a WebSphere MQ service:

- The broker starts and stops automatically when its associated queue manager starts and stops. For a multi-instance broker, this action can occur during failover of the active queue manager.
- A multi-instance broker cannot be started in standby mode when its WebSphere MQ service is defined as active.
- You can stop the broker manually by using the **mqsistop** command, but the broker does not restart until the queue manager is stopped and started again. Alternatively, you can start the broker manually with the **mqsistart** command, which invokes the WebSphere MQ service to start the broker.
- On UNIX systems the broker environment is inherited from WebSphere MQ. Set any required environment variables (such as ODBCINI) by using a script in the `work_path/common/profiles` directory. See “Command environment: Linux and UNIX systems” on page 124 for more information.

Related tasks:

“Creating the WebSphere MQ multi-instance queue manager” on page 549

How you create the WebSphere MQ multi-instance queue manager that you need for your multi-instance IBM Integration Bus.

“Creating a multi-instance broker with Windows on the Domain Controller” on page 558

How you create a multi-instance broker for a high availability configuration, when Windows is installed on the Domain Controller.

“Configuring a broker as a WebSphere MQ service” on page 614

Use these topics to make changes when your broker is operating as a WebSphere MQ service.

Related reference:



mqscreatebroker command

Use the **mqscreatebroker** command to create a broker and its associated resources.



mqsichangebroker command

Use the **mqsichangebroker** command to change one or more of the configuration parameters of the broker.

Modifying the WebSphere MQ service for a broker

How you modify the WebSphere MQ service for a broker.

About this task

To alter the status of an IBM Integration Bus so that it runs as a WebSphere MQ service use the **mqsichangebroker** command.

Carry out the following procedure:

Procedure

1. Stop the broker and its associated queue manager. If it is a multi-instance broker, you must stop all broker instances, and the multi-instance queue manager, if you want the changes to take affect immediately after the queue manager restarts.
2. Use the **mqsichangebroker** command with the **-d defined** option to activate the WebSphere MQ service as follows:

```
mqsichangebroker MyBroker -d defined
```

where

MyBroker

Is the name of the broker that you want to modify.

If the WebSphere MQ service did not previously exist, it is defined, and the service CONTROL attribute is set to QMGR.

You should no longer start the broker with the **mqsistart** command. The broker starts and stops automatically when the queue manager starts and stops.

You can stop the broker manually with the **mqsistop** command, but the broker will not restart until the queue manager is stopped and started again. For a multi-instance broker this can occur during failover of the active queue manager.

3. Use the **mqsichangebroker** command with the **-d undefined** option to remove the WebSphere MQ service as follows:

```
mqsichangebroker MyBroker -d undefined
```

where

MyBroker

Is the name of the broker that you want to modify.

When a WebSphere MQ service is being removed, and the queue manager is stopped, the **mqsichangebroker** command needs to start the queue manager to perform the deletion of the service. The command then stops the queue manager. The following example shows a possible output:

```
>mqsichangebroker BK3 -d undefined
WebSphere MQ queue manager 'QM3' starting.
5 log records accessed on queue manager 'QM3' during the log replay phase.
Log replay for queue manager 'QM3' complete.
Transaction manager state recovered for queue manager 'QM3'.
```

```
WebSphere MQ queue manager 'QM3' started.  
Waiting for queue manager 'QM3' to end.  
Waiting for queue manager 'QM3' to end.  
WebSphere MQ queue manager 'QM3' ended.  
BIP8071I: Successful command completion.
```

You can start the broker manually using the **mqsistart** command, which invokes the WebSphere MQ service to restart the broker.

Related tasks:

“Starting and stopping a broker as a WebSphere MQ service” on page 614
Configuring a broker to start and stop as a WebSphere MQ service.

“Creating a multi-instance broker with Windows on the Domain Controller” on page 558

How you create a multi-instance broker for a high availability configuration, when Windows is installed on the Domain Controller.

“Configuring a broker as a WebSphere MQ service” on page 614

Use these topics to make changes when your broker is operating as a WebSphere MQ service.

Related reference:



mqsichangebroker command

Use the **mqsichangebroker** command to change one or more of the configuration parameters of the broker.

Deleting the WebSphere MQ service for a broker

How you delete a WebSphere MQ service for a broker.

About this task

You can use either of the following methods to delete the WebSphere MQ service from a broker:

Procedure

1. Use the **mqsichangebroker** command with the **-d undefined** option, as described in step 3 on page 616 of “Modifying the WebSphere MQ service for a broker” on page 616 This reverts the broker to manual stop and start operation.
2. Use the **mqsdeletebroker** command. Using this command also removes a previously configured WebSphere MQ service when the broker is removed.

Related tasks:

“Modifying the WebSphere MQ service for a broker” on page 616

How you modify the WebSphere MQ service for a broker.

“Configuring a broker as a WebSphere MQ service” on page 614

Use these topics to make changes when your broker is operating as a WebSphere MQ service.

Related reference:



mqsicreatebroker command

Use the **mqsicreatebroker** command to create a broker and its associated resources.



mqsichangebroker command

Use the **mqsichangebroker** command to change one or more of the configuration parameters of the broker.

Reporting and displaying the status of a broker that runs as a WebSphere MQ service

How you report and display the status of a broker that runs as a WebSphere MQ service

Procedure

1. Use the **mqsireportbroker** command to show the property that displays the broker running as a WebSphere MQ service.

The `StartAsMQService` property is set to **defined** if the broker is set to start as a WebSphere MQ service, otherwise this property is set to **undefined**.

See the Windows, Linux, and UNIX systems version of the **mqsireportbroker** command for an example output from the command.

2. Use the **mqsilist** command to display the status of a broker that starts as a WebSphere MQ service. Message `bip1297` is output when a broker is stopped, but will start as a WebSphere MQ service.

Related tasks:

“Creating a multi-instance broker with Windows on the Domain Controller” on page 558

How you create a multi-instance broker for a high availability configuration, when Windows is installed on the Domain Controller.

“Configuring a broker as a WebSphere MQ service” on page 614

Use these topics to make changes when your broker is operating as a WebSphere MQ service.

Related reference:



mqsilist command

Use the **mqsilist** command to list installed brokers and their associated resources.



mqsireportbroker command

Use the **mqsireportbroker** command to display broker registry entries.

Appendix. Installation

Use the reference information in this section to understand installation requirements, installation options, and how they affect your computer.

- “System requirements”
- “Installation and uninstallation interfaces” on page 652
- “Installation and uninstallation authorization” on page 661
- “Multicultural support” on page 663
- System changes

Read the product `readme.html` file for late changes to the installation instructions; this file is provided on DVD and installed with the product. You can find the most up-to-date version on the web, as described in “Finding the latest information” on page 1.

Product requirements are also available on the web, and are occasionally updated. Check for the latest information about IBM Integration Bus Requirements.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

Related information:

 [IBM Integration Bus requirements](#)

System requirements

Use the reference information in this section to understand the hardware, software, and license requirements.

- “Hardware requirements” on page 620
- “Software requirements” on page 625
- “License requirements” on page 641

The product `readme` file `readme.html` might contain updates to the information in this chapter. The `readme` file includes information pertinent to all components and platforms, and is maintained in US English on the product `readmes` website:

www.ibm.com/support/docview.wss?uid=swg27006917

You must check this file to ensure that you have the latest information. Translated `readme` files are available on the documentation FTP site:

<ftp://public.dhe.ibm.com/software/integration/integrationbus/docs/>

A `readme` file is included with the product; it contains a minimum level of information and directs you to the online version. It is available in these locations:

- Before installation, on the product media.

The `readme` file is included on the DVDs for all operating systems in location `\readmes\locale\` (where *locale* identifies country, region, or language, for example `en_US`).

- After installation, in the installation directory:
 - For runtime components, in *install_dir\readmes\locale* (where *locale* identifies country, region, or language, for example en_US).
 - For the IBM Integration Toolkit, in *package_group_directory\wmbt\documentation\locale* (where *locale* identifies country, region, or language, for example en_US).

For further support information, including latest fixes and troubleshooting techniques, visit the IBM Integration Bus support web page:

www.ibm.com/software/integration/wbimessagebroker/support/

The supported hardware and software environments are updated occasionally. View the latest requirements information at the IBM Integration Bus Requirements website: Type IBM Integration Bus into the **Full or partial product name:** field and click the **Search product** icon. Select from the list of products and options returned.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

Related information:

 [IBM Integration Bus requirements](#)

Hardware requirements

View the processor and related hardware requirements on all platforms.

- “Supported processors”
- “Memory and disk space requirements” on page 621
- “Communications” on page 624

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Software requirements” on page 625

View the operating system, database, and other software requirements.

“License requirements” on page 641

Use the reference information in this section to understand license requirements.

Related information:

 [IBM Integration Bus requirements](#)

Supported processors

IBM Integration Bus is supported on multiple processors.

The hardware requirements for each supported operating system are given in the following table. All support statements assume that the stated systems can run the required level of a compatible operating system and have enough storage for the IBM Integration Bus components that you install, and all prerequisite products.

Table 13. Hardware requirements

Operating system	Requirements ¹
AIX	64-bit IBM System p [®] systems Any hardware from IBM or other vendors that can run trademarked AIX systems ²
HP-Itanium	Itanium systems
Linux on POWER	64-bit System i [®] and System p IBM POWER processor-based systems only
Linux on x86	IBM eserver System x [®] or equivalent Intel-based servers ³
Linux on x86-64	AMD64, EM64T, or compatible processor servers ³
Linux on System z	Any server capable of running one of the supported Linux on System z releases
Solaris on SPARC	Sun Microsystems SPARC processor servers
Solaris on x86-64	AMD64, EM64T, and compatible processor servers
Windows 32-bit	Windows x86 technology-compatible PC hardware ³
Windows 64-bit	AMD64, EM64T, or compatible processor servers ³
z/OS ⁴	Any server capable of running one of the supported z/OS releases

Notes:

1. Always check the IBM Integration Bus Requirements website and the readme.html file for the latest information about supported processors.
The readme.html file that is supplied on the product DVD (for all components) provides a minimum level of information, and directs you to the online file on the product readmes web page, which is updated regularly. Always use the online file to check that you have the latest level of information.
2. You can use AIX systems only if they have passed a set of verification tests for compliance with the AIX application binary and programming interfaces.
3. The IBM Integration Toolkit is supported on 32-bit and 64-bit systems. It requires a computer with an Intel Pentium III processor (or higher) that has a speed of at least 700 MHz. This specification is the minimum supported level; for improved performance use a 2-GHz processor.
A minimum display resolution of at least 1024 x 768 is required for some dialogs (for example, the Preferences dialog).
4. For further information, see the *Program Directory for IBM Integration Bus for z/OS* in the IBM Publication Center publication number, GI13-3314.

Related tasks:

“Installing” on page 31
Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1
Access the latest information for IBM Integration Bus.

Related reference:

“Operating system requirements” on page 627
IBM Integration Bus is supported on multiple operating systems.
“Additional software requirements” on page 633
IBM Integration Bus requires additional software products to run successfully.

Related information:

 IBM Integration Bus requirements

Memory and disk space requirements

Check the memory and disk space that is required for your installation.

Requirements for memory and disk space depend on the installation operating system, and on the IBM Integration Bus components and other products that you are installing.

Distributed systems

- 512 MB of RAM is required to support runtime operations (1 MB equals approximately 1 000 000 bytes).
- 512 MB of RAM is required to support IBM Integration Toolkit operations on Linux on x86, Linux on x86-64, or Windows. This specification is the minimum supported level; for improved performance, provide 1 GB (1 GB equals approximately 1 000 000 000 bytes).
- 512 MB of RAM is required to support the IBM Integration Explorer.
- Disk space requirements are dependent on the components that you install and the working space that is required by those components (for example, for WebSphere MQ queues and persistent messages).

Check that your computer has at least the space that is shown in the table, which provides guidance for both permanent product requirements and temporary space. Check that these requirements have not been updated in the latest product readme file `readme.html`.

If the installation directory and the temporary space are on the same partition or drive, add the two figures to check that you have enough space available. If you do not, increase the available storage or change the location of either the temporary space or the installation directory. The temporary files are deleted when installation is complete.

- On AIX, HP-UX, Linux, Solaris, and Windows systems, the default temporary space directory is pointed to by the `IATEMPDIR` system variable.

The installation wizard displays requirements for permanent space, but not for temporary space. If the figure that the installation wizard displays is greater than the figure shown in the following tables, check that your computer has sufficient space before you continue with the installation.

Table 14. Disk space requirements (Linux and Windows systems)

Component / product	Linux on POWER	Linux on x86 ¹	Linux on x86-64	Linux on System z	Windows 32-bit ¹	Windows 64-bit ¹
Integration Bus component	1.1 GB plus 2.2 GB temporary space	1 GB plus 2 GB temporary space	1 GB plus 2 GB temporary space	1.1 GB plus 2.2 GB temporary space	1.1 GB plus 2.2 GB temporary space	1.4 GB plus 2.8 GB temporary space
IBM Integration Toolkit	Unavailable	1.6 GB plus 1.6 GB temporary space	1.9 GB plus 220 MB temporary space	Unavailable	1.6 GB plus 1.6 GB temporary space	1.9 GB plus 220 MB temporary space
IBM Integration Explorer	Unavailable	250 MB plus 300 MB temporary space	250 MB plus 300 MB temporary space	Unavailable	200 MB plus 250 MB temporary space	200 MB plus 250 MB temporary space
IBM Integration ODBC Database Extender	90 MB plus 90 MB temporary space	80 MB plus 80 MB temporary space	75 MB plus 75 MB temporary space	80 MB plus 80 MB temporary space	Unavailable	Unavailable

Note:

1. The space that is required for the IBM Integration Toolkit includes space for the shared resources directory and the package group directory.

Table 15. Disk space requirements (UNIX)

Component / product	AIX	HP-Itanium	Solaris on SPARC	Solaris on x86-64
Integration Bus component	1.4 GB plus 2.6 GB temporary space	1.9 GB plus 3.8 GB temporary space	1.4 GB plus 2.6 GB temporary space	1.4 GB plus 2.6 GB temporary space
IBM Integration Toolkit	Unavailable	Unavailable	Unavailable	Unavailable
IBM Integration Explorer	Unavailable	Unavailable	Unavailable	Unavailable
IBM Integration ODBC Database Extender	80 MB plus 80 MB temporary space	200 MB plus 200 MB temporary space	130 MB plus 130 MB temporary space	130 MB plus 130 MB temporary space

- On computers on which you create a broker, up to 60 MB is required for the broker configuration data within your file system.
- If you use the embedded Integration Registry, up to 200 MB additional disk space is required in the Integration Bus component directory.
- If you create databases that are accessed by message flows, additional space is required on those computers.
- If you intend to create more than one broker on any one computer, you require additional memory and swap space. For example, you might create more than one broker at different versions to complete migration. Plan for 1 GB RAM and 1 GB of swap space for each broker, in addition to the space and memory requirements of other applications.

You must increase these figures if you deploy complex message flows to the brokers, or if you process large messages (of many megabytes), or complex messages that contain many different tags.

z/OS z/OS

Details are given in “Disk space requirements on z/OS.” You must also check for later updates to this information in the section about DASD storage requirements in the *Program Directory for IBM Integration Bus for z/OS* in the IBM Publication Center publication number, GI13-3314.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Uninstalling” on page 150

Remove the Integration Bus component, the IBM Integration Toolkit, or the IBM Integration Explorer from your computer.

Related reference:

“System requirements” on page 619

Use the reference information in this section to understand the hardware, software, and license requirements.

“Disk space requirements on z/OS”

The installation of IBM Integration Bus for z/OS uses approximately 400 MB of disk space; plan on using 500 MB to allow for the component directories, and for new service fixes to be applied.

Disk space requirements on z/OS

The installation of IBM Integration Bus for z/OS uses approximately 400 MB of disk space; plan on using 500 MB to allow for the component directories, and for new service fixes to be applied.

When you apply service, if you do not replace your existing installation (for example, you apply the new fix pack level alongside your existing installation), you must plan the same amount of disk space for the higher service level libraries.

If you are transferring the files by using *tar* to package them, you need approximately 200 MB of space for the .tar file.

You can check how much space is used and how much is free in a file system by using the OMVS command:

```
df -P /pathname
```

100 MB is 3 276 800 512 byte sectors.

The following table gives guidance on the space required for a minimum installation (base installation and verification test) of IBM Integration Bus for each component implemented on z/OS.

Component	Space required	Purpose
Component directory	20 MB If you use the embedded Integration Registry, up to 200 MB additional disk space is required.	Holds the runtime-deployed configuration information and output directories for the component. This information includes all deployment information, such as ESQL, JAR files, message sets, XSLT files, and so on. This information also includes IBM Integration Bus trace files and other user problem determination data, which might become large. Consideration must be given to the potential size of deployments to the IBM Integration Bus runtime environment and, therefore, the size of this directory (including sub directories).
Component PDSE	1 MB	Holds the customization and administration jobs, procedures, and data for the component. The data set must be allocated with a fixed record length of 80 (LRECL=80) and a format of FB 80. Reserve directory space for 50 members, or use a PDSE if possible.
Started task user ID home directory	8 GB	Collects diagnostic materials: for example, dumps. Dumps are usually more than 500 MB in size. 8 GB of space must be available in the file system, but many user IDs can have their home directory in the file system.

The Component directory and the Started task user ID home directory must be separate to ensure that, when dumps are taken in the Started task user ID home directory, they do not cause problems with the runtime broker that still has to write to the Component directory.

Related tasks:

“Customizing the z/OS environment” on page 290

If you are planning to use a z/OS environment, consider whether to create your brokers on z/OS. You must also complete a number of tasks to configure your environment.

“Using the file system on z/OS” on page 318

If you have more than one MVS image, consider how you will use the file system. You can share files in a file system with different members of a sysplex. The file system is mounted on one MVS image and requests to the file are routed to the owning system using XCF from systems which do not have it mounted.

Communications

Your system must have communications hardware that supports the protocols that brokers can use.

Choose one or more of the following protocols:

- NetBIOS
- SNA LU 6.2
- SPX
- TCP/IP

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

Related reference:

“Supported processors” on page 620

IBM Integration Bus is supported on multiple processors.

“Memory and disk space requirements” on page 621

Check the memory and disk space that is required for your installation.

Software requirements

View the operating system, database, and other software requirements.

This section provides information about requirements of IBM Integration Bus:

- “Support for 32-bit and 64-bit platforms”
- “Operating system requirements” on page 627
- “Supported databases” on page 628
- “Additional software requirements” on page 633
- “Optional software support” on page 638

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Hardware requirements” on page 620

View the processor and related hardware requirements on all platforms.

“License requirements” on page 641

Use the reference information in this section to understand license requirements.

Related information:

 [IBM Integration Bus requirements](#)

Support for 32-bit and 64-bit platforms

IBM Integration Bus operates in 32-bit mode or 64-bit mode, on supported operating systems.

The following table shows support for 32-bit and 64-bit mode. Changes since Version 6.1 are summarized beneath the table. If support on your platform has changed, you might need to make additional changes to your configuration. See “Migrating the Version 6.1 WebSphere Message Broker component” on page 225 and Chapter 5, “Configuring brokers for test and production environments,” on page 289 for further details.

Table 16. Support for 32-bit and 64-bit operation

Platform	IBM Integration Bus internal components and commands	32-bit integration servers	64-bit integration servers
AIX ¹	64-bit	No	Yes
HP-Itanium	64-bit	No	Yes
Linux on POWER	64-bit	No	Yes
Linux on x86	32-bit	Yes	No
Linux on x86-64 ²	32-bit	Yes	No
Linux on x86-64 ²	64-bit	No	Yes
Linux on System z	64-bit	No	Yes
Solaris on SPARC ¹	64-bit	No	Yes
Solaris on x86-64	64-bit	No	Yes
Windows 32-bit	32-bit	Yes	No
Windows 64-bit ²	32-bit	Yes	No
Windows 64-bit ²	64-bit	No	Yes
z/OS ¹	64-bit	No	Yes

Changes since Version 6.1:

1. On this platform, you can no longer create or run with 32-bit integration servers. Integration servers can be only 64-bit mode.
2. On the Windows 64-bit and Linux on x86-64 platforms, you can install either the 32-bit or the 64-bit version of IBM Integration Bus.

Integration servers

You can create an integration server by using one of the following:

- IBM Integration Toolkit
- IBM Integration Explorer
- `mqsicreateexecutiongroup` command
- CMP

If you do not specify the size of integration server that you require, the size that is created depends on how you create it, and the version of the target broker to which you deploy it. The following default values apply:

- If the target broker is Version 7.0 or later, the integration server is 64-bit except on those platforms that support 32-bit only (Linux on x86, and Windows).

Table 17. Integration server default sizes

Workbench	Command	IBM Integration API	32-bit only platforms	32-bit and 64-bit platforms	64-bit only platforms
Default	no options	<code>createExecutionGroup(name)</code>	32-bit	64-bit	64-bit

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.


“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.


“Migrating the Version 6.1 WebSphere Message Broker component” on page 225
Migrate your Version 6.1 components and resources to IBM Integration Bus Version 9.0.

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

 Creating an integration server using the IBM Integration Toolkit or IBM Integration Explorer

Use the IBM Integration Toolkit or IBM Integration Explorer to create integration servers on your broker.

 Creating an integration server using the `mqsicreateexecutiongroup` command
Use the `mqsicreateexecutiongroup` command to create integration servers on your broker.

Related reference:

“Hardware requirements” on page 620

View the processor and related hardware requirements on all platforms.

Operating system requirements

IBM Integration Bus is supported on multiple operating systems.

The following operating systems support IBM Integration Bus Version 9.0:

- AIX
- HP-Itanium
- Linux on POWER
- Linux on x86
- Linux on x86-64
- Linux on System z
- Solaris on SPARC
- Solaris on x86-64
- Windows 32-bit
- Windows 64-bit
- z/OS

Operating system software requirements are defined in the IBM Integration Bus Requirements web page.

In all operating environments except z/OS, defect support is available for virtualization environments where they relate to releases that are already supported by IBM Integration Bus. Unless stated elsewhere in the system requirements, IBM Integration Bus has not been tested in virtualization environments. IBM Integration Bus support is therefore unable to assist in issues related to configuration and setup, or issues that are directly related to the virtualization environment itself.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1
Access the latest information for IBM Integration Bus.

Related reference:

“Supported processors” on page 620
IBM Integration Bus is supported on multiple processors.

“Supported databases”

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

“Additional software requirements” on page 633

IBM Integration Bus requires additional software products to run successfully.

Related information:

 [IBM Integration Bus requirements](#)

Supported databases

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

The information that is shown here indicates the support for databases on each operating system, valid when the information center was published. However, database support might be enhanced after the release is made available. For the latest information about database support, visit the IBM Integration Bus Requirements website.

If you run message flows that access data that is held in databases, you must install and configure a supported database. Some data types that are supported by these databases are not supported by IBM Integration Bus; for details, see Data types of values from external databases.

In most environments, the broker and the database server must not be running on the same operating system. For details about local and remote database use, and the restrictions that apply, see “Database locations” on page 631.

IBM Integration Bus supports both transactional (XA) and non-transactional connections to databases. You can define an ODBC connection, or a JDBC type 4 connection, or both, to your database instances. XA support is referred to as a globally coordinated message flow.

On a single broker, you can use ODBC XA, or JDBC XA, but not both. This restriction applies to all supported platforms, and to all database servers for which XA is supported.

The following sections give details of this support, and describe restrictions where applicable:

- “ODBC support”
- “JDBC type 4 support” on page 630

ODBC support

The following table lists the supported database servers for each broker platform for non-XA support. XA connections are supported by most of these servers: See the notes following the table for details.

For the latest details of the versions of the database servers that are supported, visit the IBM Integration Bus Requirements website.

Unless otherwise stated, the ODBC client code for the database is at the same version and release as the server.

Operating system	DB2 ^{1,2}	Informix ⁵	Microsoft SQL Server	Oracle ^{1,3,4}	solidDB	Sybase ¹
AIX	Supported	Supported	Supported ⁶	Supported	Supported	Supported
HP-Itanium	Supported	Supported	Supported ⁶	Supported	Supported	Supported
Linux on POWER	Supported	Supported	Not supported	Supported	Not supported	Supported
Linux on x86	Supported	Supported	Supported ⁶	Supported	Supported	Supported
Linux on x86-64	Supported	Supported	Supported ⁶	Supported	Supported	Supported
Linux on System z	Supported	Supported	Supported ⁶	Supported	Not supported	Not supported
i5/OS™ and OS/400 ⁶⁷	Supported	Not supported	Not supported	Not supported	Not supported	Not supported
Solaris on SPARC	Supported	Supported	Supported ⁶	Supported	Supported	Supported
Solaris on x86-64	Supported	Supported	Supported ⁶	Supported	Supported	Supported
Windows 7	Supported	Supported ⁸	Supported	Supported ⁸	Supported ⁸	Supported ⁸
Windows 7 (64-bit)	Supported	Not supported	Supported	Supported ⁹	Supported ⁸	Not supported
Windows Server 2008	Supported	Supported ⁸	Supported	Supported ⁸	Supported ⁸	Supported ⁸
Windows Server 2008 R2	Supported	Not supported	Supported	Supported ⁹	Supported ⁸	Not supported
z/OS	Supported	Not supported	Not supported	Not supported	Not supported	Not supported

Notes:

1. XA support:

Supported releases of DB2, Oracle, and Sybase can participate as a resource manager in a distributed XA transaction, and can be coordinated by WebSphere MQ as the XA Transaction Manager, unless otherwise stated in the following notes. On z/OS, all transactions are coordinated by Recoverable Resource Services (RRS).

- Additional conditions for WebSphere MQ:

- If you use WebSphere MQ Version 7 for XA coordination on Windows, you must configure the queue manager to run under the broker service user ID by specifying the **-si** option on the **strmqm** command; for example, `strmqm -si QM_name`

- Additional conditions for databases:

- DB2 on i5/OS and OS/400, is not supported for XA connections.
- If you deploy message flows that access databases, you can define the message flows to be coordinated so that updates to those databases are synchronized with updates to other resources.
- ODBC drivers for Oracle and Sybase on all relevant systems, and for SQL Server on Linux and UNIX systems, are supplied with IBM Integration Bus. Alternative drivers are not supported on these systems. For other systems, and for other databases on all systems, obtain these files from your database vendor.

- Additional conditions for z/OS

- On z/OS, all transactions are globally coordinated by using RRS. As with other platforms you can choose for ODBC database operations to be committed or rolled back irrespective of the success or failure of the

message flow transaction as a whole. However, only one uncoordinated ODBC database connection per thread is supported.

2. Automatic Client Reroute for DB2 is supported on all platforms.
3. If you install the Oracle Database Server on 32-bit systems, you must also install the Oracle Runtime Client.
4. On all Linux, UNIX, and Windows systems, you can use Oracle RAC (Real Application Clusters) databases. For Oracle RAC XA and non-XA, failover support is limited to the Connect-Time Failover feature only.
5. Each broker system requires only the Client SDK; install the Dynamic Server on the system on which you create databases.
Large Objects (LOBs) are not supported.
6. On Linux and UNIX systems, you can remotely access an SQL Server database on Windows by using a supplied wire protocol driver.
7. You can configure message flows to access DB2 databases on i5/OS and OS/400 for user data. The message flows can run on all supported broker platforms. For further details of these restrictions, and for information about the PTFs that are required with DB2 on these operating systems, see “Database locations” on page 631.
8. Windows 32-bit edition of IBM Integration Bus only.
9. IBM Integration Bus for Windows 64-bit is supported in XA environments only when using WebSphere MQ 7.0.1.3 (or later) in non-Oracle RAC failover environments.

JDBC type 4 support

For non-XA transactions, you can create a JDBC type 4 connection from a broker to all the database servers that are listed in the ODBC support table. Connections are supported from all broker platforms, including z/OS.

For XA connections, the following restrictions apply:

- On distributed platforms, only DB2 and Oracle are supported.
- On z/OS, XA connections are not supported.

JDBC type 4 drivers are not supplied with IBM Integration Bus; obtain these files from your database vendor. For the latest details of the drivers that are supported, visit the IBM Integration Bus Requirements website.

Related tasks:

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.



Working with databases

Create and configure databases to use with your message flows.

Related reference:

“Database locations” on page 631

The broker can access databases that are set up on the local computer or on a remote server, subject to restrictions.



Data types of values from external databases

How database data types are implicitly cast to ESQL data types.

“Supported processors” on page 620

IBM Integration Bus is supported on multiple processors.

“Operating system requirements” on page 627

IBM Integration Bus is supported on multiple operating systems.

“Additional software requirements” on page 633

IBM Integration Bus requires additional software products to run successfully.

“Support for 32-bit and 64-bit platforms” on page 625

IBM Integration Bus operates in 32-bit mode or 64-bit mode, on supported operating systems.

Related information:

 IBM Integration Bus requirements

 DB2 V9.5 product documentation (distributed systems)

 DB2 V9.1 product documentation (distributed systems)

 DB2 Information Center (z/OS)

Database locations:

The broker can access databases that are set up on the local computer or on a remote server, subject to restrictions.

You can deploy message flows that access databases to one or more of your brokers.

- You can use a local or remote database for user data, subject to the following restrictions:

Databases on i5/OS and OS/400

- You can use only DB2 (UDB) on iSeries® (System i) under OS/400 or i5/OS.
- You must install OS/400 V5R2, or i5/OS V5R3 or V5R4.
- On Linux, UNIX, and Windows, you can connect using DB2 Connect. On Windows only, you can also use iSeries Access for Windows.
- You cannot define globally coordinated (XA) transactions.
- You can call stored procedures only if access to the remote database is provided by DB2 Connect.

Databases on z/OS

- You can use only DB2 (UDB) on z/OS.
- You must use DB2 Connect on Linux, UNIX, and Windows.

For further details, see the *Program Directory for IBM Integration Bus for z/OS* in the IBM Publication Center publication number, GI13-3314.

InfoSphere Federation Server databases

- You can configure message flows on all distributed systems to access user data in InfoSphere Federation Server for Linux, UNIX, and Windows, Version 9.1.
You can also configure message flows to access user data, by using JDBC only, in InfoSphere Classic Federation Server for z/OS, Version 9.1 on operating systems for which the InfoSphere Classic Federation Server client code is provided.
- You can configure message flows on z/OS brokers to access user data, by using JDBC only, in InfoSphere Classic Federation Server for z/OS, Version 8.2. This support provides connectivity with the following z/OS data sources:

- IMS
- VSAM
- ISAM
- Sequential files
- CA-IDMS
- CA-Datacom
- Software AG ADABAS
- DB2

Brokers can connect directly to InfoSphere Classic Federation Server for z/OS, Version 9.1 by JDBC only.

Brokers can connect indirectly to InfoSphere Classic Federation Server for z/OS, Version 9.1 through an intermediate DB2 system on z/OS. If you use this configuration, only an ODBC Call Attachment Facility (CAF) connection is supported between the broker and DB2; this connection therefore does not support two-phase commit.

You must configure all broker ODBC data resources as either CAF or Recoverable Resource Services Attachment Facility (RRSAF). If you use CAF, database operations are not coordinated by RRSAF.

- If you use a remote database, you must configure the ODBC connection to the database correctly. Refer to the documentation for the database product that you are using for more information.

See the documentation for the database product that you are using with IBM Integration Bus to determine the best options for your specific environment and requirements, and information about how to configure remote database access.

Related concepts:



Message flows overview

A message flow is a sequence of processing steps that run in the broker when an input message is received.



Database connections

Databases contain business data that is written and accessed by deployed message flows. You must create connections from the broker to the database by using ODBC or JDBC.

Related tasks:



Working with databases

Create and configure databases to use with your message flows.



Accessing databases from message flows

Create and configure message flows to access databases.



Configuring transactionality for message flows

A message flow runs in a single transaction, which is started when data is received by an input node, and can be committed or rolled back when all processing has completed.

Related reference:

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.



Built-in nodes

IBM Integration Bus supplies built-in nodes that you can use to define your message flows.

Additional software requirements

IBM Integration Bus requires additional software products to run successfully.

- “WebSphere MQ”
- “Microsoft Visual C++ ” on page 634
- Java runtime environment
- “IBM Installation Manager” on page 636
- “Browsers” on page 635

WebSphere MQ

All IBM Integration Bus components require WebSphere MQ at the minimum supported level that is shown in the table.

WebSphere MQ Version 7.5 (with tailored terms and conditions for use with IBM Integration Bus) is supplied on the IBM Integration Bus installation DVDs. If you have a previous version of WebSphere MQ, you can use a supplied DVD to upgrade your current installation.

A broker requires a WebSphere MQ queue manager. More than one broker cannot share a single queue manager.

Table 18. WebSphere MQ requirements

Operating system	Requirements
All distributed systems ¹	WebSphere MQ Version 7.5.0.1 ^{2, 3}
z/OS ⁴	WebSphere MQ Version 7.1 with WebSphere MQ Java Classes feature

Notes:

1. WebSphere MQ is not required on Linux on x86, Linux on x86-64, or Windows systems on which you install only the IBM Integration Toolkit.

You can configure SSL connections between the IBM Integration Toolkit or the IBM Integration Explorer and the broker. To configure an SSL connection, you must have an SSL certificate manager that is installed on the computer on which you have installed the IBM Integration Toolkit or the IBM Integration Explorer. If you choose to use WebSphere MQ to provide SSL management, install either the Client or the Server; both components include IBM Key Management tools.

2. The minimum set of components that you must install are the server and Java Messaging components.

You can install WebSphere MQ before or after you have installed IBM Integration Bus.

If you have already installed WebSphere MQ, check that your installation includes the Java Messaging component; add it from the WebSphere MQ media if it is not installed.

If you start the WebSphere MQ installation program directly on any platform, including Windows, select a custom installation and include the server and Java Messaging components.

If you want to use the WebSphere MQ Explorer, the graphical interface that is available on Linux on x86, Linux on x86-64, and Windows only, install the

WebSphere Eclipse Platform Version 3.3 and the WebSphere MQ Explorer components. The WebSphere MQ Explorer is a prerequisite if you want to use the IBM Integration Explorer.

Other WebSphere MQ components are optional.

3. On Windows, before WebSphere MQ Version 7.1 all configuration information, and most queue manager configuration information, was stored in the Windows registry. From WebSphere MQ Version 7.1 onwards all configuration information is stored in files. See the WebSphere MQ documentation for further information.
4. On z/OS, WebSphere MQ is a mandatory requirement and must be installed before you install IBM Integration Bus.

z/OS See the *Program Directory for IBM Integration Bus for z/OS* for further details about required levels of WebSphere MQ. The latest version of the program directory is available in the IBM Publication Center publication number, GI13-3314.

For details of WebSphere MQ products and supported versions, see the WebSphere MQ product information website.

Microsoft Visual C++

When you install the Windows 64-bit version of IBM Integration Bus on Windows 7 or Windows Server 2008 R2 only, Microsoft Visual C++ Runtime V10 for 32-bit and for 64-bit are required. The Visual C++ Runtime V10 installers that are supplied with the product media are in US English only. The installer for the Integration Bus component installs Visual C++ Runtime V10 automatically for you. If you want to install Visual C++ Runtime V10 separately, the installers are on the root of Runtime Disk 1.

If you want to install a multicultural version of this product, which displays a translated installation interface and product license agreement, but is otherwise identical to the US English version, you must download both the required versions that you want from the Microsoft website and install them before you install the IBM Integration Bus runtime.

JRE

A Java runtime environment (JRE) is required on all platforms:

- On distributed platforms, Java Runtime Environment (JRE) Version 7 is embedded in product components.
- On z/OS, you must acquire and install a JRE. See IBM Integration Bus Requirements for the latest information about JRE requirements for z/OS.

For more information, see the *Program Directory for IBM Integration Bus for z/OS* available in the IBM Publication Center publication number, GI13-3314. In particular, review the information in "Preventative Service Planning".

IBM Integration Bus supports all JMS providers that conform to the Java Message Service Specification, version 1.1, and requires the minimum JRE levels stated. You must consider both these factors when you select a JMS provider whose client is embedded in the broker.

Browsers

Windows On Windows, an embedded browser is started to view the information center, and is used in other parts of the IBM Integration Toolkit.

Linux On Linux, to view the information center and for other parts of the toolkit to function correctly, a compatible XULRunner must be installed.

XULRunner versions that are compatible with the IBM Integration Toolkit are documented at the following location: <http://www.eclipse.org/swt/faq.php#browserlinux>.

The compatible XULRunner versions are:

XULRunner 1.8.x - 1.9.x
3.6.x

XULRunner 2.x is not compatible. XULRunner 3.6.2.8 is recommended:
<http://ftp.mozilla.org/pub/mozilla.org/xulrunner/releases/3.6.28/runtimes/xulrunner-3.6.28.en-US.linux-i686.tar.bz2>

If you do not want to update the XULRunner on your system (for example you are running a later version), take the following steps to point the IBM Integration Toolkit to a supported XULRunner version:


1. Download XULRunner and expand the compressed file into a convenient directory (such as `/usr/lib/xulrunner_new`).
2. Add the following line to the end of the `eclipse.ini` file under the IBM Integration Toolkit installation directory:
`-Dorg.eclipse.swt.browser.XULRunnerPath=<location of unzipped download>`

For example

```
-Dorg.eclipse.swt.browser.XULRunnerPath=/usr/lib/xulrunner_new
```

3. Restart the IBM Integration Toolkit.

Related concepts:

 Security for the IBM Integration Toolkit and IBM Integration Explorer
What to consider when you set up security for the IBM Integration Toolkit and IBM Integration Explorer.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

“Customizing the version of Java on z/OS” on page 301

Check the version of Java in your enterprise, and change it, if necessary.

Related reference:

“IBM Installation Manager” on page 636

IBM Integration Toolkit is installed by the IBM Installation Manager.

“Hardware requirements” on page 620

View the processor and related hardware requirements on all platforms.

“Operating system requirements” on page 627

IBM Integration Bus is supported on multiple operating systems.

Related information:

 [IBM Integration Bus requirements](#)

IBM Installation Manager:

IBM Integration Toolkit is installed by the IBM Installation Manager.

All Rational products at Version 8.0 or later are installed by IBM Installation Manager, which also controls management, updates, licensing, and uninstallation. The IBM Integration Toolkit includes some Rational product components, and therefore includes and uses Installation Manager.

You can install IBM Integration Toolkit Version 9.0 into a package group that has the following Rational products installed by the IBM Installation Manager:

- Rational Application Developer (RAD) Version 8.0.2
- Rational Software Architect (RSA) Version 8.0.2
- IBM Integration Designer Version 7.5 or WebSphere Integration Developer Version 7
- Rational Team Concert Client (RTC) Version 3.0

For more information about package groups, see **Package groups** later in this section.

Installation Manager is included with the IBM Integration Toolkit and with the other products in this list. When you first install the IBM Integration Toolkit (or another listed product), Installation Manager installs itself into a directory that you specify, and then drives the installation of the IBM Integration Toolkit (or other listed product). If you install another product, Installation Manager detects that it is already installed, and drives only the installation of that product.

The IBM Integration Toolkit shares certain resources with these other products, if installed; for example, Eclipse features and plug-ins. All common resources that are used by the installed products must be installed into a single directory, which is known as the *shared resources directory*. You are asked to specify the location of this directory when you first install the IBM Integration Toolkit or another Rational product.

If you install another product, Installation Manager finds the shared resources directory and uses its content during the installation process; you cannot change the location of this directory.

The shared resources directory must be on a drive that is local to this computer; you cannot specify a mapped or remote drive. The drive that you specify for the shared resources directory must be of sufficient size to handle all your expected installations; you cannot change or expand this directory after installation. When you specify the directory during a first installation, specify a new directory to ensure that it does not contain any files that might cause conflicts.

Memory requirements for the IBM Integration Toolkit are listed in “Memory and disk space requirements” on page 621. If you plan to install additional Rational products, allow 2 GB for each additional product.

You must also allocate space in another directory in which you can manage the workspace resources that you create for the installed Rational products.

Package groups

When you install the IBM Integration Toolkit, you are also asked to specify a package group. Products that you install into a single package group share Eclipse features and plug-ins, and these resources are loaded and viewable in a single Eclipse instance. You can choose whether to install a product in a package group with other products, or to install IBM Integration Toolkit in a new package group.

Each package group is isolated from products in other package groups, although all package groups access a single shared resources directory. You are asked to specify the location for the package group directory; you must specify a new directory for each new package group. All product-specific files are installed into this directory.

You might choose to use separate package groups to install different combinations of the IBM Integration Toolkit and other Rational products, so that users can gain access to tailored Eclipse instances. When you install the first product, the first package group is created with the name **IBM Integration Toolkit**. This name is fixed; you cannot change it.

If you choose to install another product into a new package group, another group is created with the name **IBM Software Development Platform_1**. Each new package group name follows this same naming pattern with the numeric suffix incremented by one.

For example, you might have defined the following package groups:

- **IBM Integration Toolkit** into which you have installed the IBM Integration Toolkit and RAD.
- **IBM Integration Toolkit_1** into which you have installed the IBM Integration Toolkit and RSA.
- **IBM Integration Toolkit_2** into which you have installed WID and RSA.

When you start an Eclipse session in one of these package groups, you can access only those resources that are associated with the products installed in that group.

If you install later versions of any of the products in a different package group, the updates are available only in that group. The shared resources directory is also updated with later versions of shared files, which are maintained separately from the original versions and are used only for the upgraded products.

Each Rational product specifies which versions of plug-ins and features it requires, if appropriate. Installation Manager ensures the integrity of these requirements in each package group. If the product that you are currently installing breaks this integrity, Installation Manager prevents the installation into that package group.

Installation Manager also controls uninstallation of the IBM Integration Toolkit and the other products previously listed; you cannot uninstall Installation Manager until all listed products in all package groups have been removed.

Installation Manager is not required for any of the Integration Bus component prerequisite products, such as WebSphere MQ.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Additional software requirements” on page 633

IBM Integration Bus requires additional software products to run successfully.

Related information:

 [IBM Integration Bus requirements](#)

Optional software support

The products listed here are not required, but might be useful. Except where stated, these products are not supplied with IBM Integration Bus.

- “EIS client libraries”
- “WebSphere Adapters”
- “Tivoli License Manager”
- “Security providers”
- “JMS providers” on page 639
- “Compilers” on page 639
- Citrix XenApp
- “WebSphere Process Server and WebSphere Integration Developer” on page 639
- “WebSphere Service Registry and Repository” on page 640

EIS client libraries

If you plan to use WebSphere Adapters on any supported operating system, you must obtain the appropriate Enterprise Information System (EIS) client libraries from the relevant EIS vendor (for example, PeopleSoft, Siebel, or SAP). The client libraries are required to complete connections between Adapters nodes and the EIS; install them on each computer on which you run a broker that hosts message flows that include Adapters nodes.

WebSphere Adapters

For details of the versions of WebSphere Adapters supported by IBM Integration Bus, visit the IBM Integration Bus Requirements website. For more information about the hardware and software requirements of WebSphere Adapters, see Supported software for WebSphere Adapters.

Tivoli License Manager

To monitor the use of software products including IBM Integration Bus, install IBM Tivoli License Manager (ITLM). For the details of the version of ITLM that is required, visit the IBM Integration Bus Requirements website.

Security providers

The message flow security manager interacts with external security providers:

- Lightweight Directory Access Protocol (LDAP) provides authentication and authorization services. For details of the LDAP version that your LDAP server must support, visit the IBM Integration Bus Requirements website.
- WS-Trust V1.3 STS providers (including TFIM V6.2) provide authentication, mapping, and authorization services.
- Tivoli Federated Identity Manager (TFIM) V6.1 provides authentication, mapping, and authorization services.

JMS providers

For details of the version of the Java Message Service Specification that the JMSInput and JMSOutput nodes are compatible with, and work with, visit the IBM Integration Bus Requirements website.

Compilers

C/C++ user-defined extensions that were built with older levels of compilers can be used when migrating an existing broker or setting up a new broker.

However, C/C++ user-defined extensions on Linux might cause problems if linked against libstdc++.so.5 as IBM Integration Bus is now linked against libstdc++.so.6. Recompile your user-defined extensions using a supported compiler.

For details of the versions of the compilers that are supported, visit the IBM Integration Bus Requirements website.

WebSphere Process Server and WebSphere Integration Developer

Note: From Version 7.5 onwards, WebSphere Process Server has been renamed IBM Business Process Manager Advanced. Information in this topic that refers to WebSphere Process Server Version 7.0 is also applicable to IBM Business Process Manager Advanced Version 7.5. Similarly, from Version 7.5 onwards, WebSphere Integration Developer has been renamed IBM Integration Designer. Information in this topic that refers to WebSphere Integration Developer Version 7 is also applicable to IBM Integration Designer Version 7.5.

The SCA nodes allow interoperability with WebSphere Process Server, a business integration server that supports solutions that are based on service-oriented architecture (SOA). Service Component Architecture (SCA) is a specification that describes a model for building applications and systems using SOA and the SCA nodes support scenarios in which service components that run on WebSphere Process Server either are invoked from IBM Integration Bus or invoke an IBM Integration Bus message flow. For details of the versions of WebSphere Process Server that are supported by IBM Integration Bus, visit the IBM Integration Bus Requirements website.

WebSphere Integration Developer is the development environment for WebSphere Process Server. It is the tool for building and deploying SOA-based integration solutions on WebSphere Process Server. To enable IBM Integration Bus to interoperate with WebSphere Process Server, SCA import and export components need to be imported into IBM Integration Bus from WebSphere Integration Developer, and Broker SCA definitions need to be exported from IBM Integration Bus to WebSphere Integration Developer. For details of the versions of WebSphere Integration Developer that are supported by IBM Integration Bus, visit the IBM Integration Bus Requirements website.

WebSphere Service Registry and Repository

WebSphere Service Registry and Repository provides a central repository of documents that describe services, service interfaces, and associated policies. For details of the versions of WebSphere Service Registry and Repository that are supported by IBM Integration Bus, see the IBM Integration Bus Requirements website.

Related concepts:



WebSphere Service Registry and Repository

The WebSphere Service Registry and Repository (WSRR) is a central repository of entities. A wide range of entities can be stored and retrieved, including user-defined concepts and definitions related specifically to Web services, such as WSDL services, service interfaces, and associated policies.



Working with Service Component Architecture (SCA)

Start here to find out how you can use SCA to allow interoperability with IBM Business Process Manager Advanced Version 7.5 or WebSphere Process Server Version 7.0.

Related tasks:

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.



Testing and debugging message flows

Use the flow debugger to track messages through your message flows and check for errors, or use the Test Client to test your message flows in a safe environment before they are used on a production system.

“Installing IBM License Metric Tool” on page 115

IBM License Metric Tool (ILMT) enables you to monitor the use of IBM (and other) software products.



Compiling a C user-defined extension

Compile user-defined extensions in C for all supported operating systems.



Compiling a Java user-defined node

When you have created the code for your Java user-defined node, you must compile it for your operating system.

System requirements for Citrix XenApp:

This topic gives information about licensing issues, and the software and hardware that you need to use IBM Integration Bus in a Citrix XenApp environment.

Operating system

The required operating system is Windows Server. See IBM Integration Bus Requirements for supported versions. Client versions of Windows cannot be used because Citrix XenApp requires Terminal Server, which is not available on client versions of Windows.

The recommended version of Windows Server is Enterprise Edition. You can also use Windows Server Standard Edition, but you might need to reduce the maximum heap size for the Java virtual machine that is used by the IBM Integration Toolkit.

Software prerequisites

Citrix XenApp (previously known as Citrix Presentation Server 4.0 or 4.5) and its prerequisites must be installed on the Windows server.

Hardware prerequisites

The server must have enough CPU, RAM, disk space, and network bandwidth for each concurrent user. Performance depends on the tasks that are being performed, and on the hardware, software and networking setup. As a guide, the server should have a minimum of 1 GB of RAM per concurrent user and at least 4 GB of RAM in total. If the tasks require frequent disk access, it should help performance to have a separate physical disk for each user so that they do not slow each other down.

Licensing

Any number of users can use the IBM Integration Toolkit through Citrix, provided they all connect only to an IBM Integration Bus that has a valid license.

Related concepts:

“Users and Citrix” on page 118

Review the categories of user that might want to use with Citrix, and how to configure for concurrent multi-users.

Related tasks:

“Publishing in a Citrix XenApp environment” on page 116

Supply application location and user details to Citrix to publish an IBM Integration Bus command console or IBM Integration Toolkit.

License requirements

Use the reference information in this section to understand license requirements.

You can install IBM Integration Bus to support a full range of transformation and routing operations. If appropriate, you can install an edition that supports a restricted set of functions, if that subset fulfills your business requirements. You must ensure that your use and configuration of the product conforms to the license agreement that you have purchased:

- IBM Integration Bus for Developers. You can download this edition at no charge and you are free to use it for as long as you require, within the terms of the license. You can use all available function, and you are not limited in the number of resources that you can create and maintain.
IBM Integration Bus for Developers, which is also known as the Developer Edition, is restricted to processing one message per second. For more information, see “Operation modes” on page 5.
- IBM Integration Bus Standard Edition. Purchase this edition to satisfy either of the following sets of requirements:
 - You expect to use all or most of the features that are available, but you configure a limited environment because of low capacity requirements. You can use all available function, but are limited in the number of resources that you create and maintain.
 - You expect to use only a basic set of features, but have high capacity requirements. You can use a limited set of nodes, but there is no limit to the number of resources that you can create and maintain.
- IBM Integration Bus Express Edition. If you expect to use only a basic set of features and have only low capacity requirements, purchase this edition. You can use a limited set of nodes, and you are limited in the number of resources that you create and maintain.

- IBM Integration Bus Scale mode. If you expect to use only a basic set of features, but have high capacity requirements, purchase this license. You can use a limited set of nodes, but there is no limit to the number of resources that you can create and maintain.

WebSphere Enterprise Service Bus (WESB) customers can obtain a transfer license, which enables you to migrate from WESB to IBM Integration Bus, and to create and run brokers in Scale mode. As a current WESB customer, you can obtain the following transfer licenses:

- IBM Integration Bus WebSphere Enterprise Service Bus Transfer License
- IBM Integration Bus WebSphere Ent Svc Bus Transfer License Idle Standby

These licenses entitle you to obtain and use IBM Integration Bus in Scale mode in exchange for your WESB product licenses.

- IBM Integration Bus Remote Adapter Deployment. If you expect your typical use of IBM Integration Bus to be integration with Enterprise Information Systems (EIS), purchase this edition. This edition supports the subset of development resources that provide EIS interaction.
- IBM Integration Bus. If you want to set up a full broker environment that uses most or all of the features available, you require a full (unrestricted) license.

The following list shows which IBM Integration Bus operation modes can be used with which IBM Integration Bus license agreements:

IBM Integration Bus for Developers (Developer Edition)

developer mode

IBM Integration Bus Standard Edition

standard mode

scale mode

IBM Integration Bus Express Edition

express mode

IBM Integration Bus Remote Adapter Deployment

adapter mode

IBM Integration Bus (unrestricted license)

advanced mode

developer mode

standard mode

scale mode

express mode

adapter mode

For more information, see “Operation modes” on page 5.

IBM Integration Bus for z/OS is unavailable in Remote Adapter Deployment, Express Edition, and Developer Edition.

If you choose to change your license agreement from a full license to one of the specialized licenses, you might find that your current configuration is no longer supported. For further details about what features are available for each license, and how to configure your environment, see IBM Integration Bus technical overview.

You can upgrade to the full license from another edition, if appropriate, by purchasing another license.

You can view the full license for IBM Integration Bus by visiting the Software license agreements search website. Search for "IBM Integration Bus" and choose the license that applies to the version you are using.

You can view licenses after installation in your chosen language in directory *install_dir/license/*. Terms and conditions are also supplied for third-party products that are used by IBM Integration Bus. The file containing these details is stored in the same license subdirectory when you install one or more runtime components.

Contact your IBM representative if you want further details about license agreements, or if you want to purchase additional licenses or change the type of license that you have purchased.

Related concepts:

 [IBM Integration features](#)

Some features of IBM Integration Bus have specific requirements, such as a specific mode of operation or the availability of an additional product, such as a database. Additional terms and conditions also apply to the use of some IBM Integration Bus features, including the requirement for other product licenses.

“Operation modes” on page 5

The operation mode that you can use for your broker is determined by the license that you purchase.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Hardware requirements” on page 620

View the processor and related hardware requirements on all platforms.

“Software requirements” on page 625

View the operating system, database, and other software requirements.

Related information:

 [IBM Integration Bus requirements](#)

General industry standards supported by IBM Integration Bus

IBM Integration Bus supports general industry standards that are associated with message processing.

Supported standards

The following table details the general standards that are supported by IBM Integration Bus, with additional comments and supported versions where appropriate.

Standard	Comments or supported releases in Version 9.0
ACORD AL3	

Standard	Comments or supported releases in Version 9.0
ALE	For use with the SAP nodes
ASCII	
BAPI	For use with the SAP nodes
C	
COBOL	Including IBM and Micro Focus compilers
CORBA	2.3.1
CSV	
DFDL	1.0
EBCDIC	
EDI	
EDIFACT	
FIX	
FTP	
HIPAA	
HL7	V2 and V3, including HL7 V3 Clinical Document Architecture (CDA)
HTTP	1.0, 1.1
IDoc	For use with the SAP nodes
IMAP	
Java	1.7 (Version 7)
JDBC	
JMS	1.1
JNDI	
JSON	
Kerberos	
LDAP	Version 3
LTPA	
MIME	1.0
MTOM	
ODBC	
PHP	
PL/I	
POP3	
qRFC	For use with the SAP nodes
REST	
SAF	
SAML	1.1, 2.0
SCA	
SFTP	
SMTP	
SOAP	1.1, 1.2
SOAP/HTTP	
SOAP/JMS	1.0
SQL	
SSH	
SSL	
SwA	

Standard	Comments or supported releases in Version 9.0
SWIFT	
TCP/IP	
TDS	
TLOG	
TLS	
tRFC	For use with the SAP nodes
Unicode	
WS-Addressing	1.0 (Final 2005/08), Submission 2004/08
WSDL	1.1
WS-I	1.1
WS-ReliableMessaging	
WSSE	
WS-Security	1.0
WS-Trust	1.3
XML	
XPath	1.0
XSD	1.0
XSLT	1.0
X12	

Related concepts:



IBM Integration Bus technical overview

IBM Integration Bus enables information packaged as messages to flow between different business applications, ranging from large traditional systems through to unmanned devices such as sensors on pipelines.

Installation packages

View the installation packages that are available, and the contents of those packages.

This section contains the following topics:

- “Packaging options” on page 646
- “Package contents” on page 648
- “The Integration Bus component and IBM Integration Explorer package” on page 649
- “Toolkit package” on page 651

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related information:



IBM Integration Bus requirements

Packaging options

Both physical media and electronic images are available for the installation of IBM Integration Bus.

Physical media

You can order the physical media for IBM Integration Bus Standard Edition, Express Edition, Remote Adapter Deployment, and the full (unrestricted) licenses. Contents are described in “Package contents” on page 648.

When you install the product from the media, configure your brokers to operate in the mode that conforms to the license you have purchased. See the information center for further details.

Electronic images

If you are registered with IBM Passport Advantage, you can download electronic images from IBM Passport Advantage for IBM Integration Bus Standard Edition, Express Edition, Remote Adapter Deployment, and the full (unrestricted) licenses. You can also request one set of the physical media.

The electronic images consist of everything on the physical media that are described in “Package contents” on page 648. Although they are not formatted as DVD images, they are structured to allow individual components to be downloaded. For further information, and to register, access the IBM Passport Advantage website.

Electronic images are available on request for IBM Integration Bus for z/OS. Contact your IBM representative for further information and assistance.

When you install the product from these images, you must configure your brokers to operate in the mode that conforms to the license that you have purchased. See the information center for further details.

Developer Edition electronic images

You can download electronic images for IBM Integration Bus for Developers from the IBM Integration Bus web page.

Images are provided for the runtime Integration Bus component and the IBM Integration Toolkit. Use this edition to assess how the product can address your business requirements, and explore how you might use it with existing software in your enterprise.

IBM Integration Bus for Developers (which is also referred to as the Developer Edition) is a full-function version of the product, which you can use for evaluative purposes. You can download the Developer Edition at no charge, and you are free to use the product for as long as you require, within the terms of the license. Unlike the Trial Edition that was available for previous versions of the product, there is no expiry period for the Developer Edition license.

Developer Edition is restricted to processing one message per second. For more information, see “Operation modes” on page 5.

For the latest information about supported versions, always check the IBM Integration Bus `readme.html` file on the product readmes website.

All product prerequisites are included in the download package, including WebSphere MQ Version 7.5. You can download a single complete DVD

package by using either Download Director or HTTP, or you can download two separate packages for the runtime and toolkit components by using HTTP. If you download the two separate packages using HTTP, ensure that you extract the files to the same location in order to recreate the complete DVD image. For more information about the file structure of the DVD package, see “The Integration Bus component and IBM Integration Explorer package” on page 649.

The following restrictions apply:

- Physical images are not available for the Developer Edition.
- The Developer Edition is available for Windows x86-64 and Linux x86-64 operating systems only.

When you have downloaded IBM Integration Bus for Developers, you can install all components on Windows by using the Launchpad, or on Linux by using the `installAllSilent.sh` script.

If you are installing on Linux Ubuntu, see the information about installing WebSphere MQ Server on Linux Ubuntu in the WebSphere MQ Version 7.5 product documentation, and ensure that RPM has been installed before you use the `installAllSilent.sh` script to install WebSphere MQ. Also read the information about the additional flags that are required for RPM based distribution installations, and edit the `installAllSilent.sh` file (in the `sample-scripts` folder of the installation image) to specify the installation location. To install WebSphere MQ to the default location, replace the following line:

```
rpm -ivh *.rpm
```

with the following two lines:

```
rpm --prefix /opt/mqm -ivh --nodeps --force-debian MQSeriesRuntime-*.rpm  
rpm --prefix /opt/mqm -ivh --nodeps --force-debian *.rpm
```

If you want to install WebSphere MQ to a different location, modify these two lines as shown in the following example:

```
rpm --prefix /opt/customlocation -ivh --nodeps --force-debian MQSeriesRuntime-*.rpm  
rpm --prefix /opt/customlocation -ivh --nodeps --force-debian *.rpm
```

where *customlocation* is the name of the directory in which you want to install WebSphere MQ.

If you move from WebSphere Message Broker Version 8.0 Trial Edition to IBM Integration Bus for Developers, your trial mode brokers continue to work as before, but they are treated as developer mode brokers in the Developer Edition, with no expiry. If you subsequently revert from Developer Edition to the Version 8.0 Trial Edition, your trial mode brokers (which were treated as developer mode brokers in the Developer Edition) revert to trial mode, complete with the expiry period.

When you have installed Developer Edition, if you subsequently purchase a license and install the full version of IBM Integration Bus, any Developer mode brokers that are started with the full version are treated as Advanced mode brokers. In this case, you must also modify the operation mode of these brokers by using the `mqsimode` command to reflect the license that you have purchased. For more information, see “Changing the operation mode of your broker” on page 356.

Unless otherwise stated, you can use electronic images in the same way as the physical DVDs, and all installation and setup procedures that are described are identical for the Developer Edition and full packages.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Installation packages” on page 645

View the installation packages that are available, and the contents of those packages.

Related information:

 [IBM Integration Bus requirements](#)

Package contents

The content of the physical media packages for IBM Integration Bus is identical to the electronic images that you can download from the web. When you have obtained IBM Integration Bus through either of these media, you can select the appropriate mode of operation to match the entitlement that you have purchased: Standard, Adapter, Scale, or Express.

The Developer Edition is available only as an electronic download.

The contents of the package depend on the product that you have ordered:

IBM Integration Bus

The package includes product code for all supported distributed operating systems, plus other optional software and documentation:

- The Quick Start DVD. This DVD is always at the top of the package. The DVD includes PDF files for the Quick Start Guide (US English and translations).
- A set of plastic wallets that contain DVDs for installation of the broker, grouped by operating system. The DVDs contain all required and optional product code. The structure of the DVD content is described in “The Integration Bus component and IBM Integration Explorer package” on page 649.

The IBM Integration Toolkit package is contained on the DVDs for Linux on x86, Linux on x86-64, Windows 32-bit, and Windows 64-bit. For more information about the contents of the IBM Integration Toolkit package, see “Toolkit package” on page 651.

- The Quick Start Guide, printed in US English, French, and Japanese.

IBM Integration Bus for z/OS

The package includes product code for the z/OS operating system on tape, plus other optional software and documentation. In addition, you receive IBM Integration Bus for Linux on x86, Linux on x86-64, and Windows, because the IBM Integration Toolkit is available only on those operating systems.

The electronic images that you can download have equivalent content.

- A plastic wallet that contains four DVDs, one for Linux on x86, one for Linux on x86-64, one for Windows 32-bit, and one for Windows 64-bit. The DVDs contain all required and optional product code. The structure of the DVD content is described in “The Integration Bus component and IBM Integration Explorer package.”

The IBM Integration Toolkit package is contained on the DVDs. For more information about the contents of the IBM Integration Toolkit package, see “Toolkit package” on page 651.

- z/OS tapes

For information about tapes supplied with IBM Integration Bus for z/OS, see the *Program Directory for IBM Integration Bus for z/OS* in the IBM Publication Center publication number, GI13-3314.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Installation packages” on page 645

View the installation packages that are available, and the contents of those packages.

Related information:

 [IBM Integration Bus requirements](#)

The Integration Bus component and IBM Integration Explorer package

The Integration Bus component and IBM Integration Explorer packages contain DVDs and images that are supplied for IBM Integration Bus and associated products.

The contents that are listed in the following table are supplied for installation of the Integration Bus component.

The physical package for the DVD is marked with the symbol  .

Table 19. DVDs and images supplied for IBM Integration Bus and associated products

Operating System	DVD label	Description
AIX	IBM Integration Bus Version 9.0 AIX	Product code
	WebSphere MQ Version 7.5 AIX	Product code
	IBM Integration ODBC Database Extender	Product code
HP-Itanium	IBM Integration Bus Version 9.0 HP-Itanium	Product code
	WebSphere MQ Version 7.5 HP-Itanium platform	Product code
	IBM Integration ODBC Database Extender	Product code
Linux on POWER	IBM Integration Bus Version 9.0 Linux on POWER	Product code
	WebSphere MQ Version 7.5 Linux on POWER platform	Product code
	IBM Integration ODBC Database Extender	Product code

Table 19. DVDs and images supplied for IBM Integration Bus and associated products (continued)

Operating System	DVD label	Description
Linux on x86	IBM Integration Bus Version 9.0 Linux on x86	Product code
	IBM Integration Toolkit	Product code
	IBM Integration Explorer	Product code
	WebSphere MQ Version 7.5 Linux on x86	Product code
	IBM Integration ODBC Database Extender	Product code
Linux on x86-64	IBM Integration Bus Version 9.0 Linux on x86-64	Product code
	IBM Integration Toolkit	Product code
	IBM Integration Explorer	Product code
	WebSphere MQ Version 7.5 Linux on x86-64 platform	Product code
	IBM Integration ODBC Database Extender	Product code
Linux on System z	IBM Integration Bus Version 9.0 Linux on System z	Product code
	WebSphere MQ Version 7.5 Linux on System z platform	Product code
	IBM Integration ODBC Database Extender	Product code
Solaris on SPARC	IBM Integration Bus Version 9.0 Solaris	Product code
	WebSphere MQ Version 7.5 Solaris	Product code
	IBM Integration ODBC Database Extender	Product code
Solaris on x86-64	IBM Integration Bus Version 9.0 Solaris on x86-64	Product code
	WebSphere MQ Version 7.5 Solaris on x86-64	Product code
	IBM Integration ODBC Database Extender	Product code
Windows 32-bit	IBM Integration Bus Version 9.0 Windows 32-bit	Product code ¹
	IBM Integration Toolkit	Product code
	IBM Integration Explorer	Product code
	WebSphere MQ Version 7.5 Windows 32-bit	Product code
Windows 64-bit	IBM Integration Bus Version 9.0 Windows 64-bit	Product code ¹
	IBM Integration Toolkit	Product code
	IBM Integration Explorer	Product code
	WebSphere MQ Version 7.5 Windows 64-bit	Product code

Notes:

1. The Launchpad is included on the Windows DVD. If you use the Launchpad to install on Windows, as described in “Installing by using the Windows Launchpad” on page 60, the Launchpad navigates the DVD to find the products and components that you have chosen to install; therefore, you do not need to be familiar with the structure of the DVD contents.

The following resources are also included on all operating system DVDs:

- License files. These files are used by the installation wizard and are supplied in all supported languages.
- The readme file in English and supported languages. The readme.html file contains late updates about the product and its documentation. The latest version of the readme file is available in English only on the product readmes web page. The readme file that is included on the product media contains a link to the latest version on the product readmes web page.
- Sample scripts. Use the sample response files to run the silent interface to install and uninstall components.

For all operating systems, including Windows if you decide not to use the Launchpad, use the DVD structure that is shown in the following table to find what you need.

Table 20. DVD contents

Directory	Description
\ (root)	<ul style="list-style-type: none"> • IBM Integration Bus installation files • Launchpad (Windows only)
\ie02	IBM Integration ODBC Database Extender (Linux and UNIX only)
\license	License files
\Integration_Toolkit	IBM Integration Toolkit installation repository (Windows, Linux on x86, and Linux on x86-64 only)
\IBExplorer	IBM Integration Explorer installation repository (Windows, Linux on x86, and Linux on x86-64 only)
\readmes	Readme files
\sample-scripts	Sample response files
\WebSphere_MQ	WebSphere MQ installation images

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Installation packages” on page 645

View the installation packages that are available, and the contents of those packages.

Related information:

 [IBM Integration Bus requirements](#)

Toolkit package

Contents of the Toolkit package.

The Toolkit package is supplied on the product DVDs for installation of the IBM Integration Toolkit. The structure of the DVD content is described in “The Integration Bus component and IBM Integration Explorer package” on page 649.

The contents of the Toolkit package is shown in the following table.

Table 21. Packages supplied for IBM Integration Toolkit

Operating System	Description
Linux on x86 and Linux on x86-64	<ul style="list-style-type: none"> • Product code • Installation Manager¹ • Additional resources • Readme files • Sample scripts
Windows 32-bit and Windows 64-bit	<ul style="list-style-type: none"> • Product code • Installation Manager¹ • Additional resources • Readme files • Sample scripts • Launchpad

Notes:

1. Installation Manager is installed only if it does not exist on the target computer; this product is required to manage the IBM Integration Toolkit installation.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Installation packages” on page 645

View the installation packages that are available, and the contents of those packages.

Related information:

 [IBM Integration Bus requirements](#)

Installation and uninstallation interfaces

You can use different interfaces for installation and uninstallation of the Integration Bus component and the IBM Integration Toolkit.

When you install or uninstall the Integration Bus component, you can choose a graphical, console, or silent interface. These options are described in “How to install and uninstall the Integration Bus component” on page 653.

When you install or uninstall the IBM Integration Toolkit, you can choose a graphical or silent interface. These options are described in “How to install and uninstall the IBM Integration Toolkit” on page 657.

Examples of commands in the topics in this section use *installer* and *uninstaller* for the names of the installation or uninstallation wizard. Substitute the correct names for the platform on which you are using the wizard. Unless otherwise specified, you can use the examples for uninstalling components, or applying service; the same format is used for all three operations.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Applying service to the Integration Bus component” on page 130

Apply maintenance updates and program fixes to the Integration Bus component.

“Uninstalling the Integration Bus component” on page 151

You can uninstall the Integration Bus component on distributed systems in a number of ways.

“Applying service to the IBM Integration Toolkit” on page 141

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

“Uninstalling the IBM Integration Toolkit” on page 159

Choose the method that you want to use to uninstall the IBM Integration Toolkit and follow the instructions to remove the component.

Related reference:

“Installation and uninstallation authorization” on page 661

Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

How to install and uninstall the Integration Bus component

You can install and uninstall the Integration Bus component by using one of three interfaces.

- “Graphical interface”
- “Console interface” on page 654
- “Silent interface” on page 654

By default, the uninstaller launches in the graphical, console, or silent mode that is previously used to install the Integration Bus component. For example, if the installation uses the silent interface, the uninstaller defaults to using the silent interface. However, the uninstaller mode can be overridden on the command line by specifying the `-i swing`, `-i console`, or `-i silent` flags.

Each interface has different advantages, which are discussed in the appropriate sections. When you have chosen the interface that you want to use, complete the following steps:

- Check that your user ID has the correct authority to complete this task; see “Installation and uninstallation authorization” on page 661 for details.
- If you have multiple broker installations on your system, see “How to uninstall multiple installations of the Integration Bus component” on page 655.
- Follow the installation instructions in “Installing” on page 31 (for a new installation or to install service to existing components), or in “Uninstalling” on page 150 (for product components and service).

Note: In all cases, the installation log files are only available after you have exited the installer.

Graphical interface

The installation and uninstallation wizards open a graphical interface if you start them with no options (the default interface). The wizards guide you through the installation or uninstallation process with a series of pages that present options and defaults. You can accept the default values, or change them to suit your environment and requirements.

The graphical interface provides the highest level of information and guidance. Use this interface when you are unfamiliar with the product, or to monitor progress.

If you click **Cancel** before the Install Progress or Uninstall Progress panel appears, you can exit the setup. If you choose to exit, your system returns to the state that it was in before the wizard was started. However, if you cancel the installation wizard after installation or uninstallation has completed, and the final summary panel is displayed, your system is not restored to its previous state; the wizard stops immediately. If you want to remove any program that has been installed, invoke the uninstallation wizard.

When you use the wizards, you might have to wait a few seconds to move to the next panel after clicking **Next**. Progress is not always displayed on all panels. If you click **Next** twice, you might skip an entire panel. To ensure that the installer or uninstaller is progressing, monitor your processor usage, which increases greatly during both installation and uninstallation.

Console interface

The console interface is a character-based interface with which you interact in a command window. It presents the same options as the graphical interface.

Use the console interface if you want a command-line or text interface rather than a graphical interface. This interface is suitable for users who use only the keyboard to choose values and navigate through installation, and those users with screen reader software, such as JAWS.

Invoke the installer by using the following command. Use the same format for the uninstaller.

```
installer -i console
```

Follow the on-screen instructions to navigate through the wizard.

Silent interface

Use the silent interface for automated installation or uninstallation over many identical systems. If you start a silent installation or uninstallation, the wizard runs without any interaction. Using this interface, the process is completed with default options, or according to a predefined set of options. The silent interface does not provide any feedback to the caller, therefore you must view the log to check whether the action was successful.

If you use the silent interface to uninstall the Integration Bus component, the wizard always uninstalls components from the last known Version 9.0 installation location (that is, the most recent installation), regardless of the location of the uninstallation wizard that you invoke. To remove components from an earlier Version 9.0 installation, use the console or graphical interface.

You can perform a silent installation in the following ways:

- With default options:

The installation wizard performs the following actions:

- Checks that prerequisite software is installed
- Installs to the default directory
- Installs all selectable features

Because the installation wizard checks for prerequisite software by using the silent interface, the program fails if the prerequisite software is not already installed. You can override this check if you use a response file (see “Using response files with the Integration Bus component” on page 656), or include the appropriate parameter with a non-default value on the command invocation.

The uninstallation wizard performs the following actions:

- Removes all selectable features

To run the wizard with default options, specify the `-i silent` option on the command:

```
installer -i silent
```

- With one or more non-default options:

If you want the wizard to use non-default values for one or more options, specify non-default options either on the command invocation, or in a response file, as described in “Using response files with the Integration Bus component” on page 656.

A sample response file is provided in the `sample-scripts` directory of the root DVD directory. This file includes detailed information about the options that you can change, and the values that you must enter to change them. Tailor this file to match your requirements, or generate a new response file.

To view example response files, see “Installing the Integration Bus component in silent mode” on page 73. When the response file is complete, start the installation by running the following command from within the directory in which you downloaded the Integration Bus component. If you start the installation from a directory other than the one in which the installation wizard exists, include the absolute or relative path with the command name.

```
installer -i silent -f response_file_name
```

Where *installer* is the installation wizard name. Where *response_file_name* is the name of your response file.

You can also specify the silent installation options by using the command line. For more information about completing this task, see “Installing the Integration Bus component in silent mode” on page 73.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Applying service to the Integration Bus component” on page 130

Apply maintenance updates and program fixes to the Integration Bus component.

“Uninstalling the Integration Bus component” on page 151

You can uninstall the Integration Bus component on distributed systems in a number of ways.

Related reference:

“Using response files with the Integration Bus component” on page 656

You can use a response file to define the behavior of an installation wizard that is running the silent interface.

“Installation and uninstallation interfaces” on page 652

You can use different interfaces for installation and uninstallation of the Integration Bus component and the IBM Integration Toolkit.

“Installation and uninstallation authorization” on page 661

Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

How to uninstall multiple installations of the Integration Bus component

Uninstalling the Integration Bus component when you have multiple installations.

If you install the same version and release of the Integration Bus component on a single computer (for example, Version 9.0) more than once, the installer support provided by the operating system cannot manage these installations in the normal way.

If you later want to uninstall one of the multiple installations, use the uninstallation program *uninstaller* in the `Uninstaller` directory of the specific installation that you want to remove, not the facilities provided by the operating system.

You can view the `install.properties` file to see current installations and their locations, and check the operating system representation:

AIX The first installation is recorded as `mqsivr`, for example `mqs i90`. Subsequent installations at the same `vr` level are displayed under the first one when you list installed products with `lsipp`. If you use `smitty` and `geninstall` to manage those subsequent installations, results are unpredictable.

Linux, HP-UX, and Solaris

The first installation is recorded as `mqs i/vr`, for example `mqs i/90`. Subsequent installations at the same `vr` level are recorded as `mqs i/vr-2`, and so on.

Windows

The most recent installation that you completed for any given version and release is displayed in **Add/Remove Programs**. No other installations are shown here. Similarly, the **Command Console** option in the **Start** menu is that associated with the most recent installation for any given version and release.

If you uninstall the product at a specific version and release listed by **Add/Remove Programs**, earlier installations that you completed on the computer are not reinstated in that view.

To uninstall other instances, navigate to the directory that contains the uninstallation program. For details of uninstallation tasks, see the information center.

Using response files with the Integration Bus component

You can use a response file to define the behavior of an installation wizard that is running the silent interface.

A response file is a text file that contains options that define the choices that the wizard makes. You can use response files to install or uninstall the Integration Bus component, or to apply service updates by using non-default values.

- “Editing the sample response files”
- “Recording a response file”
- “Calling response files in commands” on page 657

Editing the sample response files:

A sample response file is supplied. On Linux and UNIX systems, the file is `/sample-scripts/response.properties`. On Windows, the file is `\sample-scripts\response.properties`. The sample response file includes detailed information about the options that you can change, and the values that you can enter to change the options. You can tailor the file to match your requirements. To view example response files, see “Installing the Integration Bus component in silent mode” on page 73.

Recording a response file:

Use the following command to record a response file:

```
installer -r response_file_directory
```

On Windows, surround the path and name with quotation marks if it contains spaces (“response file”). Create this file in a directory different from the one in which the product is installed.

The installation wizard starts its graphical interface and records your responses as it progresses. When installation is complete, the response file contains all the choices that you have made during installation.

If you want to record a response file during a console installation, use the following command:

```
installer -r response_file_directory -i console
```

Calling response files in commands:

Use the following command to run the silent interface with a response file:

```
installer -i silent -f response_file_name
```

On Windows, surround the path and name with quotation marks if it contains spaces.

The wizard runs, taking its input from the response file.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Applying service to the Integration Bus component” on page 130

Apply maintenance updates and program fixes to the Integration Bus component.

“Installing the Integration Bus component in silent mode” on page 73

Install the Integration Bus component by using the installation wizard in silent mode.

“Uninstalling the Integration Bus component” on page 151

You can uninstall the Integration Bus component on distributed systems in a number of ways.

How to install and uninstall the IBM Integration Toolkit

Install and uninstall the IBM Integration Toolkit by using one of two interfaces.

- “Graphical interface”
- “Silent interface” on page 658

Each interface has different advantages, which are explained in the appropriate sections. When you have chosen the interface that you want to use:

- Check that your user ID has the correct authority to complete this task; see “Installation and uninstallation authorization” on page 661 for details.
- Before installing IBM Integration Toolkit on Linux, check that the default permissions are set correctly by running the following command under the user ID that will perform the installation:

```
umask
```

A value of `0022` should be returned, indicating the permissions are correctly set. If any other value is returned, set the correct permissions by running the following command:

```
umask 0022
```

- Follow the installation instructions in “Installing” on page 31 (for a new installation or to install service to existing components), or in “Uninstalling” on page 150 (for product components and service).

Graphical interface

The installation and uninstallation wizards open a graphical interface if you start them with no options (the default interface). The wizards guide you through the

installation or uninstallation process with a series of pages that present options and defaults. You can accept the default values, or change them to suit your environment and requirements.

The graphical interface provides the highest level of information and guidance. Use this interface when you are unfamiliar with the product, or to monitor progress.

If you click **Cancel** before the Install Progress or Uninstall Progress panel appears, you can exit the setup. If you choose to exit, your system returns to its state that it was in before the wizard was started. However, if you cancel the installation wizard after installation or uninstallation has completed, and the final summary panel is displayed, your system is not restored to its previous state: the wizard stops immediately. If you want to remove any program that has been installed, start the uninstallation wizard.

When you use the wizards, you must wait a few seconds to move to the next panel after clicking **Next**. Progress is not always displayed on all panels. If you click **Next** twice, you might skip an entire panel. To ensure that the installer or uninstaller is progressing, you can monitor your processor usage, which increases greatly during both installation and uninstallation.

Silent interface

Use the silent interface for automated installations over many identical systems. If you start a silent installation or uninstallation, the wizard runs without any interaction; the process is completed with default options, or according to a predefined set of options. The silent interface does not provide any feedback to the caller; therefore, you must check the log to determine whether the action was successful.

You can run a silent installation with default settings, or with one or more non-default values:

- With default settings, the installation wizard performs the following actions:
 - Installs to the default directories
 - Installs all supported locales

To run a default silent installation, change the path to the `Integration_Toolkit` directory of the local or remote DVD, or the network drive, and run the following command:

Linux **Linux on x86**

```
installToolkit-silent.sh
```

Windows **Windows**

```
installToolkit-silent.bat
```

- With one or more non-default settings, the wizard applies the options that you specify in a response file to determine what actions to take.

If you want the wizard to use non-default values for one or more options, specify a recorded response file, as described in “Using response files with the IBM Integration Toolkit” on page 659.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Applying service to the IBM Integration Toolkit” on page 141
Apply maintenance updates and program fixes to the IBM Integration Toolkit.

“Uninstalling the IBM Integration Toolkit” on page 159
Choose the method that you want to use to uninstall the IBM Integration Toolkit and follow the instructions to remove the component.

Related reference:

“Using response files with the IBM Integration Toolkit”
Specify a response file to define the behavior of the installation or uninstallation wizard.

“Installation and uninstallation interfaces” on page 652
You can use different interfaces for installation and uninstallation of the Integration Bus component and the IBM Integration Toolkit.

“Installation and uninstallation authorization” on page 661
Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

Using response files with the IBM Integration Toolkit

Specify a response file to define the behavior of the installation or uninstallation wizard.

You can use response files to install or uninstall the IBM Integration Toolkit, or to apply service updates.

- “Editing the sample response files”
- “Recording a response file”
- “Calling response files in commands” on page 660

Editing the sample response files:

A sample response file `mbtoolkit-silent.xml` is supplied on Linux on x86 and Windows. The files assume that IBM Installation Manager has not yet been installed, and sets options to install both Installation Manager and the IBM Integration Toolkit in the default locations.

Although you can tailor these files to match your requirements, for example by changing the installation locations, the record option on a graphical installation or uninstallation is preferable. If you use the record option, you do not have to modify the file content, which is complex because the files handle multiple installations, directories, and options.

Recording a response file:

Use the following command to record a response file:

- On Linux on x86:

```
./install -record response.xml
```

- On Windows:

```
install.exe -record response.xml
```

where *response.xml* is the full path and name of your chosen response file. On Windows, surround the path and name with quotation marks if it contains spaces. Create this file in a directory different from the one in which the product is installed.

The installation wizard opens its graphical interface, and requires your input as it progresses. Your responses are recorded during installation. When installation is complete, the response file contains all the choices that you have made during installation.

Calling response files in commands:

Use the following command to run the installation wizard with the silent interface and a recorded response file:

- On Linux on x86:

```
./install -nosplash --launcher.suppressErrors -silent -input response.xml
```

- On Windows:

```
install.exe -nosplash --launcher.suppressErrors -silent -input response.xml
```

where *response.xml* is the full path and name of the response file you recorded. On Windows, surround the path and name with quotation marks if it contains spaces.

The wizard runs, taking its input from the response file.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Applying service to the IBM Integration Toolkit” on page 141

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

“Uninstalling the IBM Integration Toolkit” on page 159

Choose the method that you want to use to uninstall the IBM Integration Toolkit and follow the instructions to remove the component.

Installation wizard names

View the installation wizards that are used to install the Integration Bus component, the IBM Integration Toolkit, and the IBM Integration Explorer.

The installation wizard has a different name on each operating system. To help you find these programs quickly, the names are shown in the following tables.

Installation wizard names for the Integration Bus component

The following table shows the installation wizard names that are used to start the installers for the Integration Bus component.

Table 22. Installation wizard names for the Integration Bus component

Operating system	Installation wizard name
AIX	setupaix
HP-Itanium	setuphpia64
Linux on POWER	setuplinuxppc
Linux on x86	setuplinuxia32
Linux on x86-64	setuplinuxx64
Linux on System z	setuplinux390
Solaris on SPARC	setupsolaris
Solaris on x86-64	setupsolarisx64
Windows 32-bit	setup32.exe
Windows 64-bit	setup64.exe

Installation wizard names for IBM Integration Toolkit

The following table shows the installation wizard names that are used to start Installation Manager that controls the installation of the IBM Integration Toolkit.

Table 23. Installation wizard names for IBM Integration Toolkit

Operating system	Installation wizard name
Linux on x86	install
Linux on x86-64	install
Windows 32-bit	install.exe ¹
Windows 64-bit	install.exe ¹

Note:

1. You can also use `installc.exe` to start Installation Manager. This program operates synchronously and does not return control to the command line until the installation has completed.

Installation wizard names for IBM Integration Explorer

The following table shows the installation wizard names that are used to start the installers for the IBM Integration Explorer.

Table 24. Installation wizard names for IBM Integration Explorer

Operating system	Installation wizard name
Linux on x86	install
Linux on x86-64	install
Windows 32-bit	install.exe
Windows 64-bit	install.exe

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Installing the Integration Bus component” on page 69

Use the installation wizard to install the Integration Bus component.

“Installing the IBM Integration Toolkit” on page 79

Use the installation wizard graphical interface to install the IBM Integration Toolkit on Windows and Linux on x86.

“Installing IBM Integration Explorer” on page 84

To use IBM Integration Explorer only, without installing the complete IBM Integration Toolkit, use the IBM Integration Explorer installation wizard to install the IBM Integration Explorer.

Related reference:

“Installation and uninstallation interfaces” on page 652

You can use different interfaces for installation and uninstallation of the Integration Bus component and the IBM Integration Toolkit.

Installation and uninstallation authorization

Check the authorization requirements for the user ID that you use to install or uninstall the Integration Bus component and the IBM Integration Toolkit.

Check that your user ID has the following authority to complete installation or uninstallation tasks:

- On AIX systems, you must log in as root to set up the security requirements for installing the Integration Bus component. The installation can then be performed by a user who does not have root privileges. For more information, see “Security on Linux and UNIX systems” on page 45.
- On Linux and on other UNIX systems, your user ID does not need to have root authority to install or uninstall the IBM Integration Toolkit. However, you will require a user ID with root authority to carry out the post installation and pre-uninstallation steps required for IBM Integration Toolkit.

You need root level authority to set up the security requirements for installing the Integration Bus component. The installation can then be performed by a user who does not have root privileges. If you do not plan to run the Integration Bus component installation as a user with root authorities, you must complete additional steps. For more information about the extra tasks that you must complete to run the Integration Bus component installer as a user without root authorities, see “Security on Linux and UNIX systems” on page 45.

- On all Windows operating systems and editions, your user ID does not need to be a member of the Administrators group to install or uninstall the IBM Integration Toolkit. However, you will require a user ID with Administrator authority to carry out the post install and pre-uninstallation steps required for IBM Integration Toolkit.

Your user ID must be a member of the Administrators group to install the Integration Bus component.

- On z/OS systems, your user ID must be no more than eight characters in length. It must also have suitable privileges to install in your environment with SMP/E. Use a supported external security manager, for example RACF or ACF2, to grant the required privileges. Your user ID must have a valid OMVS segment, because the product installs into the file system paths specified during the SMP/E APPLY processing.
- On distributed platforms, when uninstalling the Integration Bus component, you must be root or the same user ID that you used to install the Integration Bus component.

Related tasks:

“Installing” on page 31

Installation information for IBM Integration Bus is provided in the IBM Integration Bus Installation Guide.

“Applying service to the Integration Bus component” on page 130

Apply maintenance updates and program fixes to the Integration Bus component.

“Uninstalling the Integration Bus component” on page 151

You can uninstall the Integration Bus component on distributed systems in a number of ways.

“Applying service to the IBM Integration Toolkit” on page 141

Apply maintenance updates and program fixes to the IBM Integration Toolkit.

“Uninstalling the IBM Integration Toolkit” on page 159

Choose the method that you want to use to uninstall the IBM Integration Toolkit and follow the instructions to remove the component.

Related reference:

“Installation and uninstallation interfaces” on page 652

You can use different interfaces for installation and uninstallation of the Integration Bus component and the IBM Integration Toolkit.

Multicultural support

Multicultural support is available for a selection of languages on both distributed systems and z/OS.

The user interface and message catalogs are provided in the following languages on distributed systems:

- Brazilian Portuguese
- French
- German
- Italian
- Japanese
- Korean
- Simplified Chinese
- Spanish
- Traditional Chinese
- US English
- Polish
- Russian
- Turkish

The message catalogs are provided in the following languages on z/OS:

- Japanese
- Simplified Chinese
- US English

The messages that are written to the z/OS operator console (which are a subset of the messages that are written to the syslog) are in US English only, and are written in mixed case or in uppercase depending on your chosen system configuration.

IBM Integration Bus provides a selection of message catalogs that are used by the product components to report any problems that occur. Products that are used in conjunction with IBM Integration Bus might cause IBM Integration Bus to report errors using its message catalogs, or might report problems using their own techniques.

You must refer to the documentation supplied with any other products that you use to determine the process they employ. In particular, you must check the documentation that is supplied by the databases that you use and documentation that is provided with any user-defined node or parser that you integrate into the IBM Integration Bus environment.

You can install IBM Integration Bus and WebSphere MQ in any supported language; all language versions for each product are compatible with all language versions for the other product. All product language versions are included on the DVD supplied with IBM Integration Bus.

All messages that are generated for internal inter-component message exchange (for example, deployed configuration messages and log files for **mqsireadlog**) are generated in code page 1208 (utf-8).

Related tasks:



Converting code page and message encoding

You can use ESQL within a Compute node to convert data for code page and message encoding.

“Changing locales” on page 536

You can change the locale for the system on which a runtime component is installed.

Locales

Message support is provided in a number of locales.

IBM Integration Bus supports at least the following locales:

Windows	AIX	Solaris	HP-UX ¹	Linux ²	z/OS
English (United States)	en_US	en_US	en_US.iso88591, en_US.roman8	en_US	En_US.IBM-1047, En_US.IBM-037
German (Standard)	de_DE, De_DE	de	de_DE.ISO88591, de_DE.roman8	de_DE	not supported
Spanish (Modern Sort)	es_ES, Es_ES	es	es_ES.ISO88591, es_ES.roman8	es_ES	not supported
French (Standard)	fr_FR, Fr_FR	fr	fr_FR.ISO88591, fr_FR.roman8	fr_FR	not supported
Italian (Standard)	it_IT, It_IT	it	it_IT.ISO88591, it_IT.roman8	it_IT	not supported
Portuguese (Brazilian)	pt_BR, Pt_BR	pt_BR	pt_BR.ISO88591, pt_BR.utf8	pt_BR	not supported
Japanese	Ja_JP, ja_JP	ja_JP.PCK, ja	ja_JP.SJIS, ja_JP.eucJP	ja_JP	Ja_JP.IBM-939, Ja_JP.IBM-930
Simplified Chinese (China)	Zh_CN, zh_CN	zh, zh.GBK	zh_CN.hp15CN	zh_CN	Zh_CN.IBM-1388, Zh_CN.IBM-935
Traditional Chinese (Taiwan)	Zh_TW, zh_TW	zh_TW, zh_TW.BIG5	zh_TW.big5, zh_TW.eucTW	zh_TW	not supported
Korean	ko_KR	ko	ko_KR.eucKR	ko_KR	not supported

Notes:

1. Because of limited syslog support on HP-Itanium operating systems, messages are written to the log in US English only.
2. These values are the same for all Linux systems.

Other locales might be supported; check your operating system for further details.

Related tasks:

“Changing locales” on page 536

You can change the locale for the system on which a runtime component is installed.

Related reference:

“Supported processors” on page 620

IBM Integration Bus is supported on multiple processors.

“Operating system requirements” on page 627

IBM Integration Bus is supported on multiple operating systems.



Supported code pages

Application messages must conform to supported code pages.

System changes caused by installation and configuration

When you install IBM Integration Bus, the installation program makes certain changes to your computer.

Resources are also affected by how you configure your system (for example, the brokers that you create).

- Start and main menu updates
- Registry contents
- Environment variables
- Default WebSphere MQ resources

Related tasks:

“Finding the latest information” on page 1
Access the latest information for IBM Integration Bus.

Related reference:

“Software requirements” on page 625
View the operating system, database, and other software requirements.

“Hardware requirements” on page 620
View the processor and related hardware requirements on all platforms.

“Supported processors” on page 620
IBM Integration Bus is supported on multiple processors.

“Operating system requirements” on page 627
IBM Integration Bus is supported on multiple operating systems.

“Supported databases” on page 628
You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

Start and main menu updates after installation

When you install IBM Integration Bus components on Linux on x86 or Windows, the installation program updates the main or Start menus.

IBM Integration Toolkit on Linux on x86 running Red Hat

The main menu is populated with the following entries:

- **Programming > IBM Integration Toolkit**
- **Programming > IBM Installation Manager**

If you install multiple products with Installation Manager on this operating system, a single entry exists in the main menu and refers to the last product that you installed. The package groups of installed products are not reflected in the main menu. To invoke applications that you installed previously, including other installations of the IBM Integration Toolkit, you must use the command-line interface.

If you want to start Installation Manager, for example to update or uninstall one or more packages, you cannot use the main menu items, because these actions require root authority. Become root and start the Installation Manager program `/eclipse/IBMIM` in the Installation Manager installation directory on the command line.

IBM Integration Toolkit on Linux on x86 running SUSE

The main menu is populated with the following entries:

- **All Applications > Development > IBM Integration Toolkit**
- **All Applications > Development > IBM Installation Manager**

If you install multiple products with Installation Manager on this operating system, a single entry exists in the main menu and refers to the last product that you installed. The package groups of installed products are

not reflected in the main menu. To invoke applications that you installed previously, including other installations of the IBM Integration Toolkit, you must use the command-line interface.

If you want to start Installation Manager, for example to update or uninstall one or more packages, you cannot use the main menu items, because these actions require root authority. Become root and start the Installation Manager program `/eclipse/IBMIM` in the Installation Manager installation directory on the command line.

IBM Integration Toolkit on Windows

The **Start** menu is populated with the following entries:

- **IBM Software Development Platform > IBM Integration Toolkit > IBM Integration Toolkit**
- **IBM Software Development Platform > IBM Integration Toolkit > Release Notes**
- **IBM Software Development Platform > IBM Integration Toolkit > Start Toolkit Help**
- **IBM Software Development Platform > IBM Integration Toolkit > Stop Toolkit Help**

These entries assume that you have installed into the first or default package group. Additional products that you install in this package group might also appear in this start list.

Entries are also added to the **Start** menu when Installation Manager is first installed:

- **Programs > IBM Installation Manager > IBM Installation Manager**
- **Programs > IBM Installation Manager > Release Notes**
- **Programs > IBM Installation Manager > Uninstall IBM Installation Manager**
- **Programs > IBM Installation Manager > View Installed Packages**

IBM Integration Explorer on Linux on x86

No additions are made to the main menu on either SUSE or Red Hat Linux systems.

IBM Integration Explorer on Windows

The **Start** menu is populated with the following entry:

- **Programs > IBM Integration Explorer**

Integration Bus component on Windows

The **Start** menu is populated with the following entries:

- **Start > All Programs > IBM Integration Bus 9.0.0.0 > IBM Integration Console 9.0**
- **Start > All Programs > IBM Integration Bus 9.0.0.0 > IBM Integration Java APIs > IBM Integration API Documentation**
- **Start > All Programs > IBM Integration Bus 9.0.0.0 > IBM Integration Java APIs > IBM Integration CMP API Exerciser**
- **Start > All Programs > IBM Integration Bus 9.0.0.0 > IBM Integration Java APIs > IBM Integration Java API Documentation**

Related tasks:

“Finding the latest information” on page 1

Access the latest information for IBM Integration Bus.

Related reference:

“Software requirements” on page 625

View the operating system, database, and other software requirements.

“Hardware requirements” on page 620

View the processor and related hardware requirements on all platforms.

“Supported processors” on page 620

IBM Integration Bus is supported on multiple processors.

“Operating system requirements” on page 627

IBM Integration Bus is supported on multiple operating systems.

“Supported databases” on page 628

You can optionally configure databases to contain data that is accessed by your message flows. Databases from IBM and other suppliers are supported at specific versions on supported operating systems.

Registry changes created by installation and configuration

When you install IBM Integration Bus, the installation program creates a number of entries in a registry.

Runtime components

On Windows, the ALLUSERPROFILE environment variable is used; on Linux and UNIX systems, equivalent values are stored within the installation directory structure. On all platforms, the environment variable MQSI_REGISTRY is set to point to the exact location. Do not alter or remove these entries unless instructed to do so by your IBM Service representative.

On Windows, some entries are also added to the system registry under HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI (for 32-bit brokers on 32-bit operating system editions and 64-bit brokers on 64-bit operating system editions) or under HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBCINST.INI (for 32-bit brokers on 64-bit operating system editions); these entries record the installed 32-bit database drivers for Oracle and Sybase, which contain driver locations and parameters.

A final entry is added under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\IBM Integration Bus v*** for Event Log information, where v*** is the current three-digit product version number.

IBM Integration Toolkit

All registry entries are controlled by IBM Installation Manager. On Windows, the system registry is used; on Linux on x86, equivalent entries are stored in /var/ibm/InstallationManager/. Do not alter or remove these entries unless instructed to do so by your IBM Service representative.

The installation directory for Installation Manager is stored in HKEY_LOCAL_MACHINE\SOFTWARE\IBM\Installation Manager (for 32-bit operating system editions) or HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IBM\Installation Manager (for 64-bit operating system editions), under the location value.

Uninstall keys are created for all products that are installed by Installation Manager under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Current\Version\Uninstall\IBM Installation Manager, under the name IM-*packagegroupname* where *packagegroupname* is the name of the package group in

which the product is installed; for example, IBM Software Development Platform.

IBM Integration Explorer

Related reference:

“Supported processors” on page 620

IBM Integration Bus is supported on multiple processors.

“Operating system requirements” on page 627

IBM Integration Bus is supported on multiple operating systems.

Environment variables after installation

On distributed systems, ensure that your environment is set up correctly.

If you have installed on z/OS, see “Customizing the z/OS environment” on page 290.

Sample profile files are provided with IBM Integration Bus; check their content to ensure the variables are set correctly for your environment before you use the product or configure any resources.

- On Linux and UNIX systems, the profile is *install_dir/bin/mqsipprofile*.
- On Windows systems, a command console is set up when you install components. Access this through the **Start** menu. When you select this option, a command window containing the correct environment is opened.

If you prefer, run *install_dir\bin\mqsipprofile.cmd* in a command window before working with IBM Integration Bus in the same command window.

If you have more than one installation on a single system, ensure that you run the correct profile, or access the correct command console, for the installation that you want to work with. If an installation is at Version 8.0, the profile is incompatible with the Version 9.0 profile. You must log off and log on again before you run a second profile.

Check the readme file (*readme.html*) to ensure that you have the latest version of the profile.

Related tasks:

Chapter 5, “Configuring brokers for test and production environments,” on page 289

Create one or more brokers on one or more computers, and configure them on your test and production systems to process messages that contain your business data.

“Setting up a command environment” on page 119

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.



Are the Linux and UNIX environment variables set correctly?

Use the **mqsipprofile** command to set a command environment.

Default WebSphere MQ resources created during installation and configuration

When you install IBM Integration Bus and create components, WebSphere MQ resources are created for use by those components.

The names of these resources begin with the reserved characters SYSTEM. The resources are additional to the default WebSphere MQ objects that are created when you install that product. The following table lists the resources that are created. You must not change or delete any of these resources.

Resource name	Type	Description
SYSTEM.BROKER.ADAPTER.FAILED	Queue	Failure messages that are generated by adapters.
SYSTEM.BROKER.ADAPTER.INPROGRESS	Queue	In progress messages that are generated by adapters.
SYSTEM.BROKER.ADAPTER.NEW	Queue	New request messages that are generated by adapters.
SYSTEM.BROKER.ADAPTER.PROCESSED	Queue	Processed messages that are generated by adapters.
SYSTEM.BROKER.ADAPTER.UNKNOWN	Queue	Unknown messages that are generated by adapters.
SYSTEM.BROKER.ADMIN.QUEUE	Queue	Target for request messages that are sent by CMP applications (including commands) to modify the configuration and operation of the broker.
SYSTEM.BROKER.ADMIN.REPLYTODM	Queue	Target for internal messages that are sent by the broker.
SYSTEM.BROKER.AGGR.CONTROL	Queue	Used internally to store a control message for each aggregate group.
SYSTEM.BROKER.AGGR.REPLY	Queue	Used internally to store each known response message that is received by an AggregateReply node.
SYSTEM.BROKER.AGGR.REQUEST	Queue	Used internally to store each request message that forms part of an aggregate group.
SYSTEM.BROKER.AGGR.TIMEOUT	Queue	Used internally in the timeout and unknown timeout processing within an AggregateReply node.
SYSTEM.BROKER.AGGR.UNKNOWN	Queue	Stores each unknown response that is received by an AggregateReply node.
SYSTEM.BROKER.AUTH	Queue	Stores authorization records for administration requests against the broker.
SYSTEM.BROKER.CD.MODEL	Queue	Used as a model queue for CD nodes.
SYSTEM.BROKER.CONTROL.QUEUE	Queue	Used for broker control messages.
SYSTEM.BROKER.DC.AUTH	Queue	Stores authorization records for administration requests associated with record and replay.
SYSTEM.BROKER.DC.BACKOUT	Queue	Target for any publications that cannot be read by the record and replay resource manager.
SYSTEM.BROKER.DC.RECORD	Queue	Target for publications that are received as a result of the record and replay resource manager's subscriptions.
SYSTEM.BROKER.DEPLOY.QUEUE	Queue	Target for publish/subscribe control requests that applications send to the broker.
SYSTEM.BROKER.DEPLOY.REPLY	Queue	Target for response messages that are sent by publish/subscribe control requests that applications send to the broker.
SYSTEM.BROKER.DIRECT.QUEUE	Queue	Used to communicate directly with unresponsive execution groups.
SYSTEM.BROKER.DIRECT.REPLY	Queue	Target for responses from unresponsive execution groups.
SYSTEM.BROKER.EDA.COLLECTIONS	Queue	Used internally to store event handler information for message collections that are generated by the Collector node.
SYSTEM.BROKER.EDA.EVENTS	Queue	Used internally to store messages that are received by the Collector node.
SYSTEM.BROKER.EXECUTIONGROUP.QUEUE	Queue	Target for messages that are sent to integration servers by internal broker processes.
SYSTEM.BROKER.EXECUTIONGROUP.REPLY	Queue	Target for response messages that are sent by integration servers to internal broker processes.
SYSTEM.BROKER.FTE.MODEL	Queue	Used as a model queue for FTE nodes.

Resource name	Type	Description
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	Queue	Target for publish/subscribe messages that are sent by neighbor brokers.
SYSTEM.BROKER.MB.TOPIC	Topic	Used to publish broker changes.
SYSTEM.BROKER.MODEL.QUEUE	Queue	Model for dynamic response queues.
SYSTEM.BROKER.SEQ.EXPIRY	Queue	Used by sequence nodes to hold sequence expiry data.
SYSTEM.BROKER.SEQ.GROUP	Queue	Used by sequence nodes to hold sequence group data.
SYSTEM.BROKER.SEQ.NUMBER	Queue	Used by sequence nodes to hold sequence number data.
SYSTEM.BROKER.TIMEOUT.QUEUE	Queue	Used internally to support the TimeoutControl and TimeoutNotification nodes.
SYSTEM.BROKER.WEBADMIN.SUBSCRIPTION	Queue	Used to send live updates to the IBM Integration Bus web user interface server to refresh the display of objects.
SYSTEM.BROKER.WS.ACK	Queue	Used internally for Web Services client support.
SYSTEM.BROKER.WS.INPUT	Queue	Used internally for Web Services client support.
SYSTEM.BROKER.WS.REPLY	Queue	Used internally for Web Services client support.
SYSTEM.BKR.CONFIG	SVRCONN	A connection channel for IBM Integration Toolkit and IBM Integration Explorer clients.

Related reference:

“Designing the WebSphere MQ infrastructure” on page 16

You must create and manage the WebSphere MQ resources that are required to support your brokers, and the applications that connect to them to supply or receive messages.

“Software requirements” on page 625

View the operating system, database, and other software requirements.

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 grant 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 106-0032, 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 the program(s) described in this publication at any time without notice.

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

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

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 Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

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 herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for 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 claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

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

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software for use with this program.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Important: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol ([®] or [™]), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and trademark information (www.ibm.com/legal/copytrade.shtml).



Figure 19. Java compatibility logo

Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To make comments about the functions of IBM products or systems, talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, to this address:

User Technologies Department (MP095)
IBM United Kingdom Laboratories
Hursley Park
WINCHESTER,
Hampshire
SO21 2JN
United Kingdom

- By fax:
 - From outside the U.K., after your international access code use 44-1962-816151
 - From within the U.K., use 01962-816151
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
 - IBMLink: HURSLEY(IDRCF)
 - Internet: idrcf@hursley.ibm.com

Whichever method you use, ensure that you include:

- The publication title and order number
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.



Printed in USA