

MQSeries<sup>®</sup> Integrator for Windows NT<sup>®</sup>



# Installation Guide

*Version 2.0.1*



MQSeries<sup>®</sup> Integrator for Windows NT<sup>®</sup>



# Installation Guide

*Version 2.0.1*

**Note!**

This document contains information of a proprietary nature. All information contained herein shall be kept in confidence. This information should not be divulged to persons other than IBM employees authorized by the nature of their duties to receive such information, or individuals and organizations authorized by the IBM Corporation in accordance with existing policy regarding release of company information.

Before using this information and the product it supports, be sure to read the general information under "Appendix E. Notices" on page 141.

**Third edition (November 2000)**

This edition applies to IBM MQSeries Integrator for Windows NT Version 2.0.2 and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2000. All rights reserved.**

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

---

# Contents

<b>Figures.</b> . . . . .	<b>v</b>		Migration considerations. . . . .	20
<b>Tables.</b> . . . . .	<b>vii</b>		MQSeries Integrator Version 1 . . . . .	21
			MQSeries Integrator Version 2.0 . . . . .	23
			MQSeries Publish/Subscribe . . . . .	25
<b>About this book</b> . . . . .	<b>ix</b>			
Who this book is for . . . . .	ix			
What you need to know to understand this book. . . . .	ix			
Terms used in this book . . . . .	x			
Where to find more information. . . . .	x			
MQSeries Integrator publications . . . . .	x			
MQSeries publications. . . . .	xi			
MQSeries Publish/Subscribe publications . . . . .	xii			
MQSeries Workflow publications . . . . .	xii			
DB2 publications . . . . .	xii			
MQSeries information available on the Internet . . . . .	xiii			
<b>Summary of changes</b> . . . . .	<b>xv</b>			
Changes for this edition (GC34-5600-02) . . . . .	xv			
<b>Chapter 1. What is MQSeries Integrator?</b> . . . . .	<b>1</b>			
MQSeries Integrator overview . . . . .	1			
The Configuration Manager . . . . .	1			
Message brokers . . . . .	1			
Message flows . . . . .	2			
Messages . . . . .	3			
The Control Center . . . . .	3			
Applications . . . . .	4			
<b>Chapter 2. Planning for installation</b> . . . . .	<b>5</b>			
System setup . . . . .	5			
Hardware requirements . . . . .	5			
Software requirements . . . . .	7			
Database summary. . . . .	10			
License information . . . . .	11			
National language support . . . . .	11			
Setup types . . . . .	11			
Product components . . . . .	12			
Primary components . . . . .	12			
Secondary components . . . . .	14			
Security considerations . . . . .	17			
Using Windows NT primary or trusted security domains . . . . .	18			
			<b>Chapter 3. Installing MQSeries Integrator</b> . . . . .	<b>27</b>
			Delivery media . . . . .	27
			Preparing for installation. . . . .	28
			Before you start. . . . .	28
			Installation procedure. . . . .	29
			System update information . . . . .	33
			Setting up for LAN installation . . . . .	33
			What to do if something goes wrong during installation . . . . .	34
			Silent installation . . . . .	35
			Recording a response file. . . . .	35
			Running the silent installation . . . . .	36
			Response File . . . . .	36
			Detecting Errors from a silent install . . . . .	37
			<b>Chapter 4. Getting Started with MQSeries Integrator</b> . . . . .	<b>41</b>
			Configuring a simple broker domain. . . . .	41
			Designing a simple configuration . . . . .	44
			Setting up user IDs and groups . . . . .	46
			Creating and connecting to the databases . . . . .	48
			Setting up database authorizations . . . . .	50
			Configuring your broker domain . . . . .	52
			Checking the components . . . . .	58
			Starting your broker domain . . . . .	59
			Verifying your installation . . . . .	63
			Preparing for verification . . . . .	63
			Running the predefined verification applications . . . . .	67
			Building and using a simple message flow . . . . .	74
			<b>Chapter 5. A more complex configuration</b> . . . . .	<b>79</b>
			The components of the complex configuration . . . . .	79
			Preparing to extend the broker domain . . . . .	80
			Stopping the Control Center, broker, and Configuration Manager . . . . .	80
			Changing the broker and Configuration Manager . . . . .	80
			Creating new resources . . . . .	82
			Creating the User Name Server . . . . .	82

Creating MQSeries resources . . . . .	83	Collecting information . . . . .	115
Restarting your broker domain. . . . .	85	Schema installation . . . . .	115
Starting the listeners and channels . . . . .	85	Creating tablespaces . . . . .	115
Starting the User Name Server . . . . .	87	DB2 tablespaces . . . . .	116
Restarting the broker . . . . .	87	Configuring DB2 . . . . .	116
Restarting the Configuration Manager . . . . .	87	Oracle tablespaces. . . . .	118
Restarting the Control Center . . . . .	87	Configuring Oracle7 . . . . .	119
<b>Appendix A. System changes after</b>		Step 1: creating the database . . . . .	119
<b>installation . . . . .</b>	<b>89</b>	Step 2: configuring a client connection . . . . .	120
Directory structure . . . . .	89	Step 3: creating tablespaces . . . . .	122
Environment variables . . . . .	90	Configuring Oracle8 . . . . .	124
Windows NT registry . . . . .	91	Step 1: creating the database . . . . .	124
Database contents . . . . .	92	Step 2: configuring a client connection . . . . .	124
Default MQSeries resources . . . . .	95	Step 3: creating table spaces . . . . .	127
Principals . . . . .	96	User-Defined Segments (for Sybase and SQL	
<b>Appendix B. Uninstalling MQSeries</b>		Server 6.5) . . . . .	128
<b>Integrator . . . . .</b>	<b>99</b>	Setting up the server. . . . .	129
Before you start . . . . .	99	Sybase . . . . .	129
Using the uninstall program . . . . .	101	SQL Server 6.5 . . . . .	131
Migration considerations . . . . .	102	Installing the database schema (all	
Manually uninstalling MQSeries Integrator	102	databases) . . . . .	133
Unattended Removal of IBM MQSeries		Editing the database connection file. . . . .	134
Integrator Version 2.0.1 on Windows NT . . . . .	104	Migrating rules and formats . . . . .	135
MQSeries Integrator stanza . . . . .	105	NNRie overview . . . . .	135
Contacting your IBM Support Center . . . . .	106	NNFie overview . . . . .	136
<b>Appendix C. Applying maintenance . . . . .</b>	<b>109</b>	Migrating your rules and formats . . . . .	136
Applying the maintenance information . . . . .	109	Setting up the environment . . . . .	137
Restoring a previous service level . . . . .	111	Checking consistency . . . . .	137
Installing new components after applying		Setting the system environment variable . . . . .	138
maintenance . . . . .	111	Exporting rules and formats . . . . .	139
Checking the service level . . . . .	111	Modifying your environment for import . . . . .	139
Applying maintenance to IBM DB2		Importing rules and formats into your	
Universal Database . . . . .	112	Version 2 database . . . . .	140
<b>Appendix D. Using NEONFormatter and</b>		Using NEONRules and NEONFormatter . . . . .	140
<b>NEONRules nodes . . . . .</b>	<b>113</b>	<b>Appendix E. Notices . . . . .</b>	<b>141</b>
Setting up the database environment . . . . .	113	Trademarks . . . . .	143
DB2 . . . . .	113	<b>Glossary of terms and abbreviations . . . . .</b>	<b>145</b>
Oracle. . . . .	114	<b>Index . . . . .</b>	<b>151</b>
SQL Server 6.5 . . . . .	114	<b>Sending your comments to IBM . . . . .</b>	<b>155</b>
Sybase . . . . .	114		

---

## Figures

1. MQSeries Integrator for Windows NT installation procedure . . . . .	30	7. Configuring your broker domain	52
2. Configuring a simple broker domain	42	8. Create Configuration Manager: screen 1	53
3. Setting up user IDs and groups on Windows NT . . . . .	46	9. Create Configuration Manager: screen 2	53
4. Creating and connecting to the databases . . . . .	48	10. Create broker: screen 1 . . . . .	56
5. Defining the ODBC connections on Windows NT . . . . .	49	11. Create broker: screen 2 . . . . .	57
6. Setting up database authorizations	50	12. Starting your broker domain. . . . .	59
		13. Starting the Control Center . . . . .	62
		14. Modify Configuration Manager. . . . .	81
		15. Modify broker . . . . .	81
		16. Create User Name Server. . . . .	82





---

## Tables

	1.	Component disk space requirements	6		7.	Non-zero return codes in the MQSI2.log	39
	2.	Supported databases for brokers and user data . . . . .	10		8.	Directory structure after installation	89
	3.	Summary of installation options . . . . .	16		9.	Database tables for brokers . . . . .	92
	4.	Summary of authorizations . . . . .	19		10.	Database tables for the configuration repository . . . . .	93
	5.	Parameters for silent installation and response files . . . . .	36		11.	Database tables for the message repository . . . . .	94
	6.	Result codes for silent installation.	38		12.	MQSeries Integrator default objects	95



---

## About this book

This book provides an overview of IBM MQSeries Integrator Version 2.0.1. It explains how to plan for, install, and verify installation of the product.

“Chapter 1. What is MQSeries Integrator?” on page 1 provides a brief overview of the concepts and capabilities of MQSeries Integrator Version 2.0.1 at a high level.

“Chapter 2. Planning for installation” on page 5 describes the preparation you need to complete prior to product installation.

“Chapter 3. Installing MQSeries Integrator” on page 27 provides detailed installation information for MQSeries Integrator for Windows NT Version 2.0.1.

“Chapter 4. Getting Started with MQSeries Integrator” on page 41 explains how you complete product setup, using a simple installation to illustrate the tasks you need to complete. It also helps you to deploy your broker network and verify its operation using supplied verification programs.

Appendixes cover the configuration established by the default installation options, guidelines for preparing to use the NEONFormatter and NEONRules nodes, and details of servicing and removing the product.

A glossary is also provided.

For further information about the product, and planning for its use, refer to the *MQSeries Integrator Version 2 Introduction and Planning* book.

For details of administrative tasks, including configuration and problem determination, see the *MQSeries Integrator Version 2 Administration Guide*.

---

## Who this book is for

This book is for administrators of systems on which MQSeries Integrator Version 2.0.1 components will be installed and tested.

---

## What you need to know to understand this book

To understand this book, you need to be familiar with the system facilities of Windows NT. You also need to be familiar with the administration facilities of MQSeries for Windows NT V5.1.

## About this book

The *MQSeries for Windows NT V5.1 Quick Beginnings*, *MQSeries System Administration*, and *MQSeries Integrator Version 2 Administration Guide* books provide useful reference information for installation and post-installation tasks.

---

## Terms used in this book

All references to MQSeries Integrator are to MQSeries Integrator Version 2.0.1 unless otherwise stated.

All new terms introduced in this book are defined in “Glossary of terms and abbreviations” on page 145. These terms are shown like *this* at their first use.

The book uses the following shortened names:

- MQSeries: a general term for IBM MQSeries messaging products.
- MQSeries Publish/Subscribe: the MQSeries Publish/Subscribe SupportPac™ available on the Internet for several MQSeries server operating systems (the Internet URL is given in “MQSeries information available on the Internet” on page xiii).
- DB2®: a general term to encompass IBM DB2 Universal Database® Enterprise Edition, Connect Enterprise Edition, and Extended Enterprise Edition.

---

## Where to find more information

Becoming familiar with the MQSeries Integrator library will help you accomplish MQSeries Integrator tasks quickly. The library covers planning, installation, administration, and client application tasks.

The library also contains references to complementary product libraries, including the MQSeries Family library.

**Note:** If you cut and paste examples of commands from the Portable Document File (PDF) of a book, to a command line for execution, you must check that the content is correct before you press the Enter key. Some characters might be corrupted by local system and font settings.

## MQSeries Integrator publications

The following books make up the MQSeries Integrator Version 2.0.1 library:

- IBM MQSeries Integrator Version 2 Introduction and Planning, GC34-5599
- IBM MQSeries Integrator for Windows NT Version 2 Installation Guide, GC34-5600 (this book)
- IBM MQSeries Integrator Version 2 Messages, GC34-5601
- IBM MQSeries Integrator Version 2 Using the Control Center, SC34-5602

- IBM MQSeries Integrator Version 2 Programming Guide, SC34-5603
- IBM MQSeries Integrator Version 2 Administration Guide, SC34-5792

This book is provided in hardcopy with the product. The *MQSeries Integrator Introduction and Planning* book is also available in hardcopy.

All books in the MQSeries Integrator Version 2.0.1 library are provided in softcopy, in Adobe Portable Document Format (PDF) in a searchable PDF library. You can:

- Install the library (by doing a full installation or by specifying the Documentation component on a custom installation).
- Access the library directly from the Docs subdirectory under the root directory on the supplementary CD-ROM without installing them.
- Access the library after installation by selecting *Start->Programs->IBM MQSeries Integrator 2.0.1->Documentation*.

The MQSeries Integrator Version 1.1 publications are also supplied as PDFs and can be installed with MQSeries Integrator Version 2.0.1 (the Documentation component). They can also be retrieved from the MQSeries Web site given in “MQSeries information available on the Internet” on page xiii.

- IBM MQSeries Integrator Version 1.1 Installation and Configuration Guide, GC34-5503
- IBM MQSeries Integrator Version 1.1 User’s Guide, GC34-5504
- IBM MQSeries Integrator Version 1.1 System Management Guide, SC34-5505
- IBM MQSeries Integrator Version 1.1 Programming Reference for NEONRules, SC34-5506
- IBM MQSeries Integrator Version 1.1 Programming Reference for NEONFormatter, SC34-5507
- IBM MQSeries Integrator Version 1.1 Application Development Guide, SC34-5508

You can read PDFs using Adobe Acrobat Reader or in a Web browser (with Acrobat Reader as a plug-in). Version 4 is required. You can also print your own copies of these books.

You can download a free copy of Acrobat Reader from the Adobe Web site at <http://www.adobe.com>

### MQSeries publications

The following books are referred to in this book to point you to the information you need to complete MQSeries Messaging product tasks as part of MQSeries Integrator tasks.

## MQSeries family publications

For Windows NT installation tasks you might need:

- IBM MQSeries for Windows NT V5.1 Quick Beginnings, GC34-5389.  
This book is included, in hardcopy, in the MQSeries Integrator package.  
It provides detailed planning and installation guidance.

For planning and configuration tasks you might need:

- IBM MQSeries Command Reference, SC33-1369.  
This book contains the syntax of the MQSC commands.
- IBM MQSeries System Administration, SC33-1873.  
This book supports day-to-day management of local and remote MQSeries objects.
- IBM MQSeries Clients, GC33-1632.  
This book describes how to install, configure, use, and manage MQSeries clients.

For a complete list of MQSeries product publications, refer to the information on the MQSeries Web site (given in “MQSeries information available on the Internet” on page xiii).

### MQSeries Publish/Subscribe publications

If you have installed MQSeries Publish/Subscribe, and plan to migrate to MQSeries Integrator Version 2, or to establish a mixed broker network, refer to the following publication:

- IBM MQSeries Publish/Subscribe User’s Guide, GC34-5269

This book and the MQSeries Publish/Subscribe SDK package are available on the MQSeries Web site (given in “MQSeries information available on the Internet” on page xiii).

### MQSeries Workflow publications

The MQSeries Workflow product has a comprehensive library. Refer to the following book for introductory information, and for details about other product publications:

- IBM MQSeries Workflow Concepts and Architecture, GH12-6285

For a complete list of MQSeries Workflow product publications, refer to the information on the MQSeries Web site (given in “MQSeries information available on the Internet” on page xiii).

### DB2 publications

The following DB2 publications are referenced in this book.

- IBM DB2 Quick Beginnings, GC09-2835
- IBM DB2 Message Reference, GC09-2846
- IBM DB2 Troubleshooting Guide, SI0J-8169

## MQSeries family publications

You can download these publications from the DB2 Web site at  
<http://www.ibm.com/software/data/db2>

### **MQSeries information available on the Internet**

The MQSeries Business Solution, of which MQSeries Integrator is a part, has a Web site at:

<http://www.ibm.com/software/ts/mqseries>

By following links from this Web site you can:

- Obtain the latest information about all MQSeries family products.
- Access all the books for the MQSeries family products.
- Down-load MQSeries SupportPacs.

## MQSeries family publications



---

## Summary of changes

This section describes changes in this edition of *MQSeries Integrator for Windows NT Installation Guide*. Changes since the previous edition of the book are marked by vertical lines to the left of the changes.

---

### Changes for this edition (GC34-5600-02)

Changes for this edition include:

- Additional information about the following:
  - Migration from MQSeries Integrator Versions 1 and 2.0.
  - Setting the MQSI\_PARAMETERS\_FILE environment variable.
- Minor technical and editorial improvements throughout the book

## Changes

---

## Chapter 1. What is MQSeries Integrator?

This chapter provides an overview of the product and describes some of its features. It also provides pointers to more detailed information on these features. A detailed introduction to MQSeries Integrator is given in the *MQSeries Integrator Introduction and Planning* book.

---

### MQSeries Integrator overview

MQSeries Integrator Version 2.0.1 is a *message broker* product, addressing the needs of business and application integration through management of information flow. It provides services that allow you to:

- Route a message to several destinations, using rules that act on the contents of one or more of the fields in the message or message header.
- Transform a message, so that applications using different formats can exchange messages in their own formats.
- Store and retrieve a message, or part of a message, in a database.
- Modify the contents of a message (for example, by adding data extracted from a database).
- Publish a message to make it available to other applications. Other applications can choose to receive publications that relate to specific topics, or that have specific content, or both.
- Extend the function of MQSeries Integrator Version 1.

These services are based on the messaging transport layer provided by the MQSeries products: communications between all components in the broker domain are provided by MQSeries.

### The Configuration Manager

An MQSeries Integrator system is controlled by the *Configuration Manager*. The components and resources managed by the Configuration Manager constitute the *broker domain*. The Configuration Manager maintains the broker domain configuration in the *configuration repository*. You can create and modify this configuration using the facilities of the *Control Center*, a graphical user interface described in “The Control Center” on page 3.

### Message brokers

The broker domain includes one or more message brokers, usually referred to as brokers, that support services that you create to handle your message traffic. You can install and configure any number of brokers, and can support the same or different services in each one. Brokers can be installed on

## MQSeries Integrator overview

different machines (that may be running different operating systems) to the Control Centers and configuration managers.

Brokers can be connected together to provide communication throughout your broker domain. This ensures messages such as publications can be propagated to all subscribers (see “Applications” on page 4).

### Message flows

When you design your broker domain, you decide what services need to be defined and supported by your brokers. Each service performs a set of actions, or *rules*, on each message, between receipt of the message by the broker, and delivery of the message to the target applications.

Each action, or subset of actions, is implemented as a *message processing node*, and these are grouped together in a sequence to form a *message flow*. You create message flows using the Control Center (described in “The Control Center” on page 3).

Message flows can range from the very simple, performing just one action on a message, to the complex, performing a number of actions on the message to transform its format and content. A message flow can process one message in several ways to deliver a number of output messages, perhaps with different format and content, to a number of target applications.

The message flows you create receive messages at **MQInput** nodes. Every input node represents an MQSeries queue, and every message flow must have at least one input node.

Message flows usually complete their activity by sending one or more messages to one or more recipients from **MQOutput** nodes that represent MQSeries queues, or from **Publication** nodes that redistribute the message to interested subscribers using MQSeries queues. Other message flows might simply store the message in a database for later processing, and not use an output node at all.

The other nodes between input and output provide the actions you want taken against the messages. MQSeries Integrator supplies a number of predefined message processing nodes. For example, Filter (on message data content) and Compute (for example, add data from a database) nodes are provided. These supplied nodes are described in *MQSeries Integrator Using the Control Center*.

You can create new nodes, using a system programming interface supplied by MQSeries Integrator, to provide other options for message processing. This interface is described in the *MQSeries Integrator Programming Guide*.

### Messages

Each message flowing through your system has a specific content and structure, referred to as a *message template*. Message template information identifies the structure of the data it contains. You can define the structure using the Control Center, or the NEONFormatter graphical interface, before you use messages with these structures. Alternatively, you can use messages in XML (Extensible Markup Language) format that are *self-defining*.

You can group related messages together in a *message set*. All message set definitions are maintained in the broker domain's *message repository*. This repository is managed by the Configuration Manager.

Message template information for predefined messages is usually included in the message header, so the message flows recognize the messages when they receive them. Other messages might not use the expected header, but you can set up your message flow input nodes to indicate how the messages will be processed.

The message bit-stream is decoded by *message parsers*. MQSeries Integrator supplies several message parsers ready for use on known message templates and message headers.

You can create new parsers, using a system programming interface supplied by MQSeries Integrator, if you need to process other types of message. This interface is described in the *MQSeries Integrator Programming Guide*.

### The Control Center

The functions and facilities of MQSeries Integrator are controlled using a graphical interface known as the Control Center. You can use the Control Center to:

- Define your broker domain
- Work with message flows
- Create your messages
- Control your publish/subscribe network
- Manage your broker domain

The Control Center allows you to restrict access and authority to the functions it provides, so you can control who can do what within the broker domain.

The Control Center comes with comprehensive online help, including the *MQSeries Integrator Tour*. The Tour gives you an online overview of the MQSeries Integrator product, its components, and the Control Center interface itself. The Tour is based on a simple example scenario, in which MQSeries Integrator is used to integrate the processes of an international company.

The Tour introduces the product in three ways:

## MQSeries Integrator overview

1. Providing introductory information that you can read, with links to MQSeries information on the worldwide web.
2. Providing animated sequences of actions in the Control Center. For example, you can see how a message flow and message set are created using the Control Center.
3. Creating objects in your own Control Center workspace so that you can experiment with them yourself later.

You can find a full description of the Control Center in *MQSeries Integrator Using the Control Center*.

### Applications

Applications using messages to send or receive data can communicate in several ways. Today, most MQSeries applications are using *point-to-point* communications. These applications might be using a one-way *send-and-forget* (or *datagram*) model, or a *request/reply* (client/server) model. These messages can be sent to a message flow that you have established in a broker, to carry out the required processing on the message before sending it on to the receiving application.

Brokers support a second type of communication model known as *publish/subscribe*. In this model, some applications (*publishers*) provide information, and others (*subscribers*) consume that information. You can also create applications that are both publishers and subscribers.

Publishers create messages and send them to one or more message flows at a local broker that supports publish/subscribe. Each message has an associated topic that categorizes the information in the message. Subscribers register subscriptions with their local broker, specifying the types of publication they are interested in (determined, for example, by the topic of the message, or its content, or both). When a broker receives a publication that matches a subscription that has been registered, it sends that published message to the subscriber. Brokers exchange subscriptions and publications with each other, so that subscribers can receive information published at any connected broker in the broker domain.

New and existing applications can take advantage of the broker functions through the MQSeries *Message Queue Interface* (MQI), or the MQSeries *Application Messaging Interface* (AMI). Both interfaces support point-to-point and publish/subscribe programming models. If you have existing applications written to these interfaces, you'll find they can usually run unchanged in this new message broker environment.

You can find information about the design of new applications, and the reuse of existing applications, in *MQSeries Integrator Introduction and Planning*.

---

## Chapter 2. Planning for installation

Successful installation of MQSeries Integrator involves three steps:

1. Planning and preparation:

Careful planning of your installation will help you clarify your requirements and the actions needed to achieve the environment you want. All aspects of planning are covered in full in this chapter, and include the following:

- a. "System setup".
- b. "Setup types" on page 11.
- c. "Product components" on page 12.
- d. "Security considerations" on page 17.
- e. "Migration considerations" on page 20.

2. Installation:

When you have decided which components you want to install, follow the guidance in "Chapter 3. Installing MQSeries Integrator" on page 27. The installation program checks for the prerequisite products, if any, required by the components you choose.

3. Configuration:

When you have completed installation, you need to perform some initial configuration. These additional tasks will allow you to define and activate the resources in your installation. These steps are covered in "Chapter 4. Getting Started with MQSeries Integrator" on page 41.

A simple configuration is used to illustrate the tasks needed and the results of the steps taken. A set of simple tests that verify that the installation has worked is also described.

---

### System setup

This section provides details of the prerequisite products for installation, and related planning and setup information.

#### Hardware requirements

The hardware requirements for MQSeries Integrator for Windows NT are listed in the following sections.

#### General requirements

The general hardware requirements for MQSeries Integrator for Windows NT are:

- Any Year 2000 compliant Intel® Pentium II (or above) processor-based IBM PC machine or compatible, that is explicitly compatible and fully capable of

## System setup

running the specified operating system, all the corresponding supporting software shown below, and any associated applications unmodified.

- Any communications hardware supporting NetBIOS, SNA LU 6.2, SPX, and TCP/IP.
- A minimum of 512 MB of RAM to support run-time operation of all components on a single system.

### Control Center requirements

The Control Center is a specialized graphical program that runs on Windows NT. It has more demanding requirements on the operating environment.

The following setup is recommended for systems on which it is installed:

- Any Year 2000 compliant Intel Pentium II (or above) processor-based IBM PC machine or compatible, with 500+MHz processor speed
- A minimum of 256 Megabytes (MB) of RAM

### Disk space required

The installation requirements depend on which components you install and how much working space you need. This in turn depends primarily on your use of MQSeries resources such as queues and persistent messages.

Table 1 gives the component storage requirements in MB:

*Table 1. Component disk space requirements*

Component	MB
Broker	146
Configuration Manager	147
Control Center	223
User Name Server	148
Online documentation	36
Samples and SDK	2
NEON support	60
TOTAL	510

Components other than the online documentation and Samples and SDK, each include a set of common files. The figures in Table 1 reflect this. Therefore, if you install all components you will find that the total storage is less than the sum of the individual components. This is shown in the total line. This is also true if you install more than one of these components.

After installation, you must create databases for the configuration repository, the message repository, and the broker tables, each of which requires



approximately 10MB of space. If you have user databases (for example, for NEONFormatter message formats), you will require additional space for these.

You also need from 150MB to 300MB of additional space for temporary files (approximately 150MB for a broker or full installation, 300MB for a custom installation). This space is required in the location pointed to by the TEMP system variable, not in the directory into which you install the product. These temporary files are deleted when installation has completed.

If you are installing DB2 as part of MQSeries Integrator installation, you need an additional 250MB.

### Software requirements

Minimum supported levels are shown. Later compatible levels, if any, are supported unless otherwise stated.

#### Prerequisite software

The following products are prerequisites:

- Microsoft Windows NT Version 4.0, including TCP/IP, NetBIOS, and SPX, with Service Pack 5 or Service Pack 6A, either of which provides relevant Year 2000 fixes and Euro support.

Both Windows NT Workstation and Windows NT Server products are supported. MQSeries Integrator installs on both NTFS and FAT file systems.

**Note:** Service Pack 6 is not supported.

The installation program checks that you have the correct level of operating system installed. If not, you are recommended to exit the program and upgrade your system. You can continue the installation and upgrade your system later, but you must do this before you can use MQSeries Integrator. You can download Windows NT upgrades from the Microsoft Web site:

<http://support.microsoft.com/directory/>

- IBM MQSeries for Windows NT Version 5.1.

This must be at service level Corrective Service Diskette (CSD) 4.<sup>1</sup> The product and CSD are supplied in the MQSeries Integrator package.

You should look at `readme.txt` on the product CD-ROM for the latest levels of software required.

The MQSeries components required depend on the MQSeries Integrator components you choose to install. For details, see “Primary components” on page 12.

---

1. If you are planning to establish an integrated broker network that includes both MQSeries Integrator and MQSeries Publish/Subscribe, see also “MQSeries Publish/Subscribe” on page 25.

## System setup

The installation program checks that you have MQSeries for Windows NT Version 5.1 installed, and that it is at the correct service level. If you choose to install the Control Center, it also checks if you have the correct version of the MQSeries Client for Java™. This can be either the client code installed from the MQSeries server CD, or the client supplied by MQSeries Supportpac MA88.

If you do not have the MQSeries components you need, you are recommended to install these before you continue with MQSeries Integrator installation.

**Note:** Version 5.0 is not supported at any service level.

MQSeries for Windows NT requires a number of other software products to install and operate a server successfully.

If you install only an MQSeries client with your MQSeries Integrator components, check the client installation details in the MQSeries Release Notes folder to determine the client's prerequisites.

MQSeries for Windows NT server prerequisites are:

- Internet Explorer Version 4.01 with service pack 1  
This is available from the Microsoft Web site:  
<http://www.microsoft.com>
- Active Directory Services Interface Version 2.0.  
This is provided on the MQSeries CD.
- Microsoft Management Console Version 1.1.  
This is provided on the MQSeries CD.
- A database product.

For a summary of the supported databases see Table 2 on page 10.

MQSeries Integrator broker and Configuration Manager components require access to a database for internal caching and for storing internal control information. The remaining components do not need access to a database.

If your installation choices require a database to be present, the MQSeries Integrator installation program checks for a suitable database installed on this system.

If the installation program detects that you have a level of DB2 prior to that indicated here, it highlights the need to upgrade your existing license and lists the supported levels. You can continue with installation, but you must upgrade your database before you can use MQSeries Integrator.

If you do not have DB2 installed, the MQSeries Integrator installation program launches the installation program for DB2 Version 6.1, which is included on the MQSeries Integrator CD.

DB2 has no additional prerequisite products, but it does require an additional 250MB of disk storage.

**Note:** This DB2 product has **restricted license terms and agreements**. You must only use this DB2 installation in association with your licensed use of MQSeries Integrator for message management, and only the MQSeries Integrator components can make calls to the DB2 database.

The use of a database by the MQSeries Integrator components is independent of the use of databases by your applications and message flows. You are not restricted to the databases listed here for application and data storage and retrieval. Fully globally coordinated message flows that use a DB2 database are only supported using DB2 Version 6.1. See also “Migration considerations” on page 20.

### Optional products

The following products are options, not prerequisites.

- Connectivity

These products support communications between MQSeries queue managers, and between clients and queue managers.

- IBM Communications Server for Windows NT Version 5.0
- IBM Personal Communications for Windows NT Version 5.0
- Attachmate Extra! Personal Client Version 6.1 or Version 6.2
- Microsoft SNA Server Version 2.11 or Version 3.0

- Databases

For a summary of the supported databases see Table 2 on page 10.

The NEONRules and NEONFormatter subcomponents of the broker support message definitions created and maintained in a number of databases.

**Note:** These databases are for message definitions created through the NEONFormatter only. The databases required for internal product use are listed earlier in this section.

- MQSeries Integrator Tour

If you intend to run the MQSeries Integrator Tour (from the help menu of the Control Center), the following products are required:

- Microsoft Internet Explorer Version 5.  
This is available from the Microsoft Web site:  
<http://www.microsoft.com>
- Java Plugin 1.1.2\_001  
This is available on the MQSeries Integrator CD.
- Lotus Media  
This is available on the MQSeries Integrator CD.

## System setup

- Application programming support

The following software compilers are supported:

- IBM VisualAge® e-business for Windows Version 1.01.
- IBM VisualAge for C++ for Windows Version 4.0.
- IBM C and C++ Compiler Version 3.6.4.
- IBM VisualAge COBOL for Windows NT Version 2.1.
- IBM VisualAge COBOL Enterprise Version 2.2
- IBM VisualAge for Java eBusiness Edition for Windows 95 and Windows NT.
- IBM VisualAge for Java Enterprise Version 2.0.
- IBM VisualAge for Java Professional Version 2.0.
- Microsoft Visual C++ for Windows 95 and Windows NT Version 6.0.
- Microsoft Visual Basic for Windows Version 4.0 (16 bit) or 5.0 (32 bit).
- Micro Focus Object COBOL for Windows NT Version 3.3 or Version 4.0.
- Micro Focus COBOL Version 3.1J for Windows NT.

**Note:** If you have installed VisualAge for Java, and selected the MQSeries Connector as part of that installation, you must ensure that the CLASSPATH entry for VisualAge for Java appears **after** the CLASSPATH entries for MQSeries for Windows NT Version 5.1 (server or Java client). This is to ensure that the Configuration Manager accesses the correct MQSeries classes not the VisualAge classes.

## Database summary

Table 2 lists the databases supported.

*Table 2. Supported databases for brokers and user data*

Database	AIX	Sun Solaris	Windows NT
DB2 <sup>1 3</sup>	6.1 <sup>4</sup>	6.1 <sup>4</sup>	5.2 plus fixpack 12 6.1 <sup>4</sup>
Microsoft SQL Server	not applicable	not applicable	6.5 plus SP 5a 7.0 plus SP1
Oracle	7.3.4 <sup>2</sup> 8.1.5	7.3.4 <sup>2</sup> 8.1.5	7.3.4 <sup>2</sup> 8.1.5
Sybase	11.5 12 <sup>5</sup>	11.5 12 <sup>5</sup>	11.5 12 <sup>5</sup>

**Notes:**

1. DB2 6.1 is the only DBMS supported by MQSeries Integrator that permits a database to participate as a Resource Manager in a distributed XA transaction, and coordinated by MQSeries as the XA Transaction Manager. In MQSeries Integrator, this is referred to as supporting a globally coordinated message flow.
2. Oracle 7.3.4 is not supported for use as a broker internal database.
3. You must use DB2 for the configuration and message repository databases maintained by the Configuration Manager. No other database is supported for this purpose.
4. Please check the Readme.txt file for your product to check if a Fixpack is required.
5. Sybase 12 is not supported by NEONRules and NEONFormatter.

## License information

Under the terms of the MQSeries Integrator Version 2.0.1 license agreement, you can install one instance of each component at any one time on any one system, with the exception of the Control Center. You can install the Control Center on multiple systems providing that each Control Center is interacting with the same single Configuration Manager. You can create multiple brokers on a single system.

## National language support

MQSeries Integrator Version 2.0.1 is enabled for national language support, but the user interface and message catalogs are currently available in US English only.

MQSeries Integrator Version 2.0.1 can process and construct messages in any code page supported by MQSeries for Windows NT Version 5.1.

**Note:** The NEONRules and NEONFormatter nodes support only the Latin1 code page in ASCII and EBCDIC. If you include these nodes within a message flow, this might restrict the messages that can be processed.

MQSeries Integrator interacts with MQSeries installed in any supported language. All languages for the MQSeries messaging products are included on the single MQSeries for Windows NT Version 5.1 CD.

All messages generated for internal intercomponent message exchange are generated in code page 1208.

DB2 Version 6.1 is NLS-enabled.

---

## Setup types

MQSeries Integrator contains a number of components that work together to provide the full function of a broker domain. You can choose to install any component on any system running Windows NT that can be connected to the others using MQSeries.

You are recommended initially to install all components on a single system. This gives you a good understanding of what is installed, and how to configure and start up a broker domain. The configuration guidance given in “Configuring a simple broker domain” on page 41 assumes that you have installed in this way. If you want details about how to set up more complex broker domains, you can work through “Chapter 5. A more complex configuration” on page 79, and refer to the *MQSeries Integrator Introduction and Planning* book for planning tasks, and the *MQSeries Integrator Administration Guide* for implementation tasks.

## Setup types

When you install, you must select a *setup type*, that defines the components to be installed. Three setup types are available:

- **Full**

A full installation installs all the components on the current system. This type is the default.

The simple configuration used to illustrate post-installation configuration, activation, and testing assumes you have chosen to do a full installation on one system.

- **Broker only**

This type provides the simplest installation. The only components installed by this option are the broker and NEON support. If you are setting up a broker domain distributed across many different physical systems, you are likely to use this option.

- **Custom**

This type allows you to select exactly the components you want. You are presented with a list of valid choices, and can select one or more. Selecting all components results in a full installation.

You can use this setup type to add extra components to an already installed system.

**Note:** If you choose to install a new component, or to replace an existing component, after you have installed any CSDs for MQSeries Integrator, a warning dialog is displayed indicating that you must reinstall the CSD after you have installed the new component. This action ensures product code consistency is maintained.

---

## Product components

MQSeries Integrator has four primary components and three secondary components. A set of common files is also installed with each major component.

The dependencies of each of these components are described: these are summarized in Table 3 on page 16.

### Primary components

The four primary components can be installed separately and independently, or all together, depending on the environment you want to establish. For a functional description of these components, see “Chapter 1. What is MQSeries Integrator?” on page 1.

The four major components are:

- **Broker.** You can install the broker component on one or more systems subject to your license agreement (see “License information” on page 11 for

details). You can configure and activate any number of brokers on each of the systems on which you install the component, subject to system resource constraints.

- **Configuration Manager.** You must have only one Configuration Manager in your broker domain. Install this component on one system, and create your Configuration Manager.
- **Control Center.** You can install and use the Control Center on any number of systems in your broker domain subject to your license agreement (see “License information” on page 11 for details).
- **User Name Server.** You are recommended to install and configure one User Name Server within your broker domain.

The order in which you install the components of MQSeries Integrator within your broker domain is not important. However, successful activation and operation of the broker domain does depend on order: details are given in “Chapter 4. Getting Started with MQSeries Integrator” on page 41.

### **The broker**

Each broker requires its own MQSeries queue manager. You must therefore install the MQSeries server component from the MQSeries CD if it is not already installed on every system on which you install the broker.

This option installs NEON support automatically. You also have the option of installing NEON support on its own, which allows you to use the NEON rules and formats Graphical User Interfaces on other systems.

If you plan to set up more than one broker on a single system, you need only install the component once, but you must create and configure each broker individually. Each broker requires its own queue manager. However, a single queue manager can host a single broker, and the Configuration Manager or the User Name Server, or both, if the appropriate component is installed and created on the same system as the broker.

Each broker requires access to a database to create and maintain internal data in tables. The tables hold information about the broker’s current configuration (for example, the message flows that are assigned to it). The database can be DB2, Oracle, or SQL Server. If it is DB2, you are advised to use a local database server for performance reasons, although client connection to a remote DB2 server is supported. If you use a DB2 client connection, you must consider network loading and reliability because delays will significantly impact the performance in the broker domain.

### **The Configuration Manager**

The Configuration Manager requires a MQSeries queue manager. You must therefore install the MQSeries server component from the MQSeries CD if it is not already installed.

## Product components

The Configuration Manager requires access to a database to create and maintain internal data in tables. The two sets of tables required can be created in a single database or can be set up in separate databases:

- One set holds configuration and definition information for the whole broker domain, and is known as the configuration repository.
- The second set holds definition information for messages defined or imported through the Control Center, and is known as the message repository.

**The database in which you create these repositories must be DB2.** You are advised to use a local database server for performance reasons, although client connection to a remote DB2 server is supported. If you use a DB2 client connection, you must consider network loading and reliability because delays will significantly impact performance in the broker domain.

### The Control Center

The Control Center requires the MQSeries Client for Java. You can install this client from either of the following two sources:

- The MQSeries for Windows NT Version 5.1 server CD. Select the “Custom” installation.
- The MQSeries MA88 SupportPac. You can download this SupportPac from the MQSeries Web site (given in “MQSeries information available on the Internet” on page xiii).

The Control Center includes the *MQSeries Integrator Tour*. If you invoke the Tour (from the help menu) you must have Microsoft Internet Explorer Version 5 installed, and you must install Java plug-in 1.1.2\_001 and Lotus Media from the MQSeries Integrator CD. For more information about the Tour, see “The Control Center” on page 3.

### The User Name Server

The User Name Server requires an MQSeries queue manager to be assigned to it. You must therefore install the MQSeries server component from the MQSeries CD if it is not already installed.

The User Name Server does not require access to a database.

## Secondary components

There are three optional components that you can install:

- Samples and Software Developers’ Kit (SDK)
- Online documentation
- NEON interface

These components have no prerequisites, and have no MQSeries connectivity requirements. You can install these components without a previous installation of MQSeries or a database.



### Samples and Software Developers' Kit (SDK)

This component comprises a set of application samples, and samples that illustrate how to use the plug-in extensions.

- Application samples

These applications illustrate the basic techniques of application programming to take advantage of the full range of MQSeries Integrator function.

- Sample programs

Sample programs are supplied in C and Java. These programs are fully operational, and both source and executables are supplied.

If you choose to use these samples and run them in your broker domain, you must ensure they are running in an environment in which MQSeries connectivity is available. Check the details of the operating systems and application programming languages supported by MQSeries clients, and by applications local to queue managers.

You can also copy and modify these examples to create your own applications, or you can add sections of their code to existing applications to exploit MQSeries Integrator function.

- The verification programs **Scribble**, **Postcard**, and **Results Service** are provided to help you test out your initial installation. These are described in "Running the predefined verification applications" on page 67 .

The set of programs that make up the **Results Service** application are used in the *MQSeries Integrator Programming Guide* to illustrate the various publish/subscribe programming techniques available to your application programmers.

- Libraries and header files

Library files required for building applications are included in this component.

Headers required by applications written to the Message Queue Interface (MQI) or Application Messaging Interface (AMI) are included. Their use is illustrated in the application samples.

- Software Developers' Kit

This kit contains working examples of the plug-in extensions that you can create to enhance MQSeries Integrator. Source code is provided to illustrate the programming to use the system interfaces introduced by MQSeries Integrator, for both message parser and message processing node. Executable code is also provided. The headers and library files required by parsers and message processing nodes are also supplied.

Parsers and processing nodes execute only on a system on which an MQSeries Integrator broker is installed.

## Product components

### Online documentation

Information for MQSeries Integrator is provided for online viewing using the Acrobat Reader application from Adobe. Every information unit is supplied in Portable Document Format (PDF). A searchable library in PDF, which provides a cross book index and search facility, is also provided. You can access the documentation without installing: the publications are in the \Docs subdirectory in the root directory of the supplementary CD.

You can download a free copy of Acrobat Reader (which must be at Version 4) from the Adobe Web site at

<http://www.adobe.com>

This component can be installed on any system, including one that has no other MQSeries Integrator component installed. For example, you can choose to install one copy of the documentation on a central LAN server for all users to share.

For details of all publications supplied, see “MQSeries Integrator publications” on page x.

### NEON interface

This option enables you to install NEON support on its own to allow you to run the Neon rules and formats for migration to installed brokers.

### Summary of component installation options

Table 3 provides a summary of MQSeries Integrator components.

*Table 3. Summary of installation options*

Product	Component	System to install on	
MQSeries Integrator for Windows NT	All except online documentation	Windows NT only	
	Online documentation	Windows NT	Windows NT only
		Unix	Copy from /Docs on CD

### Additional products

If you intend to use the MQSeries Integrator Tour (described in “The Control Center” on page 3), you must install two plug-ins products. These are supplied on the MQSeries Integrator CD, and you are given the option to install them after installing the MQSeries Integrator components.

The two products are:

- Java Plugin 1.1.2\_001
- Lotus Media

The Tour provides animated sequences that depend on these two plug-ins. You are recommended to install both plug-ins if you intend to use the Tour on this system.

---

### Security considerations

Security control of MQSeries Integrator components, resources, and tasks depends on the definition of users and groups of users (*principals*) to the security subsystem of the operating system (the Windows NT User Manager). MQSeries Integrator always creates a set of groups on the system on which it is installed. These local groups are:

- **mqbrkrs**
- **mqbrasgn**
- **mqbrdevt**
- **mqbrops**
- **mqbrtpic**

You must assign users (or other groups) to these local groups to allow them to perform specific tasks. The authorizations required are summarized in Table 4 on page 19.

The principals you define must be eight characters or less if you are running MQSeries Integrator in an environment that includes clients on heterogeneous platforms. If you have a homogeneous Windows NT environment, you can use up to 12 characters (an MQSeries limit). For more details about these limits, see *MQSeries Integrator Introduction and Planning*.

MQSeries Integrator draws principals either from the Windows NT local account security domain, or a Windows NT primary domain, or a Windows NT trusted domain. Principals must be defined to a specific Windows NT security domain. You must decide which domain you want to use for MQSeries Integrator, and define your principals to that domain (using the Windows NT User Manager on the security domain server). If you already have a security domain set up to control access to MQSeries resources, you are advised to use this same domain for MQSeries Integrator: this will not cause any conflict and will ease your security administration.

For a more general discussion on security for MQSeries Integrator see *MQSeries Integrator Introduction and Planning*. For further details about Windows NT security domains, see the information on the Microsoft Web site at

<http://www.microsoft.com/ntserver/security/deployment/default.asp>

## Security and principals

In particular, you are advised to review the section entitled *Security Deployment Resources Roadmap*.

### Using Windows NT primary or trusted security domains

If you are using a primary or trusted security domain, MQSeries Integrator requires global groups defined to your primary or trusted security domain that mirror the local groups that are created during installation. The global groups are:

- **Domain mqbrkrs**
- **Domain mqbrasgn**
- **Domain mqbrdevt**
- **Domain mqbrops**
- **Domain mqbrtpic**

These groups must be made members of the local security domain's equivalent MQSeries Integrator groups (**Domain mqbrkrs** must be a member of **mqbrkrs**, and so on).

- If you install MQSeries Integrator on the domain controller of a primary or a trusted security domain, the MQSeries Integrator installation program creates the local and global groups, and adds the global groups to the local groups.

If you do not intend to install MQSeries Integrator on the domain controller, you can create these groups yourself using the Windows NT User Manager.

- If you install MQSeries Integrator on a workstation member of a primary security domain, the MQSeries Integrator installation program creates the local groups. If the global groups already exist in the primary security domain, it also adds each global group to the appropriate local group in the local domain.
- If you install MQSeries Integrator on a workstation member of a trusted domain, MQSeries Integrator cannot recognize the trusted domain, and does not add the global groups to the local groups. You must do this step yourself.
- If you install MQSeries Integrator on a workstation that is a member of both a trusted security domain and a primary security domain, the installation program creates the local groups. If the global groups already exist in the primary security domain, it also adds each global group to the appropriate local group in the local domain. It cannot detect the trusted domain and therefore does not add the global groups of the trusted security domain to the local groups. If you want these trusted security domain global groups in the local groups instead of, or in addition to, the primary security global groups, you must make these updates yourself.

When you define a new user ID to your security domain, you must assign this ID to the domain group that it authorized for the tasks this user ID is to perform, so that it is authorized globally.

Table 4 summarizes the security requirements for the major tasks. It illustrates what group memberships are required if you are using a local security domain defined on your local system **SALONE**, or a primary domain named **PRIMARY**, or a trusted domain named **TRUSTED**. The contents of this table assume that you have created both the Configuration Manager and the User Name Server with the same security domain.

*Table 4. Summary of authorizations*

User is...	Local domain (SALONE)	Primary Domain (PRIMARY)	Trusted domain (TRUSTED)
Installing	<ul style="list-style-type: none"> <li>Member of <b>Administrators</b></li> </ul>	Not applicable.	Not applicable.
Uninstalling	<ul style="list-style-type: none"> <li>Member of <b>Administrators</b></li> </ul>	Not applicable.	Not applicable.
Creating broker, Configuration Manager, User Name Server	<ul style="list-style-type: none"> <li>Must be a user ID defined in SALONE</li> <li>Member of <b>Administrators</b></li> </ul>	<ul style="list-style-type: none"> <li>Must be a user ID defined in PRIMARY</li> <li>Member of <b>SALONE\Administrators</b></li> </ul>	<ul style="list-style-type: none"> <li>Must be a user ID defined in TRUSTED</li> <li>Member of <b>SALONE\Administrators</b></li> </ul>
Starting broker, Configuration Manager, User Name Server	<ul style="list-style-type: none"> <li>Member of <b>Administrators</b></li> </ul>	Not applicable.	Not applicable.
Running User Name Server (service user ID)	<ul style="list-style-type: none"> <li>Must be a user ID defined in SALONE</li> <li>Member of <b>mqbrkrs</b></li> </ul>	<ul style="list-style-type: none"> <li>Must be a user ID defined in PRIMARY</li> <li>Member of <b>PRIMARY\Domain mqbrkrs</b></li> </ul>	<ul style="list-style-type: none"> <li>Must be a user ID defined in TRUSTED</li> <li>Member of <b>TRUSTED\Domain mqbrkrs</b></li> </ul>
Running Configuration Manager (service user ID)	<ul style="list-style-type: none"> <li>Must be a user ID defined in SALONE</li> <li>Member of <b>mqbrkrs</b></li> <li>Member of <b>mqm</b></li> </ul>	<ul style="list-style-type: none"> <li>Must be a user ID defined in PRIMARY</li> <li>Member of <b>PRIMARY\Domain mqbrkrs</b></li> <li>Member of <b>SALONE\mqm</b> (see note 1)</li> </ul>	<ul style="list-style-type: none"> <li>Must be a user ID defined in TRUSTED</li> <li>Member of <b>TRUSTED\Domain mqbrkrs</b></li> <li>Member of <b>SALONE\Domain mqm</b> (see note 2)</li> </ul>
Running broker (service user ID) (see note 5)	<ul style="list-style-type: none"> <li>Must be a user ID defined in SALONE</li> <li>Member of <b>mqbrkrs</b></li> </ul>	<ul style="list-style-type: none"> <li>Must be a user ID defined in PRIMARY</li> <li>Member of <b>PRIMARY\Domain mqbrkrs</b></li> </ul>	<ul style="list-style-type: none"> <li>Must be a user ID defined in TRUSTED</li> <li>Member of <b>TRUSTED\Domain mqbrkrs</b></li> </ul>
Running Control Center (see note 3)	<ul style="list-style-type: none"> <li>Must be a user ID defined in SALONE (see note 4) For example, <b>SALONE\User1</b> is valid, <b>PRIMARY\User2</b> and <b>TRUSTED\User3</b> are not</li> <li>Member of one or more of <b>mqbrasgn, mqbrdevt, mqbrps, mqbrtpic</b></li> </ul>	<ul style="list-style-type: none"> <li>Must be a user ID defined in PRIMARY (see note 4) For example, <b>PRIMARY\User2</b> is valid, <b>SALONE\User1</b> and <b>TRUSTED\User3</b> are not.</li> <li>Member of one or more of <b>PRIMARY\Domain mqbrasgn, PRIMARY\Domain mqbrdevt,</b> and so on.</li> </ul>	<ul style="list-style-type: none"> <li>Must be a user ID defined in TRUSTED (see note 4) For example, <b>TRUSTED\User3</b> is valid, <b>SALONE\User1</b> and <b>PRIMARY\User2</b> are not.</li> <li>Member of one or more of <b>TRUSTED\Domain mqbrasgn, TRUSTED\Domain mqbrdevt,</b> and so on.</li> </ul>

## Security and principals

Table 4. Summary of authorizations (continued)

User is...	Local domain (SALONE)	Primary Domain (PRIMARY)	Trusted domain (TRUSTED)
Running publish/subscribe applications	<ul style="list-style-type: none"><li>• Must be a user ID defined in SALONE For example, SALONE\User1 is valid, PRIMARY\User2 and TRUSTED\User3 are not.</li></ul>	<ul style="list-style-type: none"><li>• Must be a user ID defined in PRIMARY For example, PRIMARY\User2 is valid, SALONE\User1 and TRUSTED\User3 are not.</li></ul>	<ul style="list-style-type: none"><li>• Must be a user ID defined in TRUSTED For example, TRUSTED\User3 is valid, SALONE\User1 and PRIMARY\User2 are not.</li></ul>
<b>Notes:</b> <ol style="list-style-type: none"><li>1. If you are running in a primary domain, you can also:<ul style="list-style-type: none"><li>• Define the user ID in the domain PRIMARY.</li><li>• Add this ID to the group <b>PRIMARY\Domain mqm</b>.</li><li>• Add the <b>PRIMARY\Domain mqm</b> group to the group <b>SALONE\mqm</b>.</li></ul></li><li>2. If you are running in a trusted domain, you can also:<ul style="list-style-type: none"><li>• Define the user ID in the domain TRUSTED.</li><li>• Add this ID to the group <b>TRUSTED\Domain mqm</b>.</li><li>• Add the <b>TRUSTED\Domain mqm</b> group to the group <b>SALONE\mqm</b>.</li></ul></li><li>3. All Control Center users need read access to the MQSeries java\lib subdirectory of the MQSeries home directory (the default is X:\Program Files\MQSeries, where X: is the operating system disk). This access is restricted to users in the local group <b>mqm</b> by MQSeries. MQSeries Integrator installation overrides this restriction and gives read access for this subdirectory to all users.</li><li>4. If a valid user ID is defined in the domain used by the Configuration Manager (for example, PRIMARY\User4) an identical user ID defined in a different domain (for example, DOMAIN2\User4) will be able to access the Control Center with the authorities of PRIMARY\User4.</li><li>5. The broker can be run as an MQSeries trusted application. If it is, security requirements are changed. See the <i>MQSeries Integrator Administration Guide</i> for full details.</li></ol>			

You can find a more comprehensive summary in the *MQSeries Integrator Administration Guide*.

---

### Migration considerations

This section provides planning information for installation if you intend to migrate from, or coexist with, one of the following related products:

- MQSeries Integrator Version 1
- MQSeries Integrator Version 2.0
- MQSeries Publish/Subscribe

It also describes the action you must take prior to installation of MQSeries Integrator.

This information is *installation-specific*, relevant to pre-installation tasks and the installation process itself. For details of planning and post-installation actions and options, you must refer to the migration information in *MQSeries*

*Integrator Introduction and Planning*, and to “Appendix D. Using NEONFormatter and NEONRules nodes” on page 113.

## MQSeries Integrator Version 1

You are not required to take any specific action at installation time. However, you are recommended to take the following actions to preserve your existing data and enable you to go back to MQSeries Integrator Version 1 if you decide to do so for any reason.

- Rules and formats
  1. If you are migrating from Version 1.1, backup existing Version 1.1 rules and formats using standard database backup facilities. This data can be used unchanged with Version 2.
  2. If you are migrating from MQSeries Integrator Version 1.0 to Version 2.0.1, your formats and rules are not compatible in their current form. If you wish to reuse your existing Version 1.0 rules and formats, you must migrate these to Version 1.1 format. Before you can migrate these formats, you must configure your databases. For details of the actions you need to take to complete this migration see “Appendix D. Using NEONFormatter and NEONRules nodes” on page 113. This step **must** be completed before you uninstall Version 1.0.

- Adding new rules and fomats

Because the rules and formats created using the MQSeries Integrator Version 1.1 NEONRules and NEONFormatter user interface tools are fully compatible with MQSeries Integrator Version 2.0.1 components, you can continue to use these programs to add new rules and formats.

If you prefer to use the MQSeries Integrator Version 2.0.1 facilities for defining rules and formats, you must use the Control Center. For a full description of these and other tasks supported by the Control Center, see the *MQSeries Integrator Using the Control Center* book.

- Access to rules and formats

The definitions of the rules and formats in the database identified in the MQSIRuleeng.mpf configuration file are needed by every broker in which you deploy a message flow that includes the NEONRules or NEONFormatter message processing nodes.

These definitions are not distributed through your broker network in the same way as the formats and rules defined by the MQSeries Integrator Version 2.0.1 Control Center.

You must therefore ensure that the brokers that need to access these definitions can do so:

- Ensure that the system on which the broker is installed has client access to the system on which the database is installed.

Make sure that the file identified in the MQSI\_PARAMETERS\_FILE environment variable contains the correct information to connect to the

## Migration

database. See the *MQSeries Integrator Version 1.1 System Management Guide* for more details about the contents of this file.

- User exits

If you have set up user exits for the NEONFormatter, you must do the following if you want to reuse them with the NEONFormatter message processing node in MQSeries Integrator Version 2.0.1:

- If you are currently using user exits with MQSeries Integrator Version 1.1, your exits will work unchanged.
- If you are currently using user exits with MQSeries Integrator Version 1.0, refer to the *MQSeries Integrator Version 1.1 Programming Reference for NEONFormatter* for instructions on how to use your user exits with MQSeries Integrator Version 1.1. They will then work with MQSeries Integrator Version 2.0.1.

- Logs and log records

Log records generated by the NEONRules and NEONFormatter message processing nodes are written to the log file defined for the message flow in which the nodes appear.

For details of how to specify a log file for a message flow, see the *MQSeries Integrator Using the Control Center* book.

For details of the content of log records generated by the NEONRules and NEONFormatter nodes, see the *MQSeries Integrator Version 1.1 System Management Guide*.

- Backup files

The configuration files established for Version 1.0 and Version 1.1 can both be reused unchanged with Version 2.0.1.

Back up the following files:

1. sqlsvses.cfg
2. MQSIruleng.mpf
3. MQSIputdata.mpf
4. MQSIgetdata.mpf

If you have user exit programs, you are recommended to back up these files too.

- Uninstall Version 1

If you choose to uninstall Version 1.1, you must select the option that preserves your existing database entries. This ensures that all current rules and formats are still available for use with Version 2. Uninstallation of Version 1 is optional.

**Note:** The MQSeries Integrator for Windows NT Version 2 installation program detects an existing installation of Version 1.0 and Version 1.1 and removes its PATH entry because the products share some file names. You must therefore take this into consideration if you want to



continue to use any processes that invoke Version 1 function. If you want to revert to using Version 1, you must replace its entry in the PATH statement.

MQSeries Integrator Version 2.0.1 currently supports the following databases:

- IBM DB2 Universal Database Version 5.0 or later
- Microsoft SQL Server
- Oracle Version 7.3.4 or Version 8.1.5
- Sybase 11

NEONFormatter and NEONRules data continue to be supported by the same set of databases as Version 1.1. You are required to indicate the database you are using with your Version 1.1 setup during the installation process.

For further guidance of migration after installation, see “Appendix D. Using NEONFormatter and NEONRules nodes” on page 113, the *MQSeries Integrator Introduction and Planning* book.

## MQSeries Integrator Version 2.0

These are the tasks you must complete following a migration from MQSeries Integrator Version 2.0 to MQSeries Integrator Version 2.0.1.

### Upgrading your Control Center

In a scenario where there are different versions of MQSeries Integrator Version 2 components, all components from both versions can be used together except that a Version 2.0.1 Control Center will not work with a Version 2.0 Configuration Manager. This combination results in message BIP1360 being issued.

You are recommended to upgrade all Control Centers to Version 2.0.1 when any broker or the Configuration Manager are upgraded to Version 2.0.1. This enables new messages to be correctly displayed by the Control Center. As an alternative, you may achieve the same effect by putting the Version 2.0.1 message catalog file `MQSIv201.properties` into the `Tool` directory of each Version 2.0 Control Center.

When upgrading your Control Center to Version 2.0.1 you are recommended to first check in any message flows that you have newly created or have checked out. When they are checked out again after the upgrade, they will reflect any changes made to IBM supplied nodes.

If you import a workspace into a Version 2.0.1 Control Center that was created by exporting from a Version 2.0 Control Center, three new IBM supplied nodes will be missing from the imported workspace. You can add these using *Message Flow->Add to workspace*.

## Migration

### Upgrading your brokers

Your brokers must be upgraded to pick up modifications to broker database tables. Your brokers must be stopped while this upgrade is performed. The upgrade command, `mqsimigratebroker`, operates at the broker database level, so if you have several brokers sharing the same database then the command need only be issued once. You are advised to back up your broker database before starting.

For each DB2 broker database in your broker domain:

1. Issue `db2cmd` to open a DB2 command window.
2. From the DB2 command window issue:
  - `mqsimigratebroker db2 <broker_database> <database_user_name> <password>`

For each Microsoft SQL Server broker database in your broker domain:

1. Open a new MS-DOS command window.
2. From the MS-DOS command window issue:
  - `mqsimigratebroker msql <broker_database> <database_user_name> <password>`

**Note:** You **must** ensure that the `mqsimigratebroker` command is run using the same database userid and password that were used on the `mqsicreatebroker` command (`-u DataSourceUserId -p DataSourcePassword`).

### Upgrading your Configuration Manager

You must delete and recreate the Configuration Manager in order to pick up new or changed IBM supplied nodes and new IBM supplied message sets, and to incorporate modifications to tables. The new nodes are **RouteToLabel**, **Label** and **FlowOrder**. The new message sets are for the MQSeries link for R/3 SAP bad message header and for the MQSeries PCF message headers.

1. Delete and recreate the Configuration Manager.
  - If you want to preserve your existing configuration data:*
    - a. Run `mqsdeleteconfigmgr` (without `-n -m`)
    - b. Issue `db2cmd` to open a DB2 command window.
    - c. From the DB2 command window issue:
      - `mqsimigrateconfigmgr <configuration_repository> <user_name> <password>`
    - d. Run `mqsicreateconfigmgr` (with appropriate parameters)
    - e. Before you start the Configuration Manager:
      - Run `mqsimrmimpexp -i` on filename `mqcfh.mrp` in the `mrm` directory
      - Run `mqsimrmimpexp -i` on filename `smqbmh.mrp` in the `mrm` directory

*If you do not want to preserve your existing configuration data:*

- a. Run `mqsideleteconfigmgr -n -m`
  - b. Run `mqsicreateconfigmgr` (with appropriate parameters)
  - c. (No `mqsimigrateconfigmgr` or `mqsimrmimpexp` steps are needed)
2. Start the Configuration Manager and a Control Center.
  3. Ensure that the instructions in “Upgrading your brokers” on page 24 are carried out.
  4. Perform a complete deploy from the Control Center Assignments view to all brokers.

Failure to follow these instructions may result in either message BIP1205 or BIP1502 appearing when you deploy, with CMRMUUUID cited as the column name in error.

**Note:** You **must** ensure that the `mqsimigrateconfigmgr` command is run using the same database userid and password that were used on the `mqsicreateconfigmgr` command (`-u DataBaseUserId -p DataBasePassword`).

## MQSeries Publish/Subscribe

If you install MQSeries Integrator Version 2.0.1 with the intention of operating both MQSeries Integrator brokers and MQSeries Publish/Subscribe brokers connected in a heterogeneous broker network, you must ensure that you have the required level of MQSeries for your MQSeries Publish/Subscribe systems:

- MQSeries Version 5.0. For MQSeries Publish/Subscribe brokers, you must install CSD7.

**Note:** You cannot run MQSeries Integrator brokers on MQSeries Version 5.0 at any service level. This option is only valid for MQSeries Publish/Subscribe brokers.

- MQSeries Version 5.1. For MQSeries Publish/Subscribe brokers, you must install CSD4 on Windows NT, or CSD1 on other platforms (AIX, Sun Solaris, and so on).

If you do not upgrade MQSeries to the specified levels on the systems on which you intend to run MQSeries Publish/Subscribe brokers, it is possible that some publications sent by MQSeries Integrator brokers will be wrongly put to the dead-letter queue (DLQ) by an MQSeries Publish/Subscribe neighbor broker.

You are not required to take any other specific action at installation time. Your current MQSeries Publish/Subscribe installation will not be affected by

## Migration

MQSeries Integrator Version 2.0.1 installation. However, you cannot create an MQSeries Integrator broker that uses a queue manager already in use by a MQSeries Publish/Subscribe broker.

For guidance on planning subsequent migration or integration of brokers, see *MQSeries Integrator Introduction and Planning*. For details of actions to take for integration and migration, see the *MQSeries Integrator Administration Guide*.

---

## Chapter 3. Installing MQSeries Integrator

This chapter tells you how to install MQSeries Integrator for Windows NT.

It covers the following:

- “Delivery media”.
- “Preparing for installation” on page 28.
- “Installation procedure” on page 29.
- “System update information” on page 33.
- “Setting up for LAN installation” on page 33.
- “What to do if something goes wrong during installation” on page 34.

---

### Delivery media

The MQSeries Integrator for Windows NT Version 2.0.1 package includes the following:

- MQSeries Integrator for Windows NT Version 2.0.1 on CD. This includes DB2 Version 6.1.
- MQSeries Integrator for Windows NT Version 2.0.1 supplemental CD. This contains:
  - The IBM DB2 Universal Database Client for Windows NT.  
The Administration client and the Run-time client are included in all available national languages.
  - MQSeries for Windows NT Version 5.1 CSD4.  
This CSD is provided to enable you to upgrade an existing installation of MQSeries for Windows NT Version 5.1.
  - Additional product service, if required.  
Up-to-date details of the service levels required are included in the MQSeries Integrator Version 2.0.1 Readme.txt file on the primary product CD.
- MQSeries for Windows NT Version 5.1 on CD.  
This CD has service level CSD4 already included. If you install the MQSeries product from this CD you do not need to apply any additional maintenance. This product is provided in all available national languages.
- MQSeries Clients on CD.  
Clients for all platforms in all available national languages are included.
- The following hardcopy installation documentation:
  - This book.
  - *MQSeries for Windows NT V5.1 Quick Beginnings*.

## Delivery media

For details about these products, and their use with MQSeries Integrator, see “Software requirements” on page 7.

---

## Preparing for installation

This section informs you of the steps you must take before you install and use MQSeries Integrator for Windows NT.

### Before you start

Before starting to install MQSeries Integrator for Windows NT, review `Readme.txt`, which you can find in the root directory of the CD. This contains the latest up-to-date information available.

You can also find the full product documentation in pdf format on the supplemental CD-ROM.

When you install MQSeries Integrator, you must be logged on with a user ID that is a member of the Windows NT **Administrators** group in the local account security domain. If you do not have this authority, you will not be able to complete installation. However, you are recommended not to use a logon ID of greater than eight characters. If your logon ID is greater than eight characters, you will receive message BIP8638W in a message dialog early in the installation process, indicating that this might cause a problem with database creation for MQSeries Integrator components.

If you are using a Windows NT trusted security domain, you are recommended to ensure that the global groups required by MQSeries Integrator are defined before this installation, if necessary. For more details, see “Using Windows NT primary or trusted security domains” on page 18.

You are recommended to exit all Windows NT programs before starting the installation procedure.

The most common way of installing this product is direct from the CD. You can also set up for installation from a shared LAN drive. This is described in “Setting up for LAN installation” on page 33.

You are guided through the installation process, and are prompted for any information required for completion. Online help is available from every window to help you complete the installation steps.

### Prerequisites

MQSeries Integrator for Windows NT installation checks for the presence of all the prerequisite software required by your installation choices. If any prerequisite is not found, you are presented with a window that details what is required. You are recommended to terminate installation at this point by

clicking **Cancel**, and to install (or upgrade) the products listed. You can, if you prefer, continue with installation by clicking **Next**. If you continue, you must install or upgrade the products specified on the window before you use MQSeries Integrator.

The prerequisites depend on the components you install. You are advised to check the full details of prerequisites for each component given in “Software requirements” on page 7.

---

### Installation procedure

The product takes between *5* and *30* minutes to install, depending on the options selected, the speed of your processor, and whether DB2 is installed during this process.

## Installation procedure

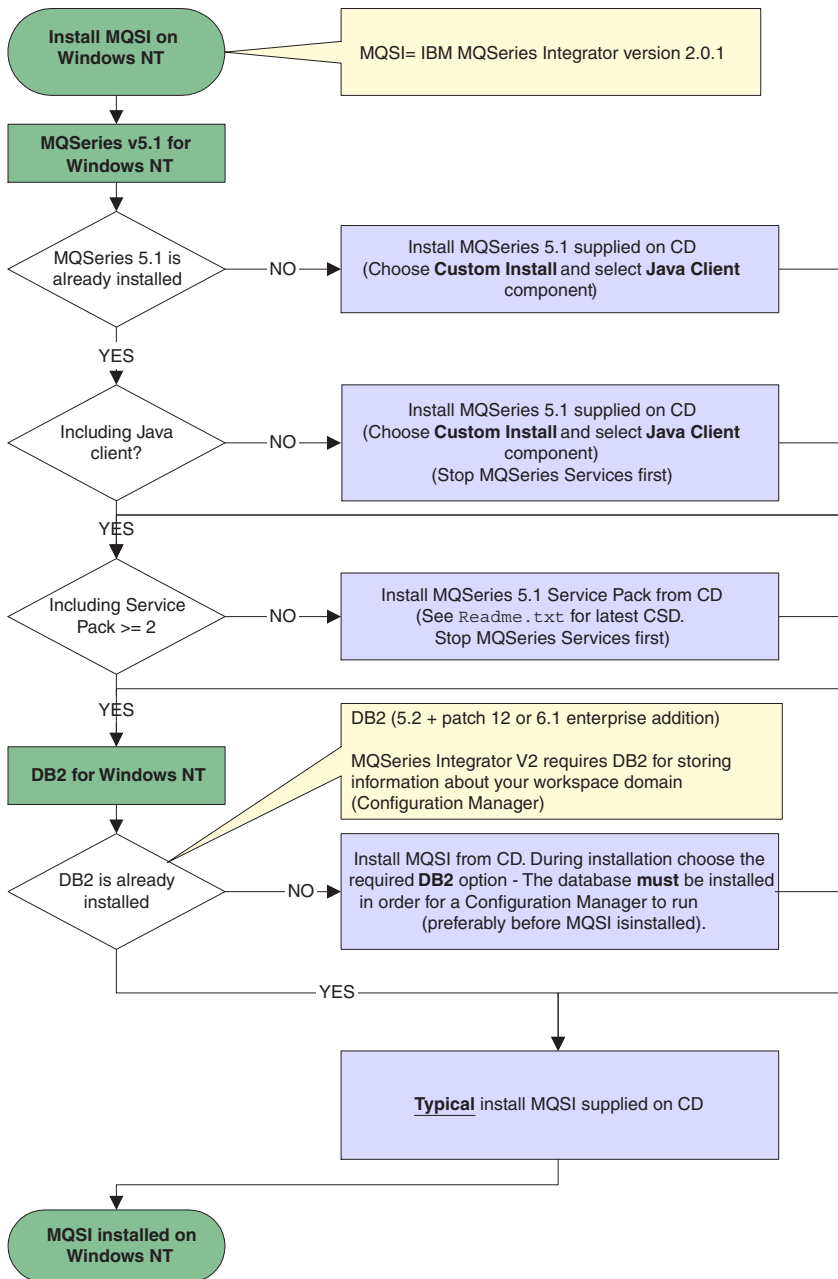


Figure 1. MQSeries Integrator for Windows NT installation procedure

The installation program leads you through a number of windows in which you choose what you want to install.



Insert the MQSeries Integrator for Windows NT CD into the CD-ROM drive. If autorun is enabled, the installation process starts. If it is not, double-click Setup.exe in the root folder on the CD to start the process.

Follow the actions described in the windows that are presented to you. In particular, note the following:

- If you have selected the broker or Configuration Manager component to install, the installation program requires a database to be present.
  - If a suitable database is not available, you are presented with the window *DB2 information*. This states that DB2 Version 6.1 will be installed for you when MQSeries Integrator installation has completed. This window also states the restrictions of use for this version of the DB2 product. You must accept these restrictions by clicking **Yes**. If you cannot accept these conditions, click **No**.

If you accept the conditions, you are presented with a series of extra windows that ask you to provide a user ID and password for the DB2 administrator. DB2 imposes a limit of eight characters for user IDs and passwords. If you do not accept the default user ID (db2admin), you must replace it with a valid user ID of eight characters or less. You must enter a password for the user ID you have specified. If this user ID does not exist, message BIP8639W is displayed asking you to confirm that you want this user ID to be created. Full guidelines on choosing a user ID and password are in the help available from this window.

If you are installing on a backup domain controller, MQSeries Integrator installation puts up a warning dialog with instructions for additional steps you must take to allow DB2 to install cleanly. It recommends that you terminate installation at this point, take the necessary action, then restart installation. If you choose to continue, DB2 installation will fail. You must take the necessary action after installation, and install DB2 later.

If you do not accept the DB2 license conditions, MQSeries Integrator installation continues but DB2 is not installed. You must install suitable database support before you can use MQSeries Integrator.

After installation, DB2 services use and retain the user ID and password you have specified here (services DB2DAS00, DB2 Governor, and DB2 JDBC Applet Server). Therefore if you later change the password for this ID, you must change it in these services. You can do this by accessing the services from the Windows NT Control Panel.

If the checks on this user ID and password fail, you can continue with MQSeries Integrator installation, but DB2 installation might fail. If so, correct the error (reported in message BIP8632W) and install DB2 by invoking Setup.exe from the DB2 directory on the MQSeries Integrator CD.

## Installation procedure

- If suitable database support is available, but is not at the required level, you can continue with MQSeries Integrator installation, but you must upgrade your database support before you can use MQSeries Integrator.
- The window *Prerequisite information* is only displayed if your system does not meet the prerequisite requirements. If you see this window, you must note and implement all the actions displayed. You are recommended to click **Cancel** to terminate the installation program and implement the changes, then restart MQSeries Integrator installation. You can continue with installation (click **Next**), but you cannot use MQSeries Integrator until you have made the changes specified in this window.
- You are presented with the window *Choose NEON rules and formats database*. This window asks you to choose the database you want to use with NEONFormatter and NEONRules data.
  - If you already have MQSeries Integrator Version 1 installed, you must either:
    - Choose the same database as the one you already use with Version 1 to ensure you can continue to use your data. OR
    - Choose DB2 if you intend to migrate your existing formats to DB2.
  - If you do not have Version 1 installed, choose the database you expect to use for Version 1 format data.
  - If you do not expect to use these formats, or are unsure, choose DB2.

**Note:** If the installation process is interrupted, you should run the process again; running the uninstallation process might not fix the problem.

When all processing has been completed, the *Setup complete* window is displayed. This allows you to choose:

- To review the release notes (Readme.txt). This option is selected by default.
- To install the plug-in Java 1.1.2\_001. This is required by the MQSeries Integrator Tour. You must accept the license agreement for this plug-in. You can select the directory into which this plug-in is installed.
- To install the plug-in Lotus Media. This is required by the MQSeries Integrator Tour.

For details of how to complete the post-installation steps that are required, see “Chapter 4. Getting Started with MQSeries Integrator” on page 41.

Click the **Finish** button. If you have chosen to view Readme.txt, this is displayed now in a *Notepad* session.

You are now asked to register your use of the product.

If DB2 has been installed, or if one (or more) of the reshippable Microsoft Windows NT DLLs shipped with MQSeries Integrator has been updated, you are presented with the *Restart computer* window. You can choose to restart your computer immediately, or later (the default). However, you cannot complete configuration until you have restarted your system and initialized the database (see “Chapter 4. Getting Started with MQSeries Integrator” on page 41 for details).

---

### System update information

The installation program updates some system resources:

- The selected components of MQSeries Integrator for Windows NT are installed.
- The product has created and populated a local directory structure, shown in “Directory structure” on page 89.
- Environment variables have been created, or modified, as shown in “Environment variables” on page 90.
- The Windows NT registry has been updated to reflect your installation. A summary of the changes is shown in “Windows NT registry” on page 91.
- The principals for MQSeries Integrator security, listed in “Principals” on page 96, have been created.

For details of how to complete the post-installation steps that are required, see “Chapter 4. Getting Started with MQSeries Integrator” on page 41.

---

### Setting up for LAN installation

You can choose one of the following two methods to make the MQSeries Integrator installation files accessible on a LAN server:

- You can make the MQSeries Integrator CD-ROM drive shareable.
- You can copy the product files from the CD-ROM to the server by following these steps:
  1. Create a folder on the LAN server to store the installation files. For example:

```
md m:\instmqsi
```
  2. Load the MQSeries Integrator CD. If you have autorun enabled, the initial window appears; cancel this window.
  3. Copy the entire CD to the new folder. For example:

```
xcopy e:\*.* m:\instmqsi /e
```

This copies the complete contents of the CD to the specified location on the server.

## LAN installation

4. Give all licensed users access to the folder that now contains the CD-ROM image (in this example, the `m:` drive). You must also add the users to the appropriate Windows NT groups (for example, the **Administrators** group).
5. Make the drive shareable using Windows NT Explorer.
6. From a command prompt on the target machine, connect to the appropriate drive and folder using the net use command:  

```
net use devicename \\servername\netname
```

For example:

```
net use x: \\mq sint\instmqsi
```

where `x:` is the required mapped drive on the target machine.

**Note:** If your shared installation directory contains spaces (for example, `mqsi install`), you must enclose it in quotes on the net use command.

**Note:** If your server is protected, you might have to enter a user ID and password on the net use command. See the Windows NT online help for more information about net use.

Alternatively, you can use the Windows NT Explorer or some other method to map the shared resource to a drive letter.

7. Change to the installation directory (in this example, `x:`). Type **Setup** and press Enter.
8. Follow the installation prompts.

---

## What to do if something goes wrong during installation

If you encounter any problems during installation, you are advised to check the following:

- Review the `Readme.txt` file supplied on the CD. This has the most up-to-date information available for product installation and operation. There might be last minute changes to the installation process that you must follow. You might also find additional information on the MQSeries Web site (the address is given in “MQSeries information available on the Internet” on page xiii).
- If a message is displayed with the MQSeries Integrator prefix of BIP, check the *MQSeries Integrator Messages* book to determine the cause of the error and the action you need to take to correct it.
- Review the MQSeries Integrator installation log, `mqsi2.1.log`. You can find this file in the directory `X:\winnt` where `X:` is the operating system drive.

This file is primarily intended for use by IBM service personnel, but you might find additional information that helps you identify the current problem.

- If DB2 has also been installed by the installation program, check the DB2 logs `mqs12db2.iss` and `mqs12db2.log` in the `\winnt` directory. These files are primarily intended for use by IBM service personnel, but you might find additional information that helps you identify the current problem.
- If a message is displayed with the DB2 prefix of DBI, check the *DB2 Troubleshooting Guide* and the *DB2 Message Reference* to determine the cause of the error and the action you need to take to correct it.
- If any other error is displayed, check the Windows NT online help.

When you have identified and corrected the error, you can run the installation program again. If this does not work, you are advised to follow the steps for manual uninstallation (described in “Manually uninstalling MQSeries Integrator” on page 102) to ensure that your system is in a consistent state before you retry.

If you are unable to resolve the problems you have, after checking the possible sources of error listed above, you must contact your IBM Support Center. See “Contacting your IBM Support Center” on page 106 for further information.

---

### Silent installation

MQSeries Integrator Version 2.0.1 supports silent installation, where no user interaction is expected. This method of installation is typically used where you have to install many identical copies of the same software across many identical machines. These machines can be remote to the installer.

To perform a silent installation you must invoke the `setup.exe` file for MQSeries Integrator from a command line. To do this enter the following command:

```
<location> setup <parameters>
```

Where `<location>` is the location of the install image and `<parameters>` are any parameters you want to specify for this installation. The available parameters are listed in Table 5 on page 36.

### Recording a response file

You can create response files manually, however this is not recommended. The preferred method is to RECORD a response file. You can do this by invoking setup with the `-r` flag. For example:

```
setup -noinst -r -f1x:\mypath\my.iss
```

## Silent installation

If you do not have MQSI installed, you will step through all the input screens you would see during a normal installation.

*Table 5. Parameters for silent installation and response files*

Parameter	Explanation
-s	Indicates that this setup is silent.
-nomsgbox	Indicates that message-box output for warnings should be ignored. This must be placed before the -s flag.
-f1	Indicates the start of a filepath and identifies the response file to be used with this installation. No spaces are allowed between the -f1 flag and the filepath. For example: -f1x:\mypath\my.iss.
-f2	Indicates the start of a filepath for the setup log produced by InstallShield for this installation. This file is created in addition to the mqsi2.log file that can be found in the Windows directory.
-r	Indicates that this setup is to record a response file.
-noinst	Indicates that this install is only to produce a response file and not to perform a real installation (so nothing will be written to the registry and no files will be copied).

Running setup this way shows the normal user input screens that you would encounter during a normal installation and records your responses.

If you have a previous installation of MQSI, you **must** do the following before you run setup.exe:

- Stop any brokers
- Stop the configuration manager and User Name Server
- Close the Control Center

### Notes:

1. You may see a warning message telling you that a previous install has been detected and that reinstall may overwrite the configuration files. Provided that you have invoked the process with the -noinst option, it is safe to ignore this warning.
2. You may not be presented with the option to install NEON support because the option was included with the original installation.

## Running the silent installation

Below is an example of a command for a silent install. The parameters in square brackets are optional.

```
setup [-nomsgbox] -s -f1x:\mypath\my.iss [-f2y:\mypath\mylog.log]
```

## Response File

Silent installation requires a response file. This is a text file containing coded responses for the dialogs that would be shown in a normal manual installation. An example of an MQSeries Integrator 2.0.1 response file is shown below:

```

[InstallShield Silent]
Version=v5.00.000
File=Response File
[File Transfer]
OverwriteReadOnly=NoToAll
[DlgOrder]
Dlg0=ID_DLG_WELCOME-0
Count=8
Dlg1=MbLicense-0
Dlg2=MbAskDestPath-0
Dlg3=MbSetupType-0
Dlg4=MbSelectFolder-0
Dlg5=MbChooseNeonFlavor-1
Dlg6=MbStartCopy-0
Dlg7=MbFinish-0
[ID_DLG_WELCOME-0]
Result=1
[MbLicense-0]
Result=1
[MbAskDestPath-0]
szDir=j:\mqsi2
Result=1
[MbSetupType-0]
Result=301
[MbSelectFolder-0]
szFolder=IBM MQSeries Integrator 2.0
Result=1
[MbChooseNeonFlavor-1]
nNeonFlavor=32501
Result=32500
[MbStartCopy-0]
Result=1
[MbFinish-0]
Result=1
bOpt1=0
bOpt2=0
bOpt3=0
bOpt4=0
bOpt5=0
bOpt6=0

```

This example was produced by performing a full install where DB2 was already installed. In this example the destination directory was changed from the default (to j:\mqsi2), DB2 has been chosen as the Neon database type (nNeonFlavor = 32501) and the readme file will not be launched (bOpt1 = 0).

### Detecting Errors from a silent install

If a normal installation fails then the installer will see a message box reporting the failure. For silent install no message box will be produced. You can verify that installation was successful by checking either the log you specified with the -f2 flag or, if this parameter was not specified, you can check the setup.log. The log for a successful installation should look like the example below:

## Silent installation

```
[InstallShield Silent]
Version=v5.00.000
File=Log File
[Application]
Name=IBM MQSeries Integrator 2.0.1
Version=2
Company=IBM
Lang=0009
[ResponseResult]
ResultCode=0
```

The 'ResultCode=0' line indicates that as far as InstallShield is concerned Setup has worked, however this does not necessarily mean that the silent install of MQSeries Integrator V2.0.1 was successful. Non-zero result code values that InstallShield may return are shown in Table 6.

*Table 6. Result codes for silent installation.*

Result code	Explanation
-1	General error
-2	Invalid mode
-3	Required data not found in the Setup.iss file
-4	Not enough memory available
-5	File does not exist
-6	Cannot write to the response file
-7	Unable to write to the log file
-8	Invalid path to the InstallShield Silent response file
-9	Not a valid list type (string or number)
-10	Data type is invalid
-11	Unknown error during setup
-12	Dialog boxes are out of order
-51	Cannot create the specified folder
-52	Cannot access the specified file or folder
-53	Invalid option selected

To verify that your Silent Installation was successful you need to check the log specified on the -f2 flag on the setup command to check that the ResultCode=0.



If a log was not specified:

- Check the setup log in the Windows directory
- Check the Mqsi2.log in the Windows directory normally C:\WINNT.

Whilst this is primarily for IBM Service use in debugging installation problems, it also contains success or failure details for that install. For a successful install the last lines of this log will look like the example below:

```
[date/time] ---> Function: ExitSetup()
[date/time] Done: Setup successful - Return Code 0
[date/time] <--- Function: ExitSetup()
[date/time] Log stopped.
Log Flush = 0.
```

For an unsuccessful install, the last lines could look like the example below:

```
[07-13 10:22:25] ---> Function: DoMessageBox()
[07-13 10:22:25]      !!! Silent Install - Message Box Intercept !!!
[07-13 10:22:25]      Message was: Unable to create target directory
[07-13 10:22:25]      Severity was: SEVERE, ErrorNo was: -30008
[07-13 10:22:25]      ABORT was set - ABORT was set
[07-13 10:04:51] ---> Function: ExitSetup()
[07-13 10:04:51] Done: Setup unsuccessful - Return Code -30008
[07-13 10:04:51] <--- Function: ExitSetup()
[07-13 10:04:51] Log stopped.
Log Flush = 0.
```

In this example the target directory (szDir=j:\mqsi2 in the example response file) was not valid on this machine. For a normal non-silent install this would not be a problem as the user would have had to have chosen a good target directory for the install to proceed, but for a silent install the response file value is fixed.

The non-zero return codes that you might find in MQSI2.log are listed in Table 7.

*Table 7. Non-zero return codes in the MQSI2.log*

Return code	Explanation
-30000	User cancelled the install (should never be seen on silent install)
-30001	There is another install/uninstall running
-30002	Userid attempting to perform the install does not have Administrator authority
-30003	Install is being attempted on a non Win32 system
-30004	Attempting to install an MQSeries Integrator CSD for the wrong version/release
-30005	Attempting to install a CSD when no base product installed
-30006	Attempting to install a CSD lower (or equal) to the currently installed CSD
-30007	Not enough disk space available to perform install
-30008	Unable to create the target directory

## Silent installation

*Table 7. Non-zero return codes in the MQSI2.log (continued)*

<b>Return code</b>	<b>Explanation</b>
-30009	The target directory found in the registry does not exist (CSD install)
-30010	Failure in creating backup directory (CSD Install)
-30011	Registry key creation failure
-30012	Registry key value setting failure
-30013	Failure in creating a program folder entry
-30014	InstallShield error
-30015	Failure in backing up a file (CSD Install)
-30016	Invalid operating system detected

---

## Chapter 4. Getting Started with MQSeries Integrator

This chapter helps you get started with MQSeries Integrator: it takes you through the set up of a very simple configuration, and shows how you can create a simple scenario to confirm that the product has been installed successfully.

The chapter has two sections:

- “Configuring a simple broker domain”.
- “Verifying your installation” on page 63.

---

### Configuring a simple broker domain

This section takes you through the steps you must complete to set up the minimum resources required in the broker domain. If you want to start with, or progress to, a more complex configuration, refer to “Chapter 5. A more complex configuration” on page 79.

There are six steps to take:

- Step 1. “Designing a simple configuration” on page 44.
- Step 2. “Setting up user IDs and groups” on page 46.
- Step 3. “Creating and connecting to the databases” on page 48.
- Step 4. “Setting up database authorizations” on page 50.
- Step 5. “Configuring your broker domain” on page 52.
- Step 6. “Starting your broker domain” on page 59.

Figure 2 on page 42 provides a schematic of these tasks: you might find it helpful to review this figure before you start the tasks detailed in this section, and to use it to check off the tasks as you complete them. You must complete the sub-tasks defined to complete each of these steps.

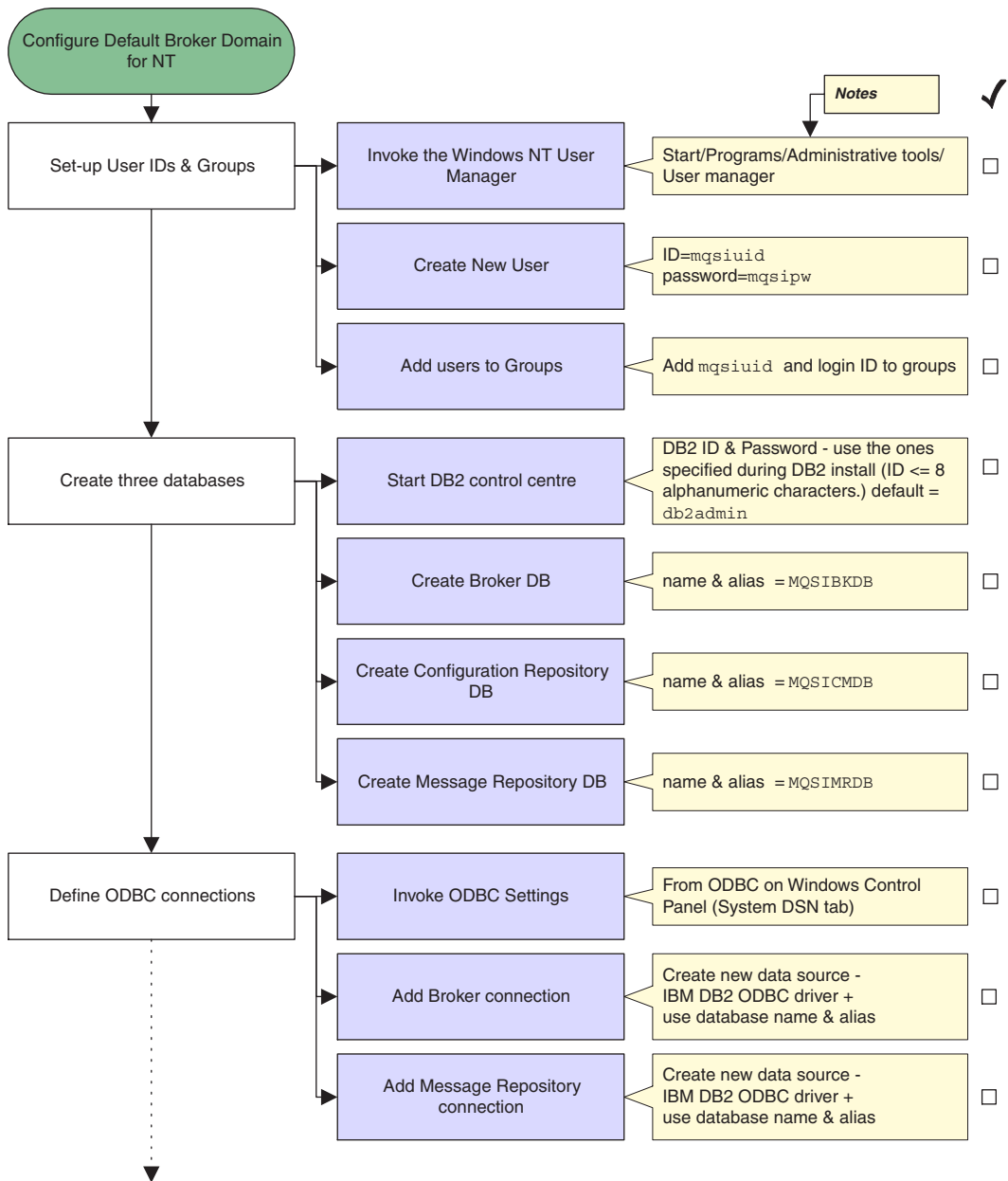
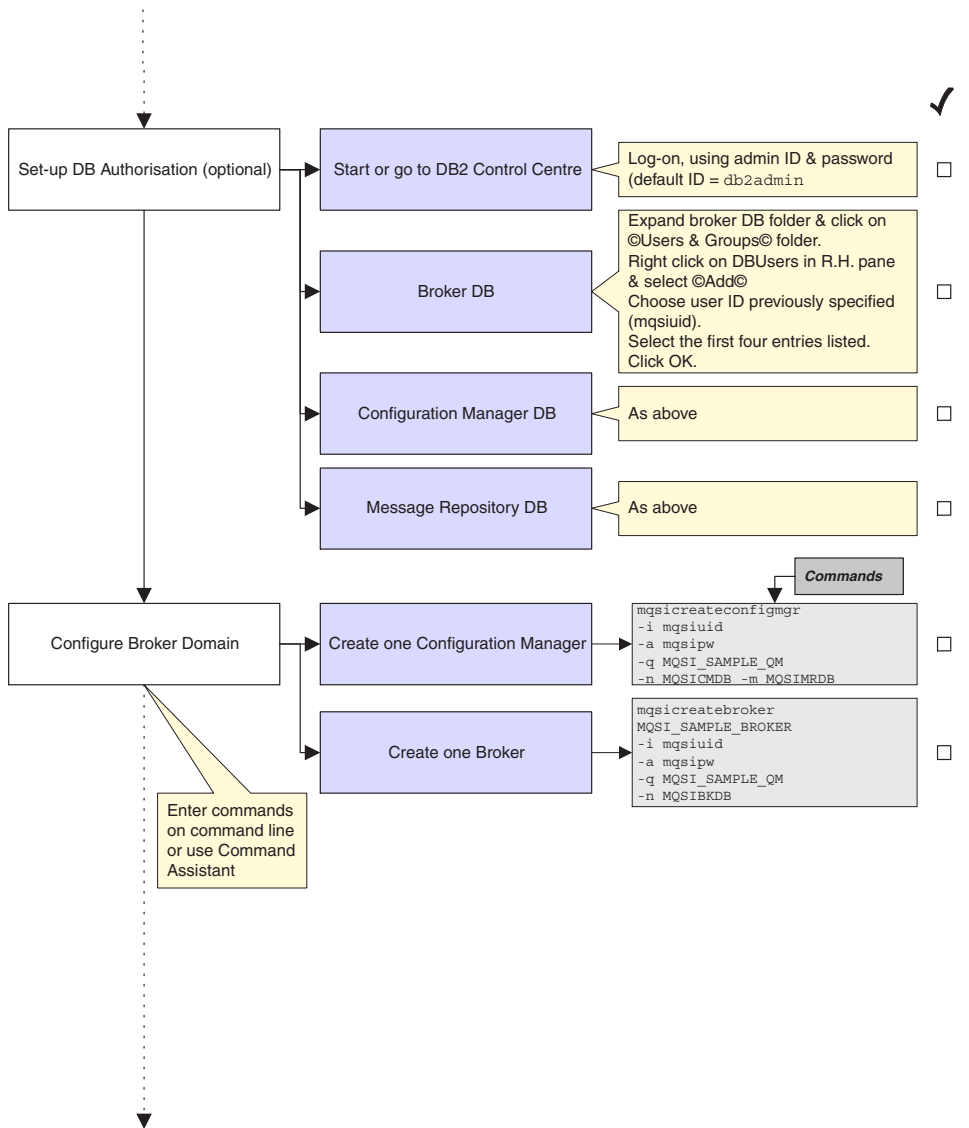
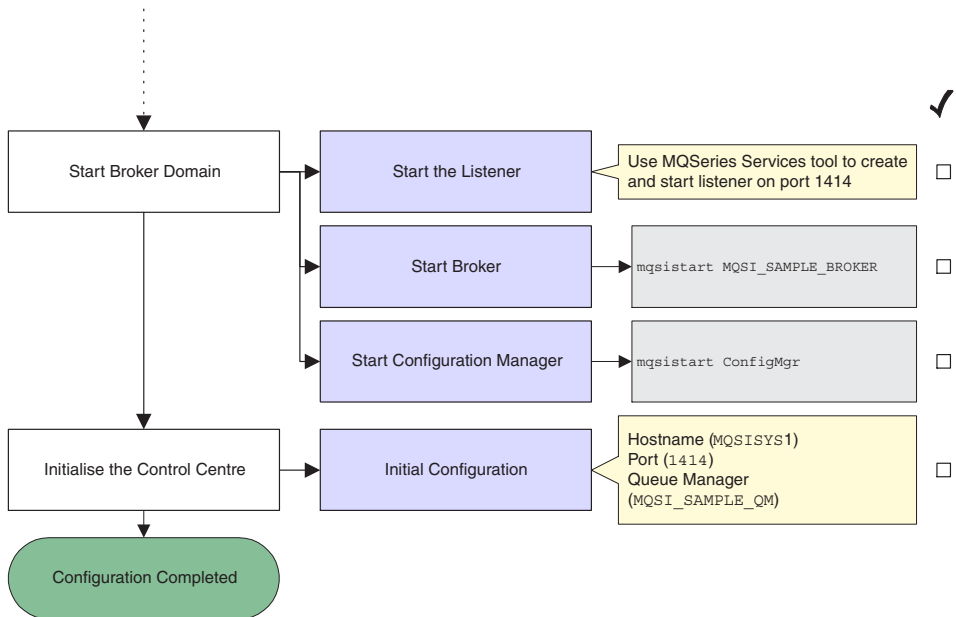


Figure 2. Configuring a simple broker domain





Most of these steps make use of commands supplied by MQSeries Integrator. Some of these commands can be invoked using the MQSeries Integrator Command Assistant; all of them can be invoked from the command line. The Command Assistant screens are illustrated, and the commands are given in full. You can choose which method you want to use to issue these commands.

The Command Assistant and the configuration commands are described in detail in the *MQSeries Integrator Administration Guide*, which provides further reference and guidance material, and describes the actions you must take if you experience any errors in completing the tasks illustrated here.

## Designing a simple configuration

Before you start to define any resources, review the assumptions made about the simple configuration that is created. If you want to understand more about MQSeries Integrator configuration in general, refer to *MQSeries Integrator Introduction and Planning*.

The assumptions for this configuration cover resource names, user IDs, and so on. If you want to override any of the assumptions, make a note of changes you want to make and apply those changes as you complete the tasks illustrated. For example, the names used for the broker and its queue manager are for illustration only. You are recommended to follow any existing naming conventions you have for MQSeries (or any other) resources. See *MQSeries Integrator Introduction and Planning* for more information about defining a naming convention.

The assumptions made in this chapter are:

- You have completed a **Full** installation. All components are therefore installed, and can be created, on a single machine.
- You have installed the product on a system with a TCP/IP hostname of MQSISYS1. You **must** replace this name wherever it is used with the hostname of your system, if it is different.
- The default MQSeries port 1414 is available. You **must** replace this port with a different port wherever it is used, if this default port is not available.
- The local system MQSISYS1 defines the security domain relevant to this configuration (that is, all users and groups are defined in the local account security domain).

**Note:** This illustrates the simplest security scenario. You can find more comprehensive security information in *MQSeries Integrator Introduction and Planning*, and more complex scenarios illustrated in the *MQSeries Integrator Administration Guide*.

- You are logged on with the same user ID that you used to install MQSeries Integrator. If you are not, you must ensure that your current logon ID is a member of the Windows NT **Administrator** group.
- You have decided to create a new user ID, the 'service userid' (*mqsuid*), to use as the user ID under which the MQSeries Integrator services (the Configuration Manager and the broker) will run, and as the user ID under which all MQSeries Integrator databases are accessed.

In most cases, you are unable to change these user IDs once your configuration has been set up, so you are advised to check where you use them very carefully.

- The configuration includes one broker and the Configuration Manager, that share a single queue manager.
- A set of sample names and other default values are used for MQSeries Integrator, MQSeries, and database resources. You can use the sample names and defaults exactly as they are shown, or you can decide to use your own names, to follow the naming conventions you have in place. If you choose to use your own names, you **must** change the names and default values to match your configuration, whenever they are used in the tasks illustrated.
- You are using DB2 for all database requirements. DB2 installation has been done as part of your installation procedure, and you have since restarted your system and verified your DB2 installation.

**Note:** You can create the broker database in SQL Server, Oracle or Sybase. These options are not illustrated in this configuration. If you are using one of these databases, refer to the *MQSeries Integrator Administration Guide*.

## A simple configuration

Once you have verified your installation, and understood and implemented the basic principles of operating your broker domain, you are very likely to need a more complex MQSeries network for your broker domain. Your brokers and the Configuration Manager are likely to be located on different physical machines, and you are likely to include a User Name Server in your broker domain. “Chapter 5. A more complex configuration” on page 79 introduces a more complex broker domain that illustrates some of the requirements of a more extensive installation. For more detailed guidance and instructions, refer to the *MQSeries Integrator Administration Guide*.

### Setting up user IDs and groups

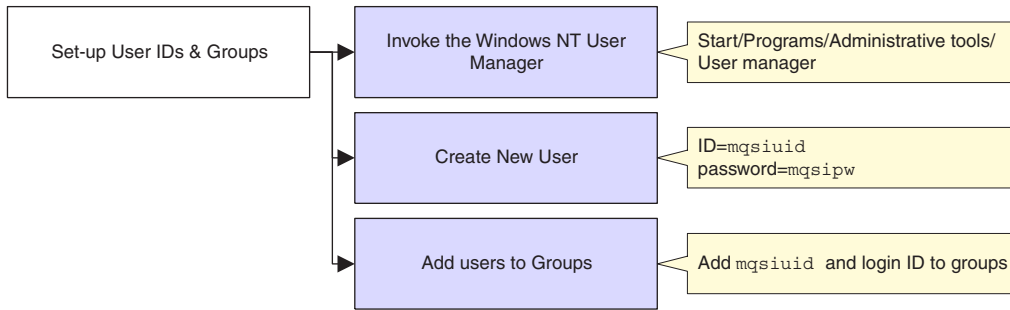


Figure 3. Setting up user IDs and groups on Windows NT

If you are creating a new user ID for MQSeries Integrator, you must invoke the Windows NT User Manager to create it and assign it to all the MQSeries Integrator groups. If you have chosen to use an existing user ID, you must invoke the User Manager to assign this ID to the MQSeries Integrator groups. These tasks are described below.

If your current logon user ID is not the ID you used to complete installation, you must ensure that it is member of the Windows NT group **Administrators**. This is independent of the user ID you define in the MQSeries Integrator groups to run MQSeries Integrator components. Membership of **Administrators** provides the level of security required by the MQSeries operations completed here.

You must be careful to specify the correct user IDs wherever they appear in this chapter. You can define and use the sample user ID defined here. However, if you decide to use a different (new or existing) user ID, make sure you use this other user ID wherever the sample user ID is shown in the following tasks.



You must take the following steps:

- Step 1. Invoke the Windows NT User Manager. You can access this program from the Windows NT Start menu (the default is *Start->Programs->Administrative Tools->User Manager*).
- Step 2. Create a new user ID.
  - a. Select the *User* menu and select *New user...*
  - b. Fill in the fields on the dialog presented to create the user ID `mqsuid`, with password `mqsipw`.  
Note that the maximum length for the `mqsuid` field is 12 characters.
- Step 3. Add the new user ID to the groups required for running the components (see Table 4 on page 19 for more details).
  - a. Click the **Groups** button on the *New User* dialog. This presents another dialog, *Group Memberships*.
  - b. Add the new user to the MQSeries Integrator group `mqbrkrs`, to the Windows NT group **Administrators**, and to the MQSeries group `mqm`.
  - c. Click **OK**. The User Manager returns you to the *New User* dialog.
  - d. Click **OK**.
- Step 4. Add your logon ID to the groups that authorize you to complete all the tasks in the Control Center.
  - a. Double click your current user ID. This presents the *User properties* dialog.
  - b. Click the **Groups** button on this dialog. This presents another dialog, *Group Memberships*.
  - c. Add the new user to `mqbrasgn`, `mqbrdevt`, `mqbrops`, and `mqrtpic`.
  - d. Click **OK**. The User Manager returns you to the *User Properties* dialog.
  - e. Click **OK**.
- Step 5. You can now close the User Manager.

## DB2 setup

### Creating and connecting to the databases

#### Note to users

On multi-way machines you need to bind the db2cli package to the configuration-manager database, by opening a DB2 Command Line Processor window and carrying out the following procedure:

1. Connect to the database name.
2. Issue the command `Bind c:\sqllib\bnd\@db2cli.lst`, blocking all grant public.
3. Connect and reset.

where `c:\` is the drive on which you installed DB2.

You must take the following steps:

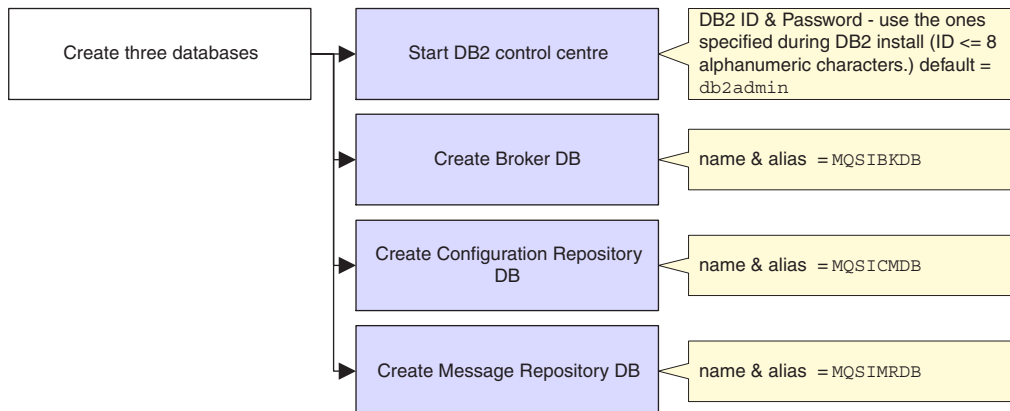


Figure 4. Creating and connecting to the databases

- Step 1.** Start the DB2 Control Center from the Start menu  
(*Start->Programs->DB2 for Windows NT->Control Center*).

You must enter a valid user ID and password on the Sign On dialog presented. Use the ID you specified when you installed DB2, the DB2 Administrator ID. If you accepted the default, the user ID is db2admin. If you changed this to another user ID, enter that user ID. Enter the password for the user ID you are using.

- Step 2.** Create the databases.

Expand the Object tree in the DB2 Control Center until you find Databases. Right-click Databases and select *Create Database using Smartguide*.

**Note:** DB2 database names are limited to eight characters.

You are recommended to create three databases, to hold three independent sets of tables:

a. The broker database

Enter the name and alias of your database. This chapter assumes you specify MQSIBKDB as both the name and the alias. If you decide to use another name, enter that name here and in all other steps in which this database is referenced.

b. The configuration repository

Enter the name and alias of your database. This chapter assumes you specify MQSICMDB as both the name and the alias. If you decide to use another name, enter that name here and in all other steps in which this database is referenced.

c. The message repository

Enter the name and alias of your database. This chapter assumes you specify MQSIMRDB as both the name and the alias. If you decide to use another name, enter that name here and in all other steps in which this database is referenced.

If you prefer, you can create a single database to hold all the tables required. Whatever scheme you choose, ensure you specify the correct name whenever you are asked to specify a database in subsequent commands.

For each database you create, you must enter a name and alias. You can let all remaining options take the default values. Click **Done**. A confirmation message indicating successful completion of the create command appears at the bottom of the window.

**Step 3.** Define the ODBC connections.

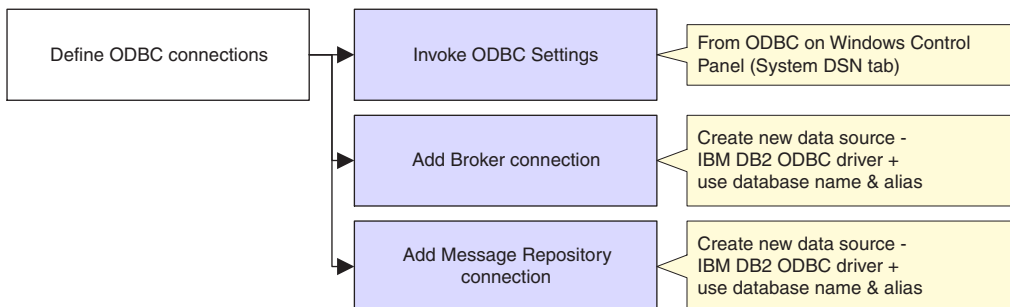


Figure 5. Defining the ODBC connections on Windows NT

## DB2 setup

- a. From the Windows NT Start menu, select *Start->Settings->Control Panel*.
- b. Within the Control Panel, double-click the ODBC icon (this will be labelled *ODBC* or *ODBC Datasources*).
- c. Click the *System DSN* tab.
- d. You must add an ODBC connection for the broker database and for the message repository. The configuration repository does not need an ODBC connection if it is created as a separate database. However, if you have created a database that will be used for either the message repository or the broker tables as well as the configuration repository, you must create an ODBC connection for that database.
  - 1) Click the Add button. The *Create New Data Source* window appears.
  - 2) Double-click **IBM DB2 ODBC DRIVER**.
  - 3) Choose the data source (database) name from the drop-down list.
  - 4) Click **OK**.

When you have completed these steps for the broker database and message repository database, click **OK**.

### Setting up database authorizations

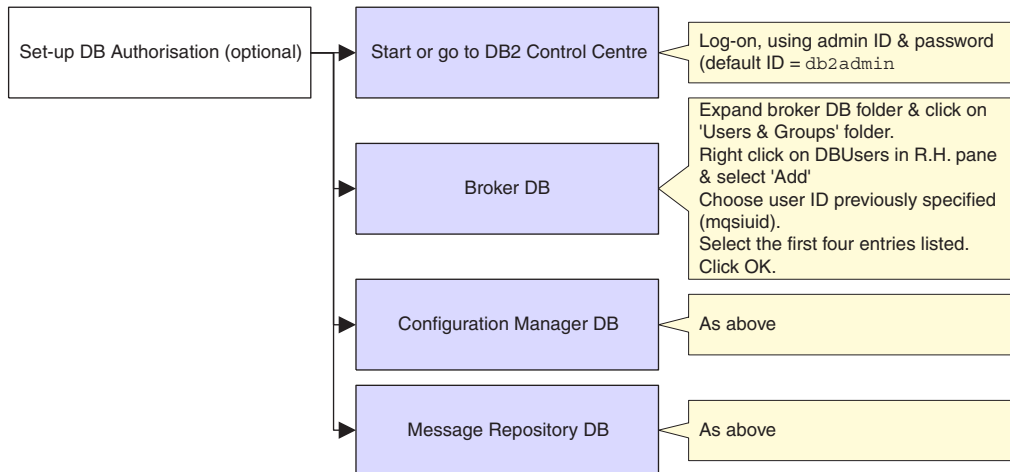


Figure 6. Setting up database authorizations

The next task is to authorize selected user IDs to access the databases you have created, to allow the MQSeries Integrator resources to operate

successfully. The steps you need to complete are shown below. If you need further guidance about any of these tasks, use the online help facility of the DB2 Control Center.

**Note:** You can omit this step if you choose to specify your DB2 administrator ID and password for the datasource and database IDs and passwords on the create broker and create Configuration Manager commands. This option is not illustrated in this book. See the *MQSeries Integrator Administration Guide* for further information.

**Step 1.** Start the DB2 Control Center, if it is not already active. Log on with the DB2 administrator user ID you used in “Creating and connecting to the databases” on page 48.

**Step 2.** Complete the following tasks for each database you created in “Creating and connecting to the databases” on page 48:

- a. Expand the object tree until you find the database.
- b. Expand the tree under this database and left-click the *User and Group Objects* folder. The *DB Users* and *DB Groups* folders are displayed in the right pane.
- c. Right-click the *DB Users* folder in the right pane and select *Add* from the pop-up menu. The Add User notebook opens.
- d. Select the user ID `mqsuid` (or the ID you are using for MQSeries Integrator database access) from the drop-down list.

Select the appropriate options from the choices in the box labelled *Choose the appropriate authorities to grant to the selected user* to all the databases you have created for MQSeries Integrator.

The ID you specify as the `ServiceUserID` on the create commands, `mqsuid` (or the user ID you are using in place of this sample ID), must have the following authority to all the databases you have created for MQSeries Integrator:

- Connect database.
  - Create tables.
  - Create packages.
  - Register functions to execute in database manager’s process.
- e. Click **OK**. The authority or authorities are granted. The dialog is closed.

**Step 3.** You can now close the DB2 Control Center.

## Configuring your broker domain

### Configuring your broker domain

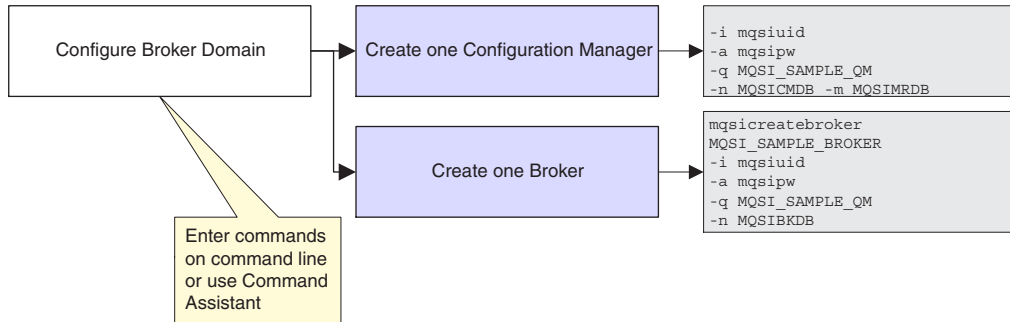


Figure 7. Configuring your broker domain

Now you are ready to define the components that make up the simple configuration.

You must work through the following steps:

**Step 1.** Create one Configuration Manager.

**Step 2.** Create one broker.

These steps are illustrated using the Command Assistant. When you enter values in the entry fields, you will see a display of the command that is generated in the lower part of the screen.

The MQSeries Integrator commands are also provided. Enter the commands at a command prompt window. You are recommended to enter the commands from the `\bin` subdirectory of the directory in which you installed MQSeries Integrator (the home directory). If you accepted the default, the home directory is

```
C:\Program Files\IBM MQSeries Integrator 2.0.1
```

#### **Step 1: creating the Configuration Manager**

Start the Command Assistant to create the Configuration Manager (select *Start->Programs->IBM MQSeries Integrator 2.0.1->Command Assistant->Create Configuration Manager*). You must complete the fields on two screens, shown in Figure 8 on page 53 and Figure 9 on page 53. You can then review and check the full command (compare it to the command shown) and click **Finish** on the third summary screen.

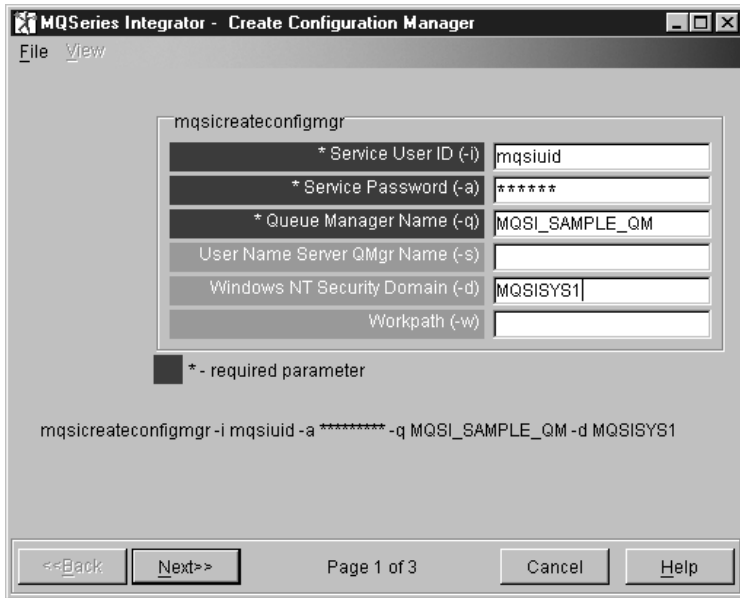


Figure 8. Create Configuration Manager: screen 1

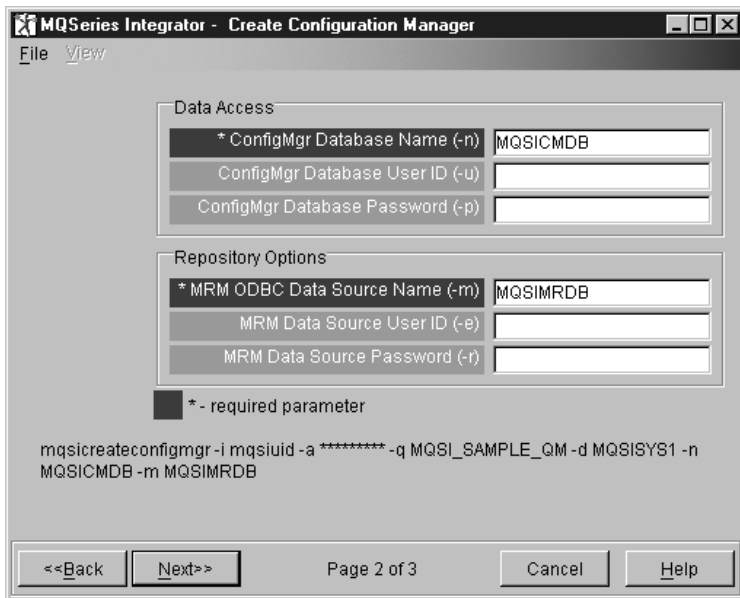


Figure 9. Create Configuration Manager: screen 2

## Configuring your broker domain

If you prefer, you can enter the following command:

```
mqsicreateconfigmgr -i mqsiuid -a mqsipw -q MQSI_SAMPLE_QM  
-d MQSISYS1 -n MQSICMDB -m MQSIMRDB
```

This identifies the queue manager (flag -q) that will host the Configuration Manager, the service user ID (flag -i) and password (flag -a) that the Configuration Manager will run under (as a Windows NT service), the security domain (flag -d) within which user authority is checked (in this case, the local account security domain defined by the hostname of this system), the database for the configuration repository (flag -n), and the datasource name for the message repository (flag -m).

**Note:** The queue manager will be created for you if it does not already exist.

The service user ID and password are also used as the user ID and password for both the configuration repository and the message repository. If you are using a different user ID and password for access to these repositories, you must specify these here (flags -u and -p for the configuration repository and flags -e and -r for the message repository).

When you type the password, it appears on the command line exactly as you type it. However, when you type it into the Command Assistant, and when it is stored in the Windows NT registry, it is displayed as asterisks for security reasons.

If you are using different names or values for any parameter on this command, you **must** replace the appropriate values with your own.

The command might take a short while to complete. The command generates the following expected responses, unless you are using the Command Assistant:

```
MQSeries queue manager created.  
Creating or replacing default objects for MQSI_SAMPLE_QM.  
Default objects statistics : 29 created. 0 replaced. 0 failed.  
Completing setup.  
Setup completed.  
MQSeries queue manager 'MQSI_SAMPLE_QM' started.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.
```



The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
security properties not found. using defaults.  
BIP8071I: Successful command completion.

**Note:** Only the message is shown in the log.

If the command detects any errors, or is unable to complete, it returns an error message on the command line, or in the Windows NT Event viewer (Application View) which includes the explanation and action to take in full. It is possible that the error has been caused by another component that MQSeries Integrator interacts with to complete this command (Windows NT, DB2, or MQSeries), so check for errors from these products, too. (The response security properties not found. using defaults shown above is not an error: you can ignore this informational message.)

**Note:** In some circumstances, you might see the following error message issued by the Java Runtime Environment (JRE).

```
address: [B@964f60
  security properties not found. using defaults.
  Can't get saved UUID state:java.io.FileNotFoundException:
  <mqsi_root>\bin\..\UUID
```

This error does not cause the **mqsicreateconfigmgr** command to fail, because the required file is created dynamically. You can therefore ignore this message.

On completion, the following have been done:

1. The Configuration Manager has been created, and the Windows NT service for it added to the Services (viewable from the Windows NT Control Panel). The service is called *IBM MQSeries Broker ConfigMgr*. It has a default start up status of manual: you can change this to automatic if you want to.
2. The queue manager MQSI\_SAMPLE\_QM has been created and started. You can check the existence and status of this queue manager using MQSeries Services from the Start menu (*Start->Programs->IBM MQSeries->MQSeries Services*).
3. The MQSeries resources required by the Configuration Manager have been defined on the queue manager. These resources are detailed in "Default MQSeries resources" on page 95.
4. The authorizations required by the Configuration Manager to access MQSeries resources have been set (the *setmqaut* messages seen in the responses to the command).
5. The database tables required by the configuration repository have been set up in the database MQSICMDB. These tables are listed in Table 10 on page 93.

## Configuring your broker domain

6. The database tables required by the message repository have been set up in the database MQSIMRDB. These tables are listed in Table 11 on page 94.
7. The Windows NT registry has been updated to record the Configuration Manager creation. Registry updates are described in “Windows NT registry” on page 91.

### Step 2: creating a broker

Start the Command Assistant to create the broker (select *Start->Programs->IBM MQSeries Integrator 2.0.1->Command Assistant->Create Broker*). You must complete the fields on two screens, shown in Figure 10 and Figure 11 on page 57. You can then review and check the full command (compare it to the command shown) and click **Finish** on the third summary screen.

MQSeries Integrator - Create Broker

File View

mqsicreatebroker

* Broker Name	MQSI_SAMPLE_BROKER
* Service User ID (-i)	mqsiuid
* Service Password (-a)	*****
* Queue Manager Name (-q)	MQSI_SAMPLE_QM
User Name Server QMgr Name (-s)	
Workpath (-w)	

\* - required parameter

```
mqsicreatebroker MQSI_SAMPLE_BROKER -i mqsiuid -a ***** -q MQSI_SAMPLE_QM
```

<<Back Next>> Page 1 of 3 Cancel Help

Figure 10. Create broker: screen 1

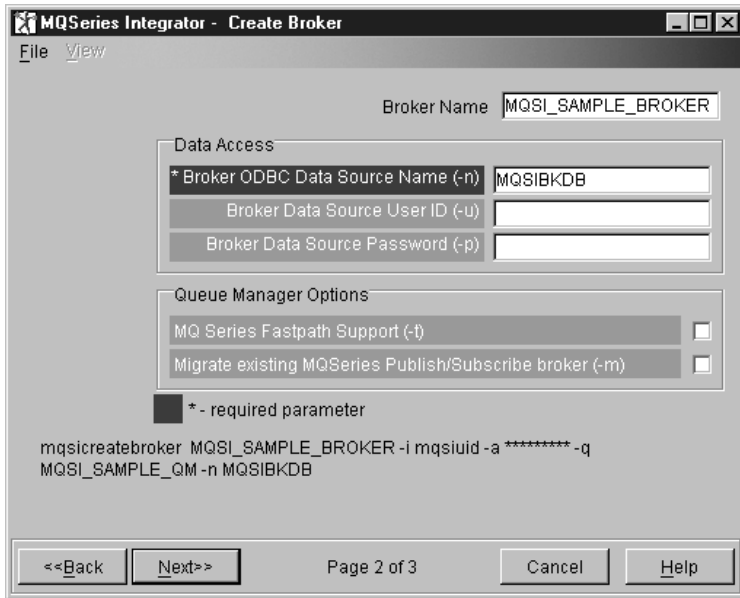


Figure 11. Create broker: screen 2

If you prefer, you can enter the following command:

```
mqsicreatebroker MQSI_SAMPLE_BROKER -i mqsiuid -a mqsipw
-q MQSI_SAMPLE_QM -n MQSIBKDB
```

This identifies the broker (MQSI\_SAMPLE\_BROKER), the broker's queue manager (flag -q) and specifies the database that has been created for the broker tables (flag -n). It also identifies the user ID (flag -i) and password (flag -a) that the broker will run under (as a Windows NT service).

**Note:** The queue manager has already been created by the **mqsicreateconfigmgr** command.

The service user ID and password are also used as the user ID and password for the broker database. If you want to use a different user ID (flag -u) and password (flag -p) for access to this database, you must specify these here and grant the user ID access to the database (described in "Setting up database authorizations" on page 50).

When you type the password on the command line, it appears on the screen exactly as you type it. However, when you type it into the Command Assistant, and when it is stored in the Windows NT registry, it is displayed as asterisks for security reasons.

## Configuring your broker domain

If you are using different names or values for any parameter on this command, you **must** replace the appropriate values with your own.

The command might take a short while to complete. The command generates the following expected responses, unless you are using the Command Assistant:

```
AMQ8110: MQSeries queue manager already exists.
MQSeries queue manager running.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
BIP8071I: Successful command completion.
```

**Note:** Only the message is shown in the log.

If the command detects any errors, or is unable to complete, it returns an error message on the command line, or in the Windows NT Event viewer (Application View) which includes the explanation and action in full. It is possible that the error has been caused by another component that MQSeries Integrator interacts with to complete this command (Windows NT, DB2, or MQSeries), so check for errors from these products too.

On completion, the following have been done:

1. The broker `MQSI_SAMPLE_BROKER` has been created, and the Windows NT service for it added to the Services (viewable from the Windows NT Control Panel). The service is called *IBM MQSeries Broker MQSI\_SAMPLE\_BROKER*. It has a default start up status of manual which you can change to automatic if you want.
2. The MQSeries resources required by the broker have been defined. These resources are detailed in “Default MQSeries resources” on page 95.
3. The required authorizations for the MQSeries resources have been set (the *setmqaut* messages seen in the responses to the command).
4. The database tables required by the broker have been set up in the database `MQSIBKDB`. These tables are listed in Table 9 on page 92.
5. The Windows NT registry has been updated to record the broker creation. Registry updates are described in “Windows NT registry” on page 91.

## Checking the components

You can check the existence of the MQSeries Integrator components you have created. This command is not available in the Command Assistant: you must enter the following command at the command prompt:

```
mqsilist
```

This interrogates the Windows NT registry and displays a list of the major components, with the queue manager which supports them. Now that both broker and Configuration Manager have been created, the command responds with:

```
BIP8099I: ConfigMgr - MQSI_SAMPLE_QM
BIP8099I: MQSI_SAMPLE_BROKER - MQSI_SAMPLE_QM
```

BIP8071I: Successful command completion.

### Starting your broker domain

When you have created the MQSeries Integrator components, you can start to activate your broker domain. You must have Windows NT **Administrator** authority to complete these steps.

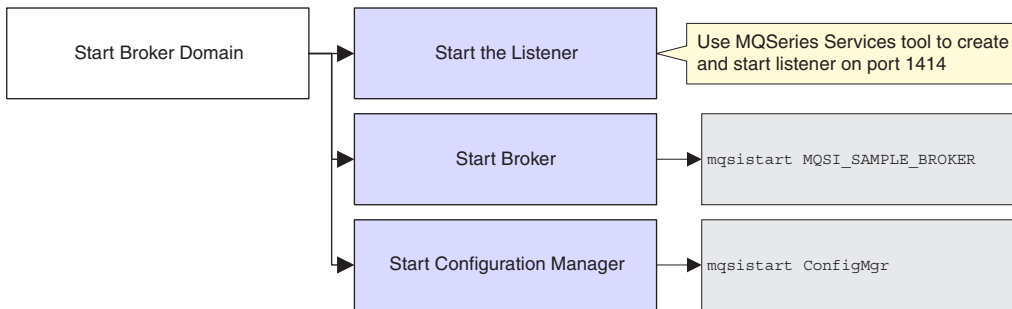


Figure 12. Starting your broker domain

You are recommended to complete the tasks described here in the following order:

- Step 1. Start the MQSeries listener.
- Step 2. Start the Configuration Manager.
- Step 3. Start your broker.
- Step 4. Start the Control Center.

The MQSeries Integrator commands are used to illustrate these steps, with sufficient information provided to complete the task. For a full description of these commands, and possible errors, see the *MQSeries Integrator Administration Guide*. (You cannot use the Command Assistant for these commands.) If errors are reported, you can also check the Application view of the Windows NT Event Viewer.

#### Step 1: starting the MQSeries listener

Most of the resources you need to support this simple configuration have already been created and started for you when you invoked the create broker and create Configuration Manager commands. There is just one extra step you need to take to enable the Control Center to communicate with the Configuration Manager.

## Starting your broker domain

- Step 1. Start the listener on the queue manager. You can use either one of two methods to do this:
- a. You are recommended to use MQSeries Services (*Start->Programs->IBM MQSeries->MQSeries Services*). Expand the left-hand pane and find and left-click the queue manager (MQSI\_SAMPLE\_QM) to display its services in the right-hand pane. If the Listener is listed, right-click the Listener, and select *All Tasks->Start*. This starts the listener as a background task.  
If the Listener is not listed, right-click the queue manager and select *New->Listener*. This creates a listener with default properties of transport type TCP and port 1414. When it has been created, right-click the Listener and select Start.  
This starts the listener as a background task.
  - b. If you prefer, you can use the following command on the command line:

```
runmqclsr -t tcp -p 1414 -m MQSI_SAMPLE_QM
```

When you use this command the listener is started as a foreground task and is not displayed in the MQSeries Services window.

**Note:** If the default MQSeries port 1414 is not available (perhaps because it is already in use by another queue manager), you must assign a different port number that is suitable. The port value must be set in the Listener properties dialog (Parameters tab), or as the `-p` parameter on the `runmqclsr` command. If the port is already in use, the Control Center will not be able to contact the Configuration Manager. For example, if you have set up a default queue manager on this system, it probably already has a listener started on this port. You can check what listeners are already active using MQSeries Services.

### Step 2: starting the Configuration Manager

Start your Configuration Manager by entering the following command on the command line (you cannot do this using the Command Assistant):

```
mqsisstart configmgr
```

This command initiates the start up of the Configuration Manager's Windows NT service and can only report on whether that service is started successfully.

You **must** check the Application view of the Windows NT Event Viewer to ensure that there were no warnings or errors.

The Configuration Manager cannot contact a broker until a reference to that broker exists in the configuration repository. You must create this reference to

the broker using the Control Center. The steps required to do this are described in “Preparing for verification” on page 63.

### **Step 3: starting the broker**

Start your broker by entering the following command on the command line (you cannot do this using the Command Assistant). If you are not using the sample broker name, substitute your broker name for MQSI\_SAMPLE\_BROKER in the command.

```
mqsisstart MQSI_SAMPLE_BROKER
```

This command initiates the start up of the broker’s Windows NT service and can only report on whether that service is started successfully.

You **must** check the Application view of the Windows NT Event Viewer to ensure that there were no warnings or errors.

## Starting your broker domain

### Step 4: starting the Control Center

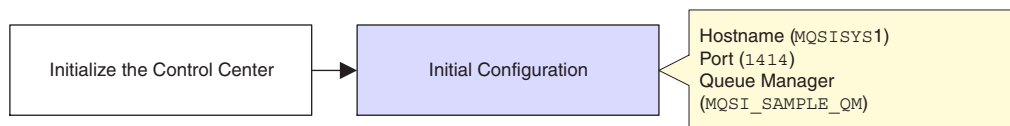


Figure 13. Starting the Control Center

Start the Control Center by double-clicking the Control Center icon in the MQSeries Integrator program folder, or by using the Windows NT Start menu (*Start->Programs->IBM MQSeries Integrator 2.0.1->Control Center*). Complete the following tasks to set up the environment you need to complete the simple verification described in “Verifying your installation” on page 63.

This section gives only a minimum of information required to complete your initial broker domain setup. For further information about the Control Center, refer to *MQSeries Integrator Using the Control Center*.

**Step 1.** Complete the initial dialog presented by the Control Center, *Configuration Manager Connection*, to provide the information needed to connect your Control Center session to the Configuration Manager. The fields are:

- a. Hostname. This is initially blank. Enter the network hostname of the system on which the Configuration Manager has been created. In the simple configuration defined in this chapter, the value you must enter here is MQSISYS1. If you are using a different host name, enter your value here.
- b. Port. This is initially blank. Enter the number of the port on which the queue manager is listening (you set this up in “Step 1: starting the MQSeries listener” on page 59: the default is 1414).
- c. Queue Manager name. This is initially blank. Enter the name of the queue manager (MQSI\_SAMPLE\_QM). This queue manager already has a definition for the server connection required by the Control Center (the channel SYSTEM.BKR.CONFIG of type SVRCONN), which was created when the Configuration Manager was created.

When you have completed these fields, click **OK**. The Control Center now contacts the Configuration Manager, which might take a few minutes. If the Control Center fails to make contact, the most likely reasons are:

- The Configuration Manager has not started successfully.
- The listener has not started successfully.
- The queue manager is not available.



- You are logged on to the local security domain, but this user ID is not a member of the MQSeries Integrator groups. Check which groups your current user ID is a member of. Also, check that you are logged on to the same security domain as the one you were logged on to when you installed MQSeries Integrator.

Check for MQSeries or MQSeries Integrator entries in the Windows NT Event log (Application view) to track down the problem.

If you want to check, or change, these settings at a later time, click *File->Connection* to bring up the connection dialog.

---

### Verifying your installation

You have now completed the configuration and activation tasks. This section explains how to deploy your broker domain, and how to verify your installation. You can choose to run one or more of a set of verification programs, that illustrate different aspects of set up and operation:

- “Preparing for verification”.
- “Running the predefined verification applications” on page 67.
  - “Running the Results Service application” on page 67.
  - “Running the Scribble application” on page 70.
  - “Running the Postcard application” on page 71.
- “Building and using a simple message flow” on page 74.

All the tasks illustrated here assume you have used the sample names and values when you completed the tasks in “Configuring a simple broker domain” on page 41. If you have changed any of these names or values, make sure that you use your values in this section.

You complete most of the tasks involved in running these verification programs using the Control Center. This section gives the minimum information you will need to complete these tasks. For further information, see *MQSeries Integrator Using the Control Center*.

### Preparing for verification

Before you can run any of the verification programs, you must complete some preparation.

#### Creating the MQSeries resources

The verification applications require local queues on the broker’s queue manager. This step creates the MQSeries queues needed by the applications. The queues are:

- For the Results Service application
  - MQSI\_SOCCER\_PUBLICATION\_QUEUE

## Preparing for verification

- MQSI\_SOCCER\_SUBSCRIPTION\_QUEUE
- For the Scribble application
  - MQSI\_SCRIBBLE\_PUBLICATION\_QUEUE
  - MQSI\_SCRIBBLE\_SUBSCRIPTION\_QUEUE
- For the Postcard application
  - MQSI\_POSTCARD\_INPUT\_QUEUE
  - MQSI\_POSTCARD\_OUTPUT\_QUEUE

An MQSC command file is provided to define these resources. The file is in the `examples\mqsc` subdirectory under the MQSeries Integrator home directory. From the command line change to this directory and type the following:

```
runmqsc MQSI_SAMPLE_QM < samples.tst
```

If you prefer, you can use MQSeries Explorer or an interactive MQSC session.

### Importing and deploying the MQSeries Integrator resources

You must now work with the MQSeries Integrator resources that are used by the applications.

- Step 1. Stop the Control Center and the Configuration Manager (`mqsisstop configmgr`).
- Step 2. Import the message set required by the Postcard application into the message repository. The message set is defined in the file `PostcardMS.mrp`.
  - a. Change to subdirectory `examples\postcard` in the MQSeries Integrator home directory. Enter the following command:

```
mqsimrmimpexp -i MQSIMRDB mqsiuid mqsiw PostcardMS.mrp
```

**Note:** The user ID and password used on this command must be the values that you specified for message repository access when you created the Configuration Manager (flags `-e` and `-r`). If you specified different values for these parameters, you must substitute the correct values here.

For more details of this command, see the *MQSeries Integrator Administration Guide*.

- Step 3. Restart the Configuration Manager (`mqsisstart configmgr`). The Configuration Manager can now access the new message set and make it available.
- Step 4. Restart the Control Center and select the *Topology* view. Check out the broker domain topology by selecting the topology, right-clicking, and selecting *Check Out*. This locks the topology, indicated by the key icon, and allows you to make changes to it.

- Step 5. The title bar currently shows that you have an empty workspace, by displaying *Untitled*. You must import the supplied workspace import file that defines the resources used by the verification programs.
- Select *File->Import*. The Import dialog is displayed. This allows you to select the type of resources you want to import, and the file that contains the resource definitions.
  - The valid resource types to import are:
    - Message flows
    - Topics
    - Topology

The file supplied by MQSeries Integrator contains message flow and topology definitions, so you must select these two types.

- Click **Browse** and locate the `\examples` subdirectory (in the MQSeries Integrator home directory). Select the sample workspace import file `SamplesWorkspaceForImport`, click **Open**, and then click **Import**.

You see a dialog box asking you if you want to save. Select **No**.

The definitions can take a few minutes to import. When import has finished, a message dialog is presented, confirming that the resources have been imported successfully. Click **OK** to dismiss the dialog. You will now see the sample broker (MQSI\_SAMPLE\_BROKER) in the topology.

- Select the *Message Flows* view. Check that the import has created two new folders of message flows, *Verification message flows* and *IBM Default message flows*. The default message flows are described in *MQSeries Integrator Introduction and Planning*. The default message flow folders are in addition to the *IBMPrimitives* folder. If you expand the tree for the verification message flows you can see three new message flows, one for each of the verification programs. They are *ScribbleInversion*, *Soccer*, and *Postcard*.

- Step 6. Select the *Message Sets* view. Right-click the Message Sets root and select *Add to Workspace->Message Set*. Select *PostcardMS* from the list and click **Finish**. The Postcard message set is now displayed in the left-hand pane.

The following steps show you how to add all the elements of the Postcard message set to your workspace. These steps are optional, but you are recommended to complete them to help you understand the message set and its contents.

- Expand the Postcard message set and right-click on *Elements*

## Preparing for verification

- b. Select *Add to Workspace->Element*. Use Ctrl+left-click to select all the displayed elements (Country, Duration, Goodtime, Location, MessageText, Recipient, Sender, and Weather). Click **Finish** to add all these to your workspace.
  - c. Right-click on *Element Lengths* within the Postcard message set. Select *Add to Workspace->Length*. Use Ctrl+left-click to select all the displayed element lengths (COUNTRY\_LEN, DEFAULT\_CHAR\_LEN, LOCATION\_LEN, MESSAGETEXT\_LEN, RECIPIENT\_LEN, SENDER\_LEN, and WEATHER\_LEN). Click **Finish** to add all these to your workspace.
  - d. Right-click on *Types* within the Postcard message set. Select *Add to Workspace->Compound Type*. Select Postcard\_Message\_TYPE and click **Finish** to add this type.
  - e. Right-click on *Messages* within the Postcard message set. Select *Add to Workspace->Message*. Select PostcardMessage and click **Finish** to add this message.
- Step 7. Select the *Assignments* view. Select and expand *Message Sets*, and drag and drop the Postcard message set on the sample broker in the right-hand pane.
- Step 8. Check that the message flows have already been assigned to the broker's default execution group (this happened when you imported the workspace); they should be showing in the Domain hierarchy and Topology panels.
- Step 9. You must now save the changes that you have made. Select *File->Check in->All (Save to Shared)*. This causes two things to happen:
- a. The contents of the configuration repository are updated with the new definitions and assignments and everything is checked in to the repository.
  - b. The new workspace is saved locally. Because this is a new workspace, you are asked for a name for this workspace. Enter a name, for example *SampleWorkspace*, and click **Save**. This name now appears in the title bar.
- Step 10. Now you must deploy your changes to the broker. When you deploy, the Configuration Manager sends information to the broker about the resources it needs to support the message flow services.
- a. Select the *Topology* view.
  - b. Select *File->Deploy->Complete Configuration (all types)->Normal*, or right-click the Topology root and select *Deploy*.  
A message dialog confirms initiation. Select **OK** to dismiss the dialog.
- Step 11. Select the *Log* view and refresh the contents by clicking the green refresh icon. It can take a few minutes for all the deployment messages and responses flowing between the Configuration

Manager and the broker to be displayed. Keep refreshing this view until you see the completion messages. If everything is successful, the log contents appear with text that is similar to the following:

```
This message is generated at 2000-03-07 12:27:08
BIP4040I: Configuration changed successfully for Execution Group "default
The message broker received a configuration message and updated its
configuration accordingly.
```

No action required.

```
*****
```

```
This message is generated at 2000-03-07 12:57:48
BIP2056I: Broker MQSI_SAMPLE_BROKER successfully processed the entire
internal configuration message.
An internal configuration message was processed to completion.
```

No user action required.

**Note:** There may be a third item in the log concerning the “pub-sub capability”.

- Step 12. View the deployed configuration graphically in the *Operations view*. Refresh the view, and the topology view is displayed.

### Running the predefined verification applications

This section describes how to run each of the three applications supplied with MQSeries Integrator. You can run any of these, in any order, immediately after installation or at any time in the future. If you choose to run these applications later, make sure you have your system set up in the same way as the system you configured in “Configuring a simple broker domain” on page 41 (or make the appropriate adjustments as you follow these steps).

The verification applications also illustrate how MQSeries Integrator can be used to transform and route messages outside the programming logic of the participating applications, which can therefore run unaffected by updates to that transformation logic, or routing logic, or both.

#### Running the Results Service application

The Results Service application is written in the C programming language and demonstrates a number of basic publish/subscribe features. The application is a simple implementation of a soccer match results gathering service. It consists of one or more publisher applications, and one subscriber application. You can find the files that make up this application (source, header files, and executables) in the `examples\Soccer` subdirectory in the MQSeries Integrator home directory.

You can run this application by following these steps:

- Step 1. Start the subscriber application **soccerresults**.

## Results Service

You must start a single subscriber that subscribes to all soccer matches being played, and displays the results for them. The subscriber application functions as a results server. You must start soccerresults before you start any instances of the publisher application, so that the results server does not miss any publications. You can start the Results Service as follows:

- a. Change to the application directory in one of the following ways:
  - Select *Start->Programs->IBM MQSeries Integrator 2.0->Samples >Soccer ->Soccer Results Server*. This will open a command window in the correct path for this program.
  - Or, if you prefer, open a command window and change to the `examples\Soccer\bin` directory.
- b. Enter the command  
`soccerresults MQSI_SAMPLE_QM`

A message is displayed by the results server indicating that it has registered a subscription and started successfully, and you can now start the match simulator (publisher).

### Step 2. Start the publisher application, **soccergame**.

You can run one or more publishers. Each instance publishes event publications on a single soccer match. You must specify two soccer teams as input to soccergame.

You can start the publisher application as follows:

- a. Change to the application directory in one of the following ways:
  - Select *Start->Programs->IBM MQSeries Integrator 2.0->Samples >Soccer ->Soccer Game*.
  - Or, if you prefer, open a command window and change to the `examples\Soccer\bin` directory.
- b. Enter the command to start up a soccer game. You can use the “\_” character to represent a space in the name of a team.

**Note:** Team names can only contain the characters 0-9, a-z, and A-Z. For example:

```
soccergame Team1 Team2 MQSI_SAMPLE_QM
```

```
soccergame Arsenal Manchester_United MQSI_SAMPLE_QM
```

**How the Results Service works:** The Results Service application uses messages that have a standard MQSeries header, an MQRFH2 header, and a string that specifies the playing teams and their scores.

The soccer simulator **soccergame** publishes an event publication following this message template to the queue `MQSI_SOCCER_PUBLICATION_QUEUE` on

the broker's queue manager. The `MQInput` node in the *Soccer* message flow has been set up so that it identifies this queue as its input queue.

The input node retrieves the publication from this queue forwards it to the publication node. The publication can indicate:

- A match has started.
- A goal has been scored.
- A match has ended.

The results server **soccerresults** subscribes to all these event publications arriving on queue `MQSI_SOCCER_SUBSCRIPTION_QUEUE`. It processes these messages and displays the information: the start of a new game, a score update, and the end of a game.

One important feature of the soccer simulator **soccergame** is its ability to maintain a current state of all the matches being played (the multiple publishers). It achieves this by publishing a retained publication message to the broker with the latest score of each match every time the score changes. This means you can restart the results server after a failure, and the results server subscribes to all these retained publications to restore the current match state to the state it had the last time the results server was running.

If you want to see this use of retained publications, you can start several instances of the publisher application **soccergame**. When these are running, and a couple of goals have been scored, change to the window running the results server application and prematurely kill that process<sup>2</sup> using `Ctrl-C`.

Wait about 30 seconds, then restart the results server **soccerresults**. You will see that the matches being played are restored to their last known score, and updated by any remaining match changes that occurred whilst the results service was stopped.

If you restart the results server too quickly, it might fail to open the subscriber queue with reason 2042 (`MQRC_OBJECT_IN_USE`). This is because the queue manager has not yet recognized that the application has failed, and has therefore not released the queue which the application opened exclusively. You can retry the restart after a few seconds: once the queue is available it will succeed.

See the *MQSeries Integrator Programming Guide* for more details about the implementation of this application and the publish/subscribe techniques it uses.

---

2. If you have created your broker to run as an MQSeries trusted application, you must not terminate this application in this way, as the queue will not be released. For more information about MQSeries Integrator and MQSeries trusted applications, see *MQSeries Integrator Introduction and Planning*.

### Running the Scribble application

The Scribble application is written in Java and demonstrates a number of basic publish/subscribe and message transformation features. In contrast to the Results Service which works with multiple publishers and one subscriber, Scribble works with one publisher and any number of subscribers.

The publisher publishes the current coordinates of the line being drawn in its window, and each subscriber receives the inverted coordinates and displays the resulting drawing in its window. You can find the files that make up this application (source, header files, and executables) in the `examples\scribble` subdirectory in the MQSeries Integrator home directory.

You can run this application by following these steps:

**Step 1.** Start the publisher application.

You can start the publisher using either of these two methods:

- a. Select *Start->Programs->IBM MQSeries Integrator 2.0->Samples->Scribble->Publisher*.
- b. Open a command window, change to the `examples\scribble` subdirectory and run *ScribblePublisher.bat*.

You now see a dialog that prompts you for the broker queue manager name. Enter `MQSI_SAMPLE_QM` and click **OK**. A confirmation dialog, *Scribble ready*, is displayed. Click **OK**. The publisher window is displayed.

**Step 2.** Start the subscriber application.

You can start the subscriber using either of these two methods:

- a. Select *Start->Programs->IBM MQSeries Integrator 2.0->Samples->Scribble->Subscriber*.
- b. Open a command window, change to the `examples\scribble` subdirectory and run *ScribbleSubscriber.bat*.

You now see a dialog that prompts you for the queue manager name. Enter `MQSI_SAMPLE_QM`. The dialog also allows you to enter a queue name. If you want to use the default subscriber queue, `MQSI_SCRIBBLE_SUBSCRIPTION_QUEUE`, you do not have to specify this. If you are using a different queue, you must define that queue and enter the name here. Click **OK**. The subscriber window is displayed.

**Step 3.** Start dragging the mouse with either mouse button depressed to draw lines in the publisher window. These lines appear inverted in your subscriber window.

You can start multiple scribble subscribers, but you must specify a different queue for each one. The definitions you completed in "Creating the MQSeries



resources” on page 63 contain a single subscriber queue for this application, the default subscriber queue `MQSI_SCRIBBLE_SUBSCRIPTION_QUEUE`. If you want to start additional subscribers, you must define additional queues like the default one, and enter the queue name as well as the queue manager name at the dialog you see when you first start the subscriber.

**How Scribble works:** The Scribble application uses messages that have a standard MQSeries header, an MQRFH2 header, and a message body formatted in XML that specifies the drawing coordinates. When you drag the mouse across the publisher window with a mouse button depressed, it draws a line that the publisher records as a set of coordinates. It publishes each set of coordinates in an XML message to the publication queue `MQSI_SCRIBBLE_PUBLICATION_QUEUE`. The MQInput node in the *ScribbleInversion* message flow has been set up so that it identifies this queue as its input queue.

The input node retrieves the publication from this queue and the message flow inverts the drawing by manipulating the coordinates (transformation), and publishes the resulting drawing (routing) to each ScribbleSubscriber’s subscription queue (`MQSI_SCRIBBLE_SUBSCRIPTION_QUEUE` is the default).

The details of the transformation and routing performed by the *ScribbleInversion* message flow are:

- Receive the published message in the MQInput node.
- Filter on the publish/subscribe topic *scribble/coord* in the **FilterOnTopic** node. This node is of primitive type Filter.  
If the match is successful, the message is passed to the **InvertCoordinates** node, which is of primitive type Compute, for transformation.  
If no match is found, the message is sent directly to the publication node without inversion.

You can see the SQL code that implements the inversion of the coordinates in the message. Select the *Message Flows* view in the Control Center, select *ScribbleInversion* in the tree in the left pane, click the **InvertCoordinates** node with the right mouse button to display the node’s context menu, and select *Properties*.

### Running the Postcard application

The Postcard application is based on the MQSeries for Windows NT Version 5.1 Postcard application, and has been extended to demonstrate the transformation capabilities of MQSeries Integrator Version 2.0.1. It is written in C with a Java end-user interface.

## Postcard

Postcard allows you to send a postcard to another nickname, either known to this instance, or to a different instance, of the application. You must run this application on the same system as the broker. You can find the files that make up this application (source, header files and executables) in the `examples\postcard` subdirectory under the MQSeries Integrator home directory.

You can run this application by following these steps:

**Step 1.** Start the first Postcard application.

This first instance acts as the sending application. Select *Start->Programs->IBM MQSeries Integrator 2.0->Samples ->Postcard->Postcard*.

You now see a dialog that prompts you for a nickname to use for sending or receiving messages. You must enter an alphanumeric string of up to 24 characters. The dialog also asks you for the name of the broker queue manager. This is an optional field, but if you do not enter a queue manager name, the default queue manager is used. Enter `MQSI_SAMPLE_QM` to specify the correct queue manager for verification. Click **OK**. A Postcard window appears.

**Step 2.** Start the second Postcard application.

This second instance acts as the receiving application. You must enter a second nickname and the name of the broker queue manager. Click **OK**. A second Postcard window appears.

**Step 3.** Fill in the postcard and send the message.

In the sender (first) Postcard window, fill in the **To** field with the nickname of the receiving (second) postcard application. Fill in the remaining fields from the pulldown menus to build the content of the postcard (location, length of stay, and weather). Click the **Send** button. You see the message, marked *Sent*, in the box called "Postcards sent and received (transformed)" in the lower part of the Postcard window.

**Step 4.** View the received postcard message.

You will see the postcard arrive in the second (receiving) Postcard application. Select the received message in the list and click **View** to see the contents of the (received and transformed) message. The message has been transformed to include the country of the city from which the message was sent.

**Step 5.** Return a postcard to the original sender.

From the second Postcard application, select a message from the "Postcards sent and received (transformed)" box and click **Reply**. This gives you a new postcard to fill in, with the first application's nickname already in the **To** field. Fill in the remaining fields and click **Send** to send the new postcard to the first application.

**How Postcard works:** The Postcard application sends messages to, and receives messages from, the *Postcard* message flow. You deployed this message flow to the broker in “Preparing for verification” on page 63. You also stored the message set *PostcardMS* in the message repository, and deployed it to the same broker. The message set is referenced by the message processing nodes within the flow. It contains one message, *PostcardMessage*, that defines these elements:

1. **Location** (of type STRING)
2. **Country** (STRING)
3. **MessageText** (STRING)
4. **Duration** (INTEGER)
5. **Recipient** (STRING)
6. **Sender** (STRING)
7. **GoodTime** (INTEGER)
8. **Weather** (STRING)

The elements of type STRING each have an associated element length that defines the maximum number of characters valid in this element.

The application program creates and interprets the messages based on a structure defined in the C header file `postcardstruc.h` in the `examples\postcard` subdirectory. This header is an identical representation of the message in the message set in the Control Center.

When a user sends a postcard, the application puts a message to the queue associated with the *Postcard* message flow (`MQSI_POSTCARD_INPUT_QUEUE`). The message flow provides the following message processing:

- The `MQInput` node retrieves messages from the input queue `MQSI_POSTCARD_INPUT_QUEUE`.
- The input node passes the message to node **AddCountry**, an instance of the `IBMPrimitive Compute` node. This node enhances the content of the message by adding the `Country` field, containing the country of the location you selected when you sent the postcard (for example, if you selected “Adelaide”, it adds “Australia”).
- The **AddCountry** node now passes the message to the `MQOutput` node which puts it to the output queue (`MQSI_POSTCARD_OUTPUT_QUEUE`).
- The receiving postcard application retrieves the message from the output queue, reads and interprets the content, and displays the element content in the corresponding fields of the Postcard user interface window.

Although this application has a message set already defined (see “Importing and deploying the MQSeries Integrator resources” on page 64), you can create a message set based on a C header file (like `postcardstruc.h`) by importing

## Postcard

that C structure into the message repository using the import function of the Control Center. This is explained in detail in *MQSeries Integrator Using the Control Center*.

**Note:** Do not try this after importing PostcardMS: this would create duplicate entries and cause problems with deployment.

### Building and using a simple message flow

This verification scenario illustrates how you define a very simple message flow, how you assign the resources to the broker, and deploy your changes. It uses MQSeries Explorer to send messages through the message flow you create. It does not use any defined message sets. It assumes you are using the sample broker that you created in “Configuring your broker domain” on page 52. If you want to use another broker, you must ensure that you create it (using `mqsicreatebroker`) and define it in the configuration repository (from the *Topology* view in the Control Center).

The tasks assume that the broker, sample queue manager, and configuration manager are running.

The following tasks are described:

- “Creating the MQSeries resources”.
- “Creating a simple message flow”.
- “Assigning the message flow to the broker” on page 75.
- “Testing the message flow” on page 76.

#### Creating the MQSeries resources

This simple verification tests needs two queues, one for input, the other for output. This section shows how to use MQSeries Explorer to perform these definitions.

- Step 1. Start MQSeries Explorer (*Start->Programs->IBM MQSeries->MQSeries Explorer*).
- Step 2. Expand the tree under the queue manager MQSI\_SAMPLE\_QM.
- Step 3. Right-click the Queues folder and select *New->Local Queue*.
- Step 4. Enter the queue name MQSI\_INQ in the *Queue Name* field on the *Create Local Queue* dialog. You can leave all the other properties of the queue to take default values.
- Step 5. Repeat the last two steps to define the output queue MQSI\_OUTQ. Again, you can allow all other properties to take a default value.

#### Creating a simple message flow

You must now create the message flow that will process the messages you put to your input queue. The message flow is very simple: the processing it does is to retrieve the message from the input queue and put it to the output queue!

- Step 1. Select *File->New Workspace* to create a new (untitled) workspace. If you have run the verification applications, and have your sample workspace already open, you can create your new message flow in this workspace if you prefer.
- Step 2. Select the *Message Flows* view.
- Step 3. Right-click on the Message Flows root and select *Create->Message flow*.
- Step 4. Enter the name MQSI\_TEST. Click **Finish**. The new message flow appears in the tree view.
- Step 5. Expand the IBMPrimitives tree to display the supplied nodes.
- Step 6. Select the MQSI\_TEST message flow in the left-hand pane. Drag and drop an MQInput node into the right-hand pane.
- Step 7. Right-click the MQInput node in the right-hand pane and select *Properties*. On the Basic tab, type the MQSeries input queue name of your input queue (MQSI\_INQ). Click **OK**.
- Step 8. Drag and drop an MQOutput node into the right-hand pane.
- Step 9. Right-click the MQOutput node in the right-hand pane and select *Properties*. On the Basic tab, type in the queue manager name (MQSI\_SAMPLE\_QM) and the queue name (MQSI\_OUTQ) for the output queue. Click **OK**.
- Step 10. Right-click the MQInput node and select *Connect->Out*. This gives you a connector attached to your mouse. Drag this to the MQOutput node and drop by left-clicking. The connector attaches itself to the input terminal.
- Step 11. You have now completed your first message flow. Select *File->Check in->All (Save to Shared)*. This checks in all the resources to the configuration repository and saves a local copy of the workspace file. If you created a new workspace for this new message flow, you will be prompted to give the workspace a name when you save it.

### Assigning the message flow to the broker

Now you have created a message flow, you have to tell MQSeries Integrator where you want to run that message flow (that is, on which broker). To do this, you must assign the message flow to your broker.

- Step 1. Select the *Assignments* view.
- Step 2. Expand the broker name (MQSI\_SAMPLE\_BROKER) to display the broker's execution groups. The sample broker currently just has one execution group, the default one (called *default*) which is always created whenever you create a broker using **mqsicreatebroker**.
- Step 3. Right-click the default execution group and select *Check Out*. This locks the execution group for you.
- Step 4. Expand the *Message flows* tree in the center pane. This displays all the message flows available for assignment.

## A simple message flow

- Step 5. Find MQSI\_TEST and drag and drop it on the default execution group in the right-hand pane, where you can see the graphic of the broker and the default execution group. You can only drop a message flow on an execution group (not on the broker itself).
- Step 6. Check in the execution group by right-clicking the default execution group in left-hand pane and selecting *Check In*.

### Deploying the message flow to the broker

Assignment makes the connection between a message flow and a broker, but it is only when you deploy the change that the Configuration Manager updates the broker with the configuration stored in the configuration repository.

- Step 1. Before you can deploy any changes, you must have checked in everything that you have updated. If you have followed the instructions in this section, all the relevant resources are checked in. However, if you are in any doubt, you can check everything in by selecting *File->Save to Shared*.
- Step 2. In the *Assignments* view, right-click the broker name in the left-hand pane.
- Step 3. Select *Deploy->Complete Assignments Configuration*. When the Configuration Manager receives this request from the Control Center, it sends messages to the broker to give it the updated information it needs to be able to support your new message flow.
- Step 4. Check the deploy by changing to the *Log* view and clicking the refresh button (the green icon above the log pane). Check for success messages. (There might be a slight time delay before the messages appear).
- Step 5. View the deployed configuration graphically in the *Operations* view. Refresh this view and the broker, execution group and message flow are displayed with green lights, to show they are all active.

### Testing the message flow

You can use MQSeries Explorer to test your message flow.

- Step 1. Start MQSeries Explorer from the Start menu (*Start->Programs->IBM MQSeries->MQSeries Explorer*).
- Step 2. Expand the queue manager and the queues folder. Right-click the input queue name (MQSI\_INQ) and select *Put Test Message* from the menu.
- Step 3. Enter the text of your message (it can be any text string). Click **OK** to put it to the queue.
- Step 4. Right-click the output queue name (MQSI\_OUTQ) and select *Browse Messages* from the menu.
- Step 5. Your message is displayed.

Step 6. Your test is complete. Congratulations!

## A simple message flow



---

## Chapter 5. A more complex configuration

This chapter describes how you can enhance the simple broker domain that you created in “Chapter 4. Getting Started with MQSeries Integrator” on page 41. This chapter illustrates the additional configuration tasks required when you:

- Use multiple queue managers to support the MQSeries Integrator components.
- Add a User Name Server to your broker domain.

**Note:** The tasks required to implement topic-based security are not illustrated. This chapter only addresses the component configuration required.

The following topics are described:

- “The components of the complex configuration”.
- “Preparing to extend the broker domain” on page 80.
- “Creating new resources” on page 82.
- “Restarting your broker domain” on page 85.

---

### The components of the complex configuration

The configuration described in this chapter is an extension of the simple configuration described in “Chapter 4. Getting Started with MQSeries Integrator” on page 41. All the assumptions stated in “Designing a simple configuration” on page 44 are still valid, and the following additional assumptions apply:

- The User Name Server has its own queue manager, MQSI\_SAMPLE\_UNQ\_QM on the same physical system (with a TCP/IP hostname of MQSISYS1) as the broker and Configuration Manager.
- The communications channels set up between the two queue managers use the TCP/IP protocol.
- The port number for the broker queue manager is 1414. The port number for the User Name Server’s queue manager is 1415.

You can find further information about configuring more complex broker domains, and details of how to implement more complex security environments (using a primary or trusted security domain), with examples, in the *MQSeries Integrator Administration Guide*.

## Preparing to extend

---

### Preparing to extend the broker domain

Before you can extend your broker domain, you must stop the components that you have already started and modify them to be able to participate in the new extended broker domain.

#### Stopping the Control Center, broker, and Configuration Manager

Before you can change the properties of the broker or Configuration Manager, you must stop these components. Before you stop the Configuration Manager, you are recommended to stop the Control Center.

If your Control Center session is active, select *File->Exit*, or click the exit button in the top right-hand corner of the window.

Now, stop the broker and Configuration Manager by entering the following commands on the command line (you cannot do this using the Command Assistant):

```
mqsistop MQSI_SAMPLE_BROKER
```

```
mqsistop configmgr
```

#### Changing the broker and Configuration Manager

The broker and Configuration Manager must be modified so that they are aware of, and can interact with, the User Name Server. You can use the Command Assistant to do this (select *Start->Programs->IBM MQSeries Integrator 2.0.1->Command Assistant ->Modify Configuration Manager* and *Modify Broker*). The screens are illustrated in Figure 14 on page 81 and Figure 15 on page 81.

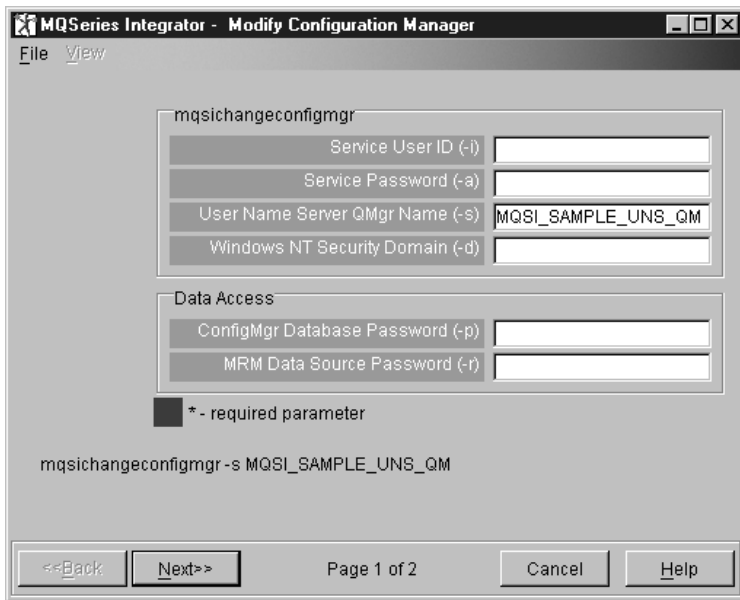


Figure 14. Modify Configuration Manager

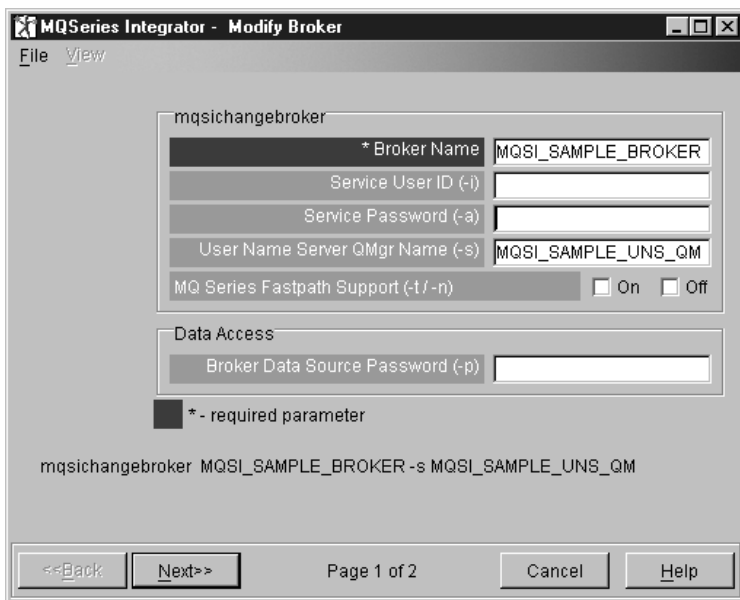


Figure 15. Modify broker

## Preparing to extend

If you prefer, you can use the following commands:

```
mqsicchangeconfigmgr -s MQSI_SAMPLE_UNQ_QM
```

```
mqsicchangebroker MQSI_SAMPLE_BROKER -s MQSI_SAMPLE_UNQ_QM
```

---

## Creating new resources

You must now create new MQSeries Integrator and MQSeries resources to extend your domain.

### Creating the User Name Server

You can create the User Name Server using the Command Assistant (select *Start->Programs->IBM MQSeries Integrator 2.0.1->Command Assistant->Create User Name Server*). You must complete the fields in the screen shown in Figure 16.

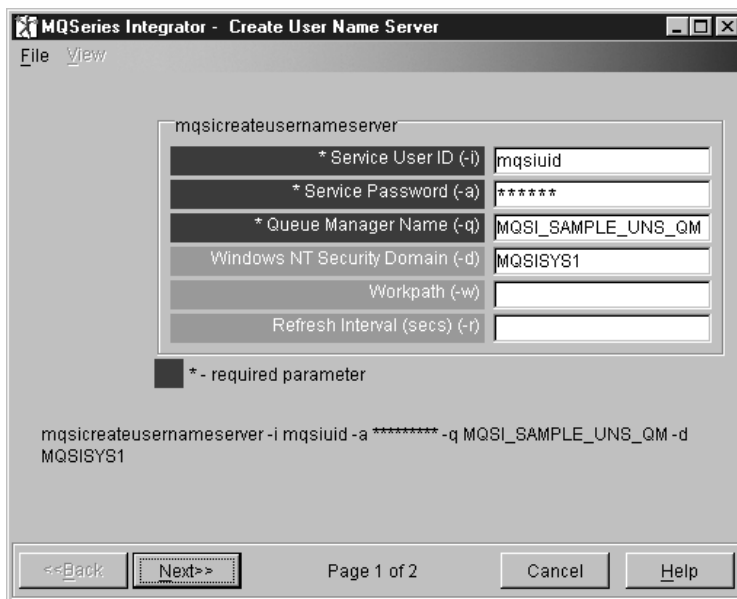


Figure 16. Create User Name Server

If you prefer, you can use the following command:

```
mqsiccreateusernameserver -i mqsiuid -a mqsipw -q MQSI_SAMPLE_UNQ_QM -d MQSISYS1
```

This identifies the queue manager (flag -q) that will host the User Name Server services, the security domain (flag -d) within which user authority is checked (in this case the local system), and the user ID (flag -i) and password (flag -a) that the User Name Server will run under (as a Windows NT service).

**Note:** The queue manager will be created for you if it does not already exist.

If you are using different names or values for any parameter on this command, you **must** replace the appropriate values with your own.

The responses generated by the command are:

```
MQSeries queue manager created.  
Creating or replacing default objects for MQSI_SAMPLE_UNQ_QM.  
Default objects statistics : 29 created. 0 replaced. 0 failed.  
Completing setup.  
Setup completed.  
MQSeries queue manager 'MQSI_SAMPLE_UNQ_QM' started.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
The setmqaut command completed successfully.  
security properties not found. using defaults.  
BIP8071I: Successful command completion.
```

If the command detects any errors, or is unable to complete, it returns an error message on the command line, or in the Windows NT Event viewer (Application View) which gives the explanation and action in full. It is possible that the error has been caused by another component that MQSeries Integrator interacts with to complete this command (Windows NT or MQSeries), so check for errors from these products too. (The response security properties not found. using defaults shown above is not an error: you can ignore this informational message.)

On completion, the following have been done:

1. The User Name Server has been created, and the Windows NT service for it added to the Services (viewable from the Control Panel). The service is called *IBM MQSeries Broker UserNameServer* and has a default start up status of manual, which you can change to automatic if you want.
2. The MQSeries resources required by the User Name Server have been defined on the queue manager MQSI\_SAMPLE\_UNQ\_QM. These resources are detailed in “Default MQSeries resources” on page 95.
3. The Windows NT registry has been updated to record the User Name Server creation. Registry updates are described in “Windows NT registry” on page 91.

### Creating MQSeries resources

Because the User Name Server does not share a queue manager with the broker and Configuration Manager, you must now define the MQSeries resources required to connect the two queue managers together.

You must complete the following tasks:

- Step 1.** Create the channels and the transmission queue on the User Name Server’s queue manager.

## Creating MQSeries resources

- Step 2. Create the channels and transmission queue on the queue manager supporting the broker and Configuration Manager.
- Step 3. Start the listeners and channels.

You can use the MQSeries Explorer or MQSC to define these resources. The definitions are illustrated using MQSC: if you are using MQSeries Explorer you can use the parameters shown in the commands to complete the MQSeries Explorer dialogs. Be careful when you enter object names: if you do not enclose names in quotes on the command line or in MQSC command scripts, the names are folded to uppercase. MQSeries Explorer does **not** fold names to uppercase.

If you use the `runmqsc` command to issue MQSC commands, you must ensure that you issue the commands on the correct queue manager. If the queue manager for the User Name Server is not the default queue manager, you must specify its name on the `runmqsc` command. For example, enter:

```
runmqsc MQSI_SAMPLE_UNSQM
```

For more information about these MQSeries tasks, refer to the *MQSeries Command Reference* for MQSC commands and to the *MQSeries System Administration* book for a description of MQSeries Explorer and the `runmqsc` command.

### User Name Server queue manager resources

The User Name Server must be able to support two way communication with the broker. Follow these steps to set up the resources required for this support by the User Name Server's queue manager. If you are using different names or values, remember to change these as you implement these steps.

- Step 1. Define the transmission queue for messages sent by the User Name Server to the broker and Configuration Manager. This is defined with the same name as the broker's queue manager to which it transmits messages, for easy configuration.

```
define qlocal('MQSI_SAMPLE_QM') get(enabled) put(enabled)  
usage(XMITQ) replace
```

- Step 2. Define the sender channel for messages sent by the User Name Server. The TCP/IP port number specified here, 1414, identifies the target for messages sent on this channel. This port is the one in use by the broker's queue manager. 1414 is reserved for MQSeries use.

```
define channel('MQSI_SAMPLE_UN_TO_BR') chltype(sdr) trptype(tcp)  
conname('MQSISYS1(1414)') xmitq('MQSI_SAMPLE_QM') replace
```

- Step 3. Define the receiver channel for messages received by the User Name Server.

```
define channel('MQSI_SAMPLE_BR_TO_UN') chltype(rcvr) trptype(tcp) replace
```

### Broker and Configuration Manager queue manager resources

The queue manager shared by the broker and Configuration Manager must be able to support two way communication with the User Name Server's queue manager. Follow these steps to set up the broker queue manager's end of the connections. If you are using different names or values, remember to change these as you implement these steps.

- Step 1. Define the transmission queue for messages sent by the broker to the User Name Server. This is defined with the same name as the User Name Server's queue manager for easy configuration.

```
define qlocal('MQSI_SAMPLE_UNQ') get(enabled) put(enabled)
usage(XMITQ) replace
```

- Step 2. Define the sender channel for messages sent by the broker. The TCP/IP port number 1415 is used here as the port on which the User Name Server is listening. 1415 is **not** reserved for MQSeries use. You must therefore change the port number to a new value that is valid in your network, if 1415 is not appropriate.

```
define channel('MQSI_SAMPLE_BR_TO_UN') chltype(sdr) trptype(tcp)
conname('MQSISYS1(1415)') xmitq('MQSI_SAMPLE_UNQ') replace
```

- Step 3. Define the receiver channel for messages received by the broker.

```
define channel('MQSI_SAMPLE_UN_TO_BR') chltype(rcvr) trptype(tcp) replace
```

---

## Restarting your broker domain

When you have created the MQSeries Integrator components, and defined the MQSeries resources required to complete the connectivity in your broker domain, you can start the new MQSeries resources and the new MQSeries Integrator resources, and restart the initial MQSeries Integrator resources.

You are recommended to complete the tasks described here in the following order:

- Step 1. Start the listeners and channels.
- Step 2. Start the User Name Server.
- Step 3. Restart your broker.
- Step 4. Restart the Configuration Manager.
- Step 5. Restart the Control Center.

If any errors are reported when you start these components, check the Application view of the Windows NT Event log. Access this log by clicking on *Start->Programs->Administrative Tools->Event Viewer*.

## Starting the listeners and channels

You must start an MQSeries listener for each queue manager. A listener detects messages inbound on the port specified in the channel definitions. The listener for the broker's queue manager should still be active (you can check

## Restarting

its status using MQSeries Services). You must start a new listener for the User Name Server's queue manager (and create one if it does not already exist).

When the listeners have started, you must start the sender channels (the listener at the receiving end must be active before the send channel can start successfully). If you are using different names or values, remember to change these as you implement these steps.

**Step 1.** Start the listener on the User Name Server's queue manager. You can use either one of two methods to do this:

- a. You are recommended to use MQSeries Services (*Start->Programs->IBM MQSeries->MQSeries Services*). Left-click the queue manager (MQSI\_SAMPLE\_UNQ\_QM) to see its services in the right-hand pane. If the Listener is listed, right-click the Listener, and select *All Tasks->Start*. This starts the listener as a background task.

If the Listener is not listed, right-click the queue manager and select *New->Listener*. This creates a listener with default properties of transport type TCP and port 1414. You must change the port value to 1415 (or another valid port), because the port 1414 is already in use by the broker's queue manager. When the User Name Server' queue manager's listener has been created, right-click the Listener and select *Start*.

This starts the listener as a background task.

- b. If you prefer, you can use the following command.

```
runmq1sr -t tcp -p 1415 -m MQSI_SAMPLE_UNQ_QM
```

When you use this command the listener is started as a foreground task and is not displayed in the MQSeries Services window.

**Note:** If the port 1415 is not available (perhaps because it is already in use by another queue manager), you must assign a different port number that is suitable. The port value must be set in the Listener properties dialog (Parameters tab), or as the `-p` parameter on the `runmq1sr` command. You must ensure that you use the same port number that you specified as the target for the sender channel you defined on the broker's queue manager in "Broker and Configuration Manager queue manager resources" on page 85.

The broker's queue manager listener is already active (you started it earlier in "Step 1: starting the MQSeries listener" on page 59). You can use MQSeries Services to check that it is still active.



Step 2. Start the sender channels. You are recommended to do this using MQSeries Explorer, which starts the channels as background tasks.

If you prefer, you can enter the commands:

```
runmqchl -m MQSI_SAMPLE_UNQ -c MQSI_SAMPLE_UNQ_TO_BR
```

```
runmqchl -m MQSI_SAMPLE_QM -c MQSI_SAMPLE_BR_TO_UN
```

When you use these commands the channels are started as foreground tasks.

### Starting the User Name Server

Start your User Name Server using the following command:

```
mqsistart usernameserver
```

The User Name Server is started as a Windows NT service.

If you start the User Name Server before you restart the broker, the broker is able to retrieve principals information immediately. If you start the broker first, it will periodically try to contact the User Name Server to request the information. Until it has made contact for the first time, the broker is unable to check ACLs, and records this fact in the Windows NT Event log. Once principals information has been returned to the broker by the User Name Server, the broker stores it so that if the User Name Server is unavailable in the future, it can rely on its current store.

However, the Configuration Manager does not maintain a local record of principals information. If it is unable to contact the User Name Server when it starts up, it records this information in the Windows NT Event log and periodically tries to make contact. Until contact has been successful, users of the Control Center are unable to work with topic ACLs (though they are able to complete other tasks that they are authorized to do).

### Restarting the broker

Restart your broker using the following command. If you are not using the sample names, remember to specify the correct broker name.

```
mqsistart MQSI_SAMPLE_BROKER
```

### Restarting the Configuration Manager

Restart your Configuration Manager using the following command:

```
mqsistart configmgr
```

The changes to your broker domain are now complete.

### Restarting the Control Center

Select *Start->Programs->IBM MQSeries Integrator 2.0.1->Control Center* to restart the Control Center. You can check that the broker has reinitialized

## Restarting

successfully by selecting the *Operations* view. The broker and default execution group should appear with a green light indicating they are active. You do not have to redeploy because the broker remembers its configuration.

You are now ready to implement topic-based security. Refer to the *MQSeries Integrator Introduction and Planning* book and the *MQSeries Integrator Administration Guide* for more information about topic-based security.

---

## Appendix A. System changes after installation

This chapter describes the changes that installation and configuration have made on the systems you have set up in your broker domain. It assumes that you have followed the guidance and details of configuration given in “Configuring a simple broker domain” on page 41.

- “Directory structure”.
- “Environment variables” on page 90.
- “Windows NT registry” on page 91.
- “Database contents” on page 92.
- “Default MQSeries resources” on page 95.
- “Principals” on page 96.

The product’s home directory, which defaults to  
C:\Program Files\IBM MQSeries Integrator 2.0.1\

is written as <mqsi\_root> wherever it appears in this chapter.

DB2’s home directory defaults to  
C:\SQLLIB\

---

### Directory structure

Table 8 lists the subdirectories created and populated within your home directory <mqsi\_root> when you do a full installation. It also provides a brief description of the contents. If you have not done a **Full** install, your directory structure will be a subset of those shown.

All files are installed with default security: all users can access and execute these files. You can use standard operating system facilities to impose stricter security on these files, or a subset of files, if you choose. Refer to the Windows NT help for details about file permissions.

*Table 8. Directory structure after installation*

Directory Name	Contents
bin	Executable files, .exe, .dll, .lil
Book	PDF files for V2.0.1 and V1.1
classes	Java class files
CmdAsst	Command Assistant files
examples	C, C++ and Java language samples at V1.1 and V2.0.1 level, MQSC samples

## Directory structure

Table 8. Directory structure after installation (continued)

Directory Name	Contents
gui_11	GUI utility programs for NEONRules and NEONFormatter
include	Header and other files for samples
install.sql	Database install scripts
lib	Shared library files
log	Error log files
messages	Description files for messages and exceptions
mrm	Message repository files
primitives	Message processing node files
Register	Product registration details
Tool	Graphical utility programs
uninst	Uninstall files

If you have installed the Tour plug-ins, other directories will have been updated (depending on your choices during installation).

---

## Environment variables

The following Windows NT system environment variables have been defined and set, or updated, by installation:

- PATH

The value of your PATH variable is updated to include the following:

```
<mqsi_root>\bin;
```

- LIB

If you install the Samples component, this environment variable is updated to include the following:

```
<mqsi_root>\lib;
```

- INCLUDE

If you install the Samples component, this environment variable is updated to include the following:

```
<mqsi_root>\include;<mqsi_root>\include\plugin;
```

- CLASSPATH

If you install the Control Center component, this environment variable is updated to include the following:

```
<mqsi_root>\Tool\help;
```

All other MQSeries Integrator components invoke class files using the `-cp` argument under the Java Runtime Environment installed as part of MQSeries Integrator.

However, if you have installed VisualAge for Java, and selected the MQSeries Connector as part of that installation, you must ensure that the CLASSPATH entry for VisualAge for Java appears **after** the CLASSPATH entries for MQSeries for Windows NT Version 5.1 (server or Java client). This is to ensure that the correct MQSeries classes are accessed by the Configuration Manager, not the VisualAge classes.

If you have also installed DB2 during MQSeries Integrator installation, the values of your PATH and CLASSPATH variables are updated with additional statements. Refer to the *DB2 Installation and Configuration Supplement* if you require further details.

---

### Windows NT registry

The Windows NT registry is updated by installation. The entries made depend on the choices made during installation. They appear under three keys under the top-level key HKEY\_LOCAL\_MACHINE:

- SOFTWARE\IBM\MQSeriesIntegrator  
Entries under this key identify the home directory, program folder and other general product settings. Entries are also added for the components installed and configured on this system.
- SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\MQSeries Integrator V2.0.1  
An entry is added here to allow MQSeries Integrator to be uninstalled from the *Add/Remove Programs* icon.
- SOFTWARE\Microsoft\Windows\CurrentVersion\Run  
An entry is added here to record the licence registration reminder, if selected.
- SYSTEM\CurrentControlSet\Services\Eventlog\Application  
Entries are added when Windows NT services are created (for example, for each broker).

If DB2 Version 6.1 is installed by MQSeries Integrator, additional updates are made to the registry for this product. Refer to the *DB2 Quick Beginnings* book for further details.

## Database contents

---

### Database contents

When you create MQSeries Integrator resources following installation, a number of database tables are created for brokers, for the configuration repository, and for the message repository. They are labelled USERSPACE1. These databases can be individually created for each set of tables, or can be shared.

Table 9 shows the tables that are created by the **mqsicreatebroker** command in the broker database. The tables are created when you create the first broker. When you create further brokers specifying the same database, new rows are created for each broker. Every row created in the table includes the broker name, therefore each row is unique to a single broker.

*Table 9. Database tables for brokers*

Table name	Description
BACLENTRIES	ACL entries
BCLIENTUSER	Maps client identifiers to durable subscriptions
BGROUPNAME	Publish/Subscribe principals: groups
BLOGICALTOPHYSNAME	Maps logical to physical names
BMQPSTOPOLOGY	Publish/Subscribe neighbor information
BNBRCONNECTIONS	Inter-broker neighbor connection information
BPHYSICALFILE	Physical file mapping
BPUBLISHERS	Registered publishers
BRETAINEDPUBS	Retained publications
BRMCONFIG	Broker configuration details
BROKERA	Broker process details to support recovery
BROKERAEG	Execution group details to support recovery
BROKERRESOURCES	Broker resources
BSUBSCRIPTIONS	Durable subscription information
BTOPOLOGY	Inter-broker neighbor information
BUSERCONTEXT	Maps client identifiers to context information
BUSERMEMBERSHIP	Publish/Subscribe principals: membership
BUSERNAME	Publish/Subscribe principals: users
BWFFRELATIONSHIP	Workfile details

Table 10 shows the tables that are created by the `mqsicreateconfigmgr` command in the configuration repository database. There is a single set of these tables, because there is a single Configuration Manager, and therefore a single configuration repository, in your broker domain.

*Table 10. Database tables for the configuration repository*

Table name	Description
CBROKER	Broker details
CBROKERCEG	Execution group to broker mapping information
CCOLLECTIVE	Collectives details
CCOLLECTIVECBROKER	Collective to broker relationships
CDELETED	Deleted resources
CEG	Execution group details
CEGMSGFLOW	Message flow mapping information
CEGMSGPROJECT	Message set mapping information
CLOG	Event messages
CMSGFLOW	Message flow details
CMSGPROJECT	Message set details
CNEIGHBOURS	Broker neighbor topology details
COUTSTANDING	Actions with outstanding responses
CPRIMITIVES	Primitive message processing node details
CPROPERTIES	UUID properties
CSUBSCRIBE	Subscriber information
CTOPIC	Set of all topics available in the broker domain
CTOPICCTOPIC	Topic tree mapping information
CTOPOLOGY	Topology details
CTRACE	Persistent trace information
CUUIDLOCKS	Locking information

## Database contents

Table 11 shows the tables that are created by the `mqsicreateconfigmgr` command in the message repository database. There is a single set of these tables, because there is a single Configuration Manager, and therefore a single message repository, in your broker domain.

*Table 11. Database tables for the message repository*

Table name	Description
CATEGORY_MEMBER	Category members
M_ATTRIBUTE	Attribute details
M_CATEGORY	Category details
M_CONST_DEF	Defined constants
M_CONTEXT_TAG	Context tags details
M_ELEMENT	Element details
M_LENGTH	Element length details
M_MEMBER_ATTRIBUTE	Member properties
M_MESSAGE	Message details
M_TRANSACTION	Transaction details
M_TYPE	Complex type details
M_VALID_VALUE	Valid values defined
MRM_USER	User information
MSG_CONTEXTTAG_MBR	Context tag details
PLUGIN	Plug in node information
PROJECT	Message sets defined
PROJECT_DEP_MEMBER	Message set members
REG_PLUGIN_MEMBER	Plug in members
REPOSITORY	Repository identification
TRANSACTION_MEMBER	Transactions details
TYPE_MEMBER	Types details
USER_MEMBER	Users contents
VALID_VALUE_MEMBER	Valid values contents

You must add all these database tables into your standard backup and recovery routines to ensure you can recover from system crashes and other emergencies.

These tables are maintained by processes that are internal to MQSeries Integrator components. You must not access these tables by any other means,



nor change the access authority required by MQSeries Integrator. You have no requirement to know or understand the contents of any of these tables.

**Default MQSeries resources**

When you create MQSeries Integrator components, some MQSeries resources are created for their use. Table 12 lists all these MQSeries resources, and indicates the component associated with the queue manager on which they are created. For details of which resources are created by which create commands, see the command descriptions in the *MQSeries Integrator Administration Guide*.

All these resource names start with reserved characters "SYSTEM". Therefore you should not find any conflict of names. There is one exception to this: if you have been using MQSeries Publish/Subscribe, it defines queue SYSTEM.BROKER.CONTROL.QUEUE which MQSeries Integrator also uses. However, the use is compatible and you do not have to take any action to continue using this queue.

*Table 12. MQSeries Integrator default objects*

Resource name	Type	Queue manager	Description
SYSTEM.BROKER.ADMIN.QUEUE	queue	broker	Target for messages sent by the Configuration Manager and commands to modify the broker's configuration and operation.
SYSTEM.BROKER.ADMIN.REPLY	queue	Config. Manager	This queue is used for responses to the Configuration Manager from brokers.
SYSTEM.BROKER.CONFIG.QUEUE	queue	Config. Manager	This queue is used for messages from the Control Center to the Configuration Manager.
SYSTEM.BROKER.CONFIG.REPLY	queue	Config. Manager	This queue is used for responses to the Control Center from the Configuration Manager.
SYSTEM.BROKER.CONTROL.QUEUE	queue	broker	Target for publish/subscribe control requests from applications.  A queue of this exact name is used by the MQSeries Publish/Subscribe. You might therefore already have defined a queue of this name on the queue manager. You can continue to use this same queue as you migrate to MQSeries Integrator.

## MQSeries resources

Table 12. MQSeries Integrator default objects (continued)

Resource name	Type	Queue manager	Description
SYSTEM.BROKER.EXECUTIONGROUP.QUEUE	queue	broker	Target for messages to the broker.
SYSTEM.BROKER.EXECUTIONGROUP.REPLY	queue	broker	Target for response messages for the broker from the User Name Server.
SYSTEM.BROKER.INTERBROKER.QUEUE	queue	broker	Target for publications from neighbor brokers.
SYSTEM.BROKER.MODEL.QUEUE	queue	all	Model for dynamic response queues.
SYSTEM.BROKER.SECURITY.QUEUE	queue	User Name Server	Target for request messages to the User Name Server. queue is used by brokers, the Configuration Manager, and the command line tools.
SYSTEM.BROKER.SECURITY.REPLY	queue	Config. Manager	Target for replies from the User Name Server to the Configuration Manager.
SYSTEM.BKR.CONFIG	SVRCONN	Config. Manager	Connection channel for Control Center instances.

**Note:** These resources are defined in addition to the MQSeries product default objects, which are defined when the MQSeries Messaging product is installed. You can find a full description of these default objects in *MQSeries System Administration*.

---

## Principals

The installation program defines the following groups to the system's local security domain:

- **mqbrasgn**
- **mqbrdevt**
- **mqbrkrs**
- **mqbrops**
- **mqbrtpic**

If you install MQSeries Integrator on a Primary Domain Controller the following groups are also defined to the primary domain:

- **Domain mqbrasgn**
- **Domain mqbrdevt**
- **Domain mqbrkrs**
- **Domain mqbrops**
- **Domain mqbrtpic**

**Note:** These global groups are not defined if you install MQSeries Integrator on the domain controller of a trusted domain. For more details about using primary and trusted security domains, see “Using Windows NT primary or trusted security domains” on page 18).

If your local system is configured in a primary or trusted domain, and global groups have been defined in that domain (described in “Security considerations” on page 17), these global groups are added to local groups as follows:

- **Domain mqbrasgn** is added to **mqbrasgn**
- **Domain mqbrdevt** is added to **mqbrdevt**
- **Domain mqbrkrs** is added to **mqbrkrs**
- **Domain mqbrops** is added to **mqbrops**
- **Domain mqbrtpic** is added to **mqbrtpic**

These definitions are not created if you install only the online documentation or the SDK.

## Principals

---

## Appendix B. Uninstalling MQSeries Integrator

This chapter gives details of the processes that allow you to uninstall any one or all of the MQSeries Integrator components on Windows NT.

**Note:** If you installed DB2 when you installed MQSeries Integrator for Windows NT, this is **not** uninstalled when you remove MQSeries Integrator. You must uninstall this independently using the DB2 Uninstall program (from the Control Panel Add/Remove Programs option, or from the Start menu, *Start->DB2 for Windows NT->Uninstall*). You are not authorized to use this copy of DB2 for any purpose other than support of MQSeries Integrator.

If you intend to uninstall MQSeries for Windows NT Version 5.1 as well as MQSeries Integrator, you **must** uninstall MQSeries Integrator first, as it depends on MQSeries function to complete successfully.

You must be logged on with a user ID that is a member of the Windows NT **Administrators** group in the local account security domain to complete uninstallation.

---

### Before you start

When you uninstall MQSeries Integrator, you can choose to uninstall the product data as well as the product files by selecting *Uninstall including data*. The product data includes database table contents and all product files in the product directory. Any files you have saved into the MQSeries Integrator home directory are preserved. However, you are recommended to back up these files before you start to uninstall MQSeries Integrator.

If you choose to uninstall including data, the following items are removed:

- All MQSeries Integrator product directories. This includes the \Tool subdirectory where you might have stored information about your Control Center workspace. (The \Tool subdirectory is the default when you select *File->Save As...* or *File->Export*).
- MQSeries Integrator environment variables.
- MQSeries Integrator Windows NT Registry entries.
- The contents of all broker database tables.
- All MQSeries objects created by MQSeries Integrator, except SYSTEM.BROKER.MODEL.QUEUE

## Uninstalling MQSeries Integrator

You are therefore recommended to take action to preserve any data you might need if you intend to recreate your configuration and workspace.

- If you want to keep a workspace after an uninstall including data, you can use one of the following two methods:
  1. Invoke *File->Save to Shared* against the workspace when it is open in the Control Center, and save it in a directory that is not under the MQSeries Integrator home directory. This saves a local file and records updates in the broker and configuration repository tables. Because uninstall including data destroys the contents of the broker database tables, you must also backup these database tables so that you can recreate them in a new installation if you choose.

**Note:** If you invoke *File->Save As...* against the workspace a local copy is saved in the hard disk of this machine (by default in the `\Tool` subdirectory) and no updates are made to the broker and configuration repository database tables. You cannot recreate data if you only save locally.

2. Export the workspace and save the exported file in a directory that is not under the MQSeries Integrator home directory (*File->Save* defaults to `\Tool`).

*File->Export* creates an XML message containing all the data for the workspace in a form that allows it to be imported into a new installation (for example, for an upgrade or on a new machine). An export of the workspace results in a file that is almost 10 times bigger than a file resulting from saving to shared because it represents a self-contained definition of the workspace.

- If you want to keep message formats created in the message repository, you **must** export them to file using the command `mqsimmimpexp`. See the *MQSeries Integrator Administration Guide* for details of this command.

If you choose the option to uninstall including data, MQSeries Integrator displays a warning dialog that you have to acknowledge by clicking **Next** before you can continue. The message indicates that all components and all product files in the home directory will be destroyed if you continue. If you have taken the recommended steps described, you will be able to recreate this data on another system, or (after reinstallation) on this system.

The following items are never removed by uninstallation, excluding data:

- MQSeries Integrator groups defined to the Windows NT User Manager in the local account domain.
- MQSeries Integrator domain groups in the primary or trusted domain, or both, if you have defined these to the domain server.

## Uninstalling MQSeries Integrator

- The contents of all tables in the configuration repository, and the contents of all tables in the message repository. If you want these removed, you must issue the **mqsdeleteconfigmgr** command specifying flags **-n -m**.
- The databases you have created to support MQSeries Integrator internal data (the broker tables, configuration repository, or message repository). The uninstall program is unable to determine if these databases are used for any other purpose. You must therefore take the appropriate action to drop these databases, if they are no longer required, or remove the tables that have been set up for MQSeries Integrator. You can check the exact list of tables created by MQSeries Integrator by referring to “Database contents” on page 92 (remember that every broker has a set of tables listed in Table 9 on page 92).

You are recommended to stop manually:

- The control center
- The configuration manager
- The username server
- Any brokers that are running

before you proceed.

---

### Using the uninstall program

You can uninstall one or more of the product components in these ways:

1. Start uninstall by selecting MQSeries Integrator from the Add/Remove Programs icon in the Control Panel. This is the recommended method.
2. Run the uninstall program directly from a command prompt, or by using Run on the Start menu. You must specify the correct directory (or change to it) for this to work: the uninstall program is located in the `\uninst` subdirectory of the MQSeries Integrator home directory. By default, this is `C:\Program Files\IBM MQSeries Integrator 2.0.1`

The program is called `mqsunin.exe`.

3. Select *Start->Programs->IBM MQSeries Integrator 2.0->Uninstall*.

When you run this program, you are presented with a window that allows you to select one of:

- Uninstall one or more components.
- Uninstall MQSeries Integrator completely, including data.
- Uninstall all of MQSeries Integrator, excluding data.
- Uninstall CSDs.

The option to uninstall CSDs is unavailable if one of the following is true:

- You have not installed any CSDs for MQSeries Integrator.

## Uninstalling MQSeries Integrator

- You have installed the latest CSD without creating a backup for replaced files.
- The backup for replaced files from the latest CSD installation cannot be found.

If you have installed more than one CSD then the first option will uninstall the latest CSD and restore the previous CSD level. If you have only installed one CSD, this option will restore the base level of code.

Uninstalling MQSeries Integrator excluding data leaves user-added components intact. For example, the Windows NT registry entries for any resources you have created and not deleted (broker, Configuration Manager, User Name Server), and any plug-in nodes. If you subsequently reinstall the product, you can access and reuse this data.

If you decide to uninstall one or more components, you must choose from the list of components to uninstall. This option excludes data: any files you have created are preserved, only product files are removed. If the choices you make mean that no components of the product remain, the uninstall sub-directory and its contents are also removed.

The uninstallation program appends records to the log file created by MQSeries Integrator installation, `mqsi2.log`, which you can find in the Windows NT operating system root directory.

---

### Migration considerations

If you have a previous installation of MQSeries Integrator Version 1 that you did not uninstall prior to installing MQSeries Integrator Version 2, you can now reestablish this product and its environment.

To reset the values required, you must restore the MQSeries Integrator Version 1 information in the PATH environment variable. This was removed when MQSeries Integrator was installed, but has not been restored. You must add the necessary stanza to the PATH variable to restore access to the Version 1 product code.

---

### Manually uninstalling MQSeries Integrator

When you install MQSeries Integrator, you must ensure that you have backup procedures in place that will guard against loss of data and media corruption. However, if you have encountered a situation from which you cannot recover automatically, and that is preventing uninstallation using the normal methods, you can uninstall MQSeries Integrator manually.



## Uninstalling MQSeries Integrator

To uninstall MQSeries Integrator manually, you must:

1. Use the **mqsilist** command to determine the components created on this system (brokers, Configuration Manager, and User Name Server).
2. Stop each component using the **mqsistop** command
3. For each broker on this system, invoke the **mqsdeletebroker** command to delete it. If the command fails, and you cannot resolve the failure, contact your IBM Support Center (refer to “Contacting your IBM Support Center” on page 106 for details of this process).
4. If the Configuration Manager exists on this system, invoke the **mqsdeleteconfigmgr** command to delete it. If the command fails, and you cannot resolve the failure, contact your IBM Support Center (refer to “Contacting your IBM Support Center” on page 106 for details of this process).
5. If the User Name Server exists on this system, invoke the **mqsdeleteusernameserver** command to delete it. If the command fails, and you cannot resolve the failure, contact your IBM Support Center (refer to “Contacting your IBM Support Center” on page 106 for details of this process).
6. If you have successfully deleted all the components on this system, you can remove the MQSeries Integrator Version 2.0.1 keys and subkeys from the Windows NT Registry. Select *Run* from the Windows NT Start menu and enter either *regedt32* or *regedit* to edit the registry.
  - a. Select and delete the MQSeries Integrator Version 2.0.1 registry key:  
HKEY\_LOCAL\_MACHINE\SOFTWARE\  
IBM\MQSeriesIntegrator
  - b. Delete the IBM registration reminder, if present:  
HKEY\_LOCAL\_MACHINE\SOFTWARE\  
Microsoft\Windows\CurrentVersion\Run
  - c. Delete the uninstallation key:  
HKEY\_LOCAL\_MACHINE\SOFTWARE\  
Microsoft\Windows\CurrentVersion\Uninstall\MQSeries Integrator V2.0.1
7. Delete the directory structure that contains the installed product files. If you used the default directory when you installed, the root directory for this structure is:  
C:\Program Files\IBM MQSeries Integrator 2.0.1

**Note:** If you have created any subdirectories holding your own data, or have enhanced the installed product by creating your own message processing nodes or parsers and have stored these in the product directory, save these files before you delete the MQSeries Integrator home directory.

## Uninstalling MQSeries Integrator

If any of these files cannot be deleted because they are in use, one or more MQSeries Integrator processes on this system must still be active. Check that the Control Center is not active. If it is, close it. If you are still unable to delete all files, and cannot identify a process holding the file lock, restart the system.

8. Remove the program folder entries from the Start menu using Windows NT Explorer. Entries for MQSeries Integrator are created by default in  
`C:\winnt\Profiles\All Users\Start Menu\Programs`  
`\IBM MQSeries Integrator 2.0`

If you have moved these entries, or selected a non-default option when you installed MQSeries Integrator, substitute your path for the default path shown.

---

### Unattended Removal of IBM MQSeries Integrator Version 2.0.1 on Windows NT

You can remove (uninstall) IBM MQSeries Integrator Version 2.0.1 without interaction. This process is called unattended (or silent) removal and uses a response file. This is a text file containing values for the options you select when you uninstall IBM MQSeries Integrator V2.0.1.

To create a response file you can do either of the following:

- Edit the response file (`mqsiumin.rsp`) supplied on the IBM MQSeries Integrator Version 2 CD-ROM using an editor such as Notepad.
- Use an editor to create your own response file.

When you have created your response file, run the uninstall program `mqsiumin.exe` from the root directory, with the fully-qualified path to `mqsiumin` in double quotes, as follows:

```
c:\>"c:\Program Files\IBM MQSeries Integrator 2.0.1\uninst\mqsiumin" -s -f1<responsefile>
```

In the preceding example:

The `-s` parameter tells the uninstall program to run in silent mode

The `-f1` parameter must always precede a response-file name

The parameter `<responsefile>` is the fully-qualified path to your response file.

If this parameter is omitted, the response file must be called `mqsiumin.rsp` and be located in the root directory.

Like the installation response file, the uninstallation response file consists of stanzas and keyword=value pairs.

- Lines beginning with a hash (#) are comments
- All text is in English

The MQSeries Integrator stanza has two required keywords, these are MQSeries Integrator and Locked Files.

### **MQSeries Integrator stanza**

The MQSeries Integrator stanza can take the following options:

#### **REMOVEDATA**

This causes uninstallation to remove all of MQSeries Integrator, including any data files you have created, on the target machine.

#### **REMOVE**

This causes uninstallation to remove MQSeries Integrator on the target machine, but leaves the data files.

In the preceding cases the [Components] stanza is not referenced and can be omitted.

#### **REMOVECOMPONENTS**

This causes removal of specified components. The components are listed in the [Components] stanza, which must be present.

In the [Components] stanza, there can be one keyword-value pair for each MQSeries Integrator component.

The keyword is the component name and the value must be KEEP or REMOVE.

The component names are the same strings used in an installation response file. Any installed components for which there is no keyword-value pair are not uninstalled.

The value KEEP is used when you do not want to remove a component.

The value REMOVE is used when you want to remove a component.

#### **Locked Files**

Before deleting files from the target computer, uninstallation checks to see if any of the files are locked. This parameter specifies what uninstallation does if any locked files are found.

#### **CONTINUE**

Ignore the locked files and continue. You must remove any locked files manually after uninstallation.

#### **CANCEL**

Terminate the uninstallation before deleting any files.

This is a sample of the response file that you will have to create in order for Silent Uninstall to execute.

## Uninstalling MQSeries Integrator

The following instructions will uninstall MQSeries Integrator Version 2 by component.

### Notes:

1. The uninstall will end if any locked files are found.
2. The Broker and NEON support components will not be removed.

```
[MQSeries Integrator]
MQSeries Integrator=REMOVECOMPONENTS
Locked Files=CANCEL
[Components]
Broker=KEEP
User Name Server=REMOVE
Configuration Manager=REMOVE
Control Center=REMOVE
Online Documentation=REMOVE
Samples and SDK=REMOVE
Neon Support=KEEP
```

---

## Contacting your IBM Support Center

If you are unable to resolve problems that you find when you use MQSeries Integrator, or if you are directed to do so by an error message generated by MQSeries Integrator, you can request assistance from your IBM support center.

Before you contact them, use the checklist below to gather key information. Some items may not necessarily be relevant in every situation. But you should provide as much information as possible to enable the IBM support center to recreate your problem.

- For MQSeries Integrator:
  - CSDs applied.
  - E-fixes applied.
  - All current trace and error logs, including relevant Windows NT Event log entries. User trace log files at debug level should be obtained for all relevant message flows and should preferably be formatted.
  - A list of the components installed. This should include details of the number of machines and their operating systems, the number of brokers and the machine on which they are running, and the existence and details of any User Name Servers.
  - The file obtained by exporting your workspace. This action is performed from the Control Center; see *MQSeries Integrator Using the Control Center* for details of how to do this.
  - The files obtained by exporting all relevant message sets. This action is performed for each message set by using the **mqsimrmimpexp** command with the -e flag set.

## Uninstalling MQSeries Integrator

- A sample of the messages being used when the problem arose.
- If relevant, the report file from the C or COBOL importer. This is located in the directory from which the file import was attempted.
- For MQSeries:
  - CSDs applied.
  - E-fixes applied.
  - All current trace and error logs, including relevant Windows NT Event log entries and First Failure Support Technology™ (FFST™) output files. You can find these files, which have the extension FDC, in the errors subdirectory within the MQSeries home directory.
  - Details of MQSeries client software, if appropriate.
- For each database you are using:
  - Product and release level (for example, DB2 6.1).
  - CSDs applied.
  - E-fixes applied.
  - All current trace and error logs, including relevant Windows NT Event log entries and First Failure Support Technology (FFST) output files. Check database product documentation for where to find these files.
- For Windows NT:
  - Version.
  - Service Pack level.
  - The version of the system files `msvcrt.dll`, `msvcp60.dll`, `msvcirt.dll`, and `mfc42.dll`. You can find these files in the `WINNT\SYSTEM32` directory. Use the Windows NT Explorer file properties to display the versions.
- Details of the operation you were performing, the results that occurred, and the results you were expecting.

## Uninstalling MQSeries Integrator

---

## Appendix C. Applying maintenance

Maintenance updates are supplied on CD in the form of a Program Temporary Fix (PTF), referred to as a Corrective Service Diskette (CSD). You can find the latest information about available CSDs on the Internet, at the address given in "MQSeries information available on the Internet" on page xiii.

You can also download CSDs from this web site.

---

### Applying the maintenance information

If you need to apply maintenance updates to MQSeries Integrator Version 2.0.1:

1. Ensure that the user ID you are logged on with is a member of the Windows NT **Administrators** group in the local security domain. You are recommended to use the same ID as the one you used for installing MQSeries Integrator.
2. Ensure that all MQSeries Integrator function is stopped. You must:
  - a. Stop execution of the Configuration Manager, the User Name Server, and all brokers on this system using the **mqsistop** command, or by stopping the Windows NT services for these components from the Services program in the Control Panel.
  - b. Stop execution of all programs that manage MQSeries Integrator Version 1.1 rules and formats (the Visual Tester, the NEONFormatter user interface, and the NEONRules user interface).
  - c. Stop execution of the Control Center.
  - d. Ensure that you do not have any MQSeries Integrator Version 2.0.1 files open.
3. You are also strongly recommended to exit all Windows NT programs before applying maintenance to MQSeries Integrator.
4. If you have downloaded maintenance from the web, you must:
  - a. Read the memo.ptf file associated with this maintenance on the Web site. This might contain additional information regarding the installation of maintenance.
  - b. Select a suitable destination as the target for download.
  - c. When the file has been downloaded, change to the download directory and run the executable file.

## Applying maintenance

You must choose a temporary directory to extract the maintenance files from the downloaded file. You are presented with a dialog where you can accept the default, or specify an alternative.

- d. Click **Next** to activate the extract.
  - e. Click **Finish** to end the dialog.
  - f. The `setup.exe` file now runs.
5. If you are installing from CD you must:
- a. Read the file `memo.ptf` in the root directory of the CD, and any `Readme.txt` files also in the root directory. These files might contain additional information about how you must install this maintenance.
  - b. Insert the maintenance CD into the appropriate drive. If you have autorun enabled, `setup.exe` will start to run automatically. If you do not, you must start `setup.exe` from the CD by doubleclicking.
6. When `setup.exe` runs, you are presented with a window that provides instructions for applying service. These include a checkbox that controls the backup of files replaced during the application of this service level. If you take the default action to replace these files (and you are recommended to do so), you can specify the directory in which the backup files are created. Click **Next** when you have made the updates you want. A confirmation window displays your choices. Click **Next** to continue, or **Back** if you want to return and make further changes.
7. The maintenance is applied to MQSeries Integrator.
- If you click **Cancel** before the maintenance process has completed, or if there is a power or system failure, the product files may be left in an unknown state. To restore these files (providing you took the default action to backup replaced files in step 6), use Windows NT Explorer to copy the files in the backup directory (`CSDbkup\1` for CSD#1, unless you chose a different location) to their original locations. Click **Yes** to replace existing files. When all the files have been copied back, you should remove the backup directory so that it can be recreated by a later install. You can now rerun the CSD installation.
8. When you have completed installation, review the `memo.ptf` file supplied as part of the CSD. Complete any manual post-install actions required.

**Note:** CSDs are cumulative, therefore you do not need to apply CSD1 before you can apply CSD2. When you have installed a CSD, you are prevented from installing a previous CSD without first restoring the system using the backed-up files.



### Restoring a previous service level

You are able to restore a previous service level if you want to, or if you are instructed to do so by your IBM Support Center. However, this is only possible if you accepted the default action to backup replaced files when you applied the current service level (see the instructions in “Applying the maintenance information” on page 109).

If you need to restore MQSeries Integrator to a previous level of maintenance:

1. Ensure that the user ID you are logged on with is a member of the Windows NT **Administrators** group in the local security domain. You are recommended to use the same ID as the one you used for installing MQSeries Integrator.
2. Access the Windows NT Control Panel (*Start->Settings->Control Panel*).
3. Double-click the Add/Remove Programs icon.
4. Find and highlight *IBM MQSeries Integrator V2.0.1*, and click **Add/Remove**. This starts the MQSeries Integrator uninstall program, and presents the uninstall dialog (see “Appendix B. Uninstalling MQSeries Integrator” on page 99 for further details of uninstallation).
5. If one or more CSDs have been applied the dialog allows you to select the option to uninstall CSDs. Select the option to uninstall the most recently installed CSD and return to the previous service level. If you want to remove all CSDs, you must invoke this option for each one in turn to return to the base service level.

If no CSDs have been applied, this option is unavailable.

---

### Installing new components after applying maintenance

If you install maintenance and then decide to install a new component from the MQSeries Integrator CD, you will be advised that you must reapply the maintenance following installation of the new component. You are strongly recommended to do this: if you do not, it is likely that the integrity of the product files will be compromised and unexpected results might ensue.

---

### Checking the service level

After initial installation, the MQSeries Integrator Service level indicates no service has been applied.

After one or more updates, the service level is updated to show the CSD most recently applied. The service level is expressed in terms of the PTF number for a particular CSD.

To check the service level, select *Start->Programs->IBM MQSeries Integrator 2.0.1->Service Level*. This displays the memo.ptf file

## Checking the service level

containing the service level and details of the maintenance applied (PTF Number). This file is installed in the program file directory.

---

## Applying maintenance to IBM DB2 Universal Database

If DB2 was installed on this system by the MQSeries Integrator installation program, it is installed with no service applied.

You can also obtain information about the current status of maintenance of this product, and download fix packs for DB2 from the Web site identified in "DB2 publications" on page xii.

---

## Appendix D. Using NEONFormatter and NEONRules nodes

You need to take some specific actions to use the NEONFormatter and NEONRules nodes after you have installed MQSeries Integrator Version 2.0.1 and its prerequisites (listed in “Software requirements” on page 7).

This appendix leads you through some of those actions and then directs you to specific NEONFormatter and NEONRules information in the MQSeries Integrator Version 1.1 documentation set for further details. The MQSeries Integrator Version 1.1 documentation is provided in the \Book subdirectory of the MQSeries Integrator Version 2.0.1 home directory. The default home directory is

`\Program Files\IBM MQSeries Integrator 2.0.1`

The actions you need to take are:

- Setting up the database environment
- Setting up tablespaces in the MQSeries Integrator database
- Configuring your database (DB2, Oracle7, and Oracle8)
- User-defined segments (Sybase and SQL Server 6.5)
- Setting up the server (Sybase and SQL Server 6.5)
- Installing the database schema (all databases)
- Editing the database connection file (sqlsvses.cfg)
- Migrating rules and formats (migrating from Version 1.0 to Version 2)

These actions are described in the sections that follow. Note that the examples given in this appendix describe a configuration that has a NEON Graphical User Interface (GUI) running on Windows NT and a Runtime NEON database running on UNIX.

---

### Setting up the database environment

Use the following information to check the environment for the database that you are using is set up correctly.

#### DB2

Verify the following:

- You can connect to an appropriately resourced DB2 database that stores MQSeries Integrator data, either directly or through a DB2 client.
- A DSN (DB2 instance) is defined, using either the ODBC Administrator tool or Client Configuration Assistant, to point to the DB2 database instance.
- The DB2 utility program `db2.exe` is in the execution path for the user doing the install.

## Setting up the database environment

- If you do not have DB2 for Windows NT, be sure database home, lib, and bin directories are in the PATH environment variable.

### Oracle

Verify the following:

- You can connect to an appropriately resourced Oracle database that stores MQSeries Integrator data, either directly or through an Oracle client.
- You know the Oracle SYS account information.
- The ORACLE\_HOME environment variable is set to the location of the database home.
- The PATH environment variable includes the product bin directory and the database bin directory.

### SQL Server 6.5

Verify the following:

- You can connect to an appropriately resourced SQL Server database that stores MQSeries Integrator data, either directly or through a SQL Server client.
- You are a database owner (or have the account information for the owner of the database).
- As database owner, that the target database is the default database.
- There is sufficient disk space for your calculated needs.
- The Microsoft SQL Server utility program isql is in the execution path for the user doing the install.
- The PATH environment variable includes the product bin directory and the database bin directory.

### Sybase

On the Windows NT workstation, verify that Sybase Client, Version 11.1.1 is installed. On the Sybase server, verify the following:

- You can connect to an appropriately resourced Sybase database that stores MQSeries Integrator data, either directly or through a Sybase client.
- You are a database owner or know the account information for the owner of the database.
- There is sufficient disk space for your calculated needs.
- The PATH environment variable includes the product bin directory and the database bin directory.
- The Sybase utility program isql is in the execution path for the user doing the install.

### Collecting information

Before beginning the installation, ensure you know the drive letter of the CD-ROM drive from which you will run the installation and the information for the appropriate operating system in the following sections:

#### DB2

- database alias
- username
- password

#### Oracle

- SYS userid
- password for SYS userid
- service name

#### Microsoft SQL Server

- database name
- server name
- username
- password

#### Sybase

- database name
- server name
- username
- password

---

### Schema installation

To prepare the database for database schema installation for Oracle and DB2, refer to the Creating Tablespaces section below. For Sybase and SQL Server, refer to the User-Defined Segments section on page “User-Defined Segments (for Sybase and SQL Server 6.5)” on page 128.

### Creating tablespaces

The database must have MQSeries Integrator tablespaces created before you can install the database schema. You do not have to create tablespaces with Microsoft SQL Server or Sybase Adaptive Server.

**Note:** The size of your tablespaces depends on the numbers of Rules and Formats used at your site. You might want to place the tablespaces on different physical disks to balance I/O to avoid disk-access bottlenecks. You should separate data tablespaces and index segments by placing them on different disks and/or controllers. This optimizes index and data access parallelism.

## DB2 tablespaces

---

### DB2 tablespaces

For information on creating a DB2 database, refer to the DB2 installation documentation.

To create DB2 tablespaces:

1. Create the DB2 database for MQSeries Integrator
2. Create the following tablespaces in the database:
  - FORMATTER\_DATA
  - FORMATTER\_INDEX
  - RULES\_DATA
  - RULES\_INDEX
3. Grant DBADM privilege on the database to the user who will perform the installation.

**Note:** If you are using DMS tablespaces, use the Oracle Minimum Size guidelines (as shown in the table in step 2 below) to create minimum DB2 tablespaces.

### Configuring DB2

There are three steps required in the configuration of DB2 before MQSeries Integrator can be installed. These are:

1. Create a new database to contain the rules and formats.
2. Configure a client connection to the new database.
3. Create tablespaces within the database for the MQSeries Integrator tables.

The first step must be done using the DB2 instance owning user (db2inst1). The other two steps can be done from a client machine (Windows NT). Once logged on as DB2 user on a UNIX machine, you must set up the .profile file to include the correct paths etc. To set this up, type the following:

```
$ /home/<db2 instance name>/sql1lib/db2profile
```

On Windows NT, there is a command line program that sets up the environment. This can be found in:

```
Start->Programs->DB2 for Windows NT->Command Window
```

There is also a program called Command Line Processor in the same directory that runs the DB2 command line required in steps 1 and 3 on a Windows NT machine.

1. To create a database, log on to the server as a DB2 database instance user and run the db2 command. At the command line, issue the following command:  

```
db2=>create database <database name>
```

where <database name> is the name of the MQSeries Integrator database that is required. Use the quit command to exit the DB2 command line program:

```
db2=>quit
```

2. To connect to a DB2 database, there must be a database alias set up. DB2 automatically creates an alias to the database on the server when it creates the database. This alias has the same name as that of the database. To connect to the database via a Windows NT client, an alias needs to be made on the client machine. If the server is already on the Windows NT machine, a new database alias must still be made because the default one does not have an ODBC driver associated with it. An ODBC driver is required to connect to the database using the MQSeries Integrator user interface. This can be done using the DB2 program called Client Configuration Assistant found in:

Start->Programs->DB2 for Windows NT

Run the program and do the following steps:

- a. Click the Add database button.
- b. A window called Add Database SmartGuide appears. Select the 'Manually configure a connection...' option and click Next.
- c. If you are connecting to a local Windows NT DB2 database:
  - 1) Select Local for the protocol and click Next
  - 2) Make sure that the database is on the same drive as shown, then click on Next
- d. If you are trying to connect to a remote server:
  - 1) Select TCPIP for the protocol and click Next
  - 2) Type name of the host machine where the database instance is running and the port on which the instance is listening. The port number is in the /etc/services directory. Use the command:

```
$ cat /etc/services | grep db2
```

to display all the ports used by different DB2 instances.
- e. Type the name of the database and click Next.
- f. Change the alias if you want a different name than that of the database and click Next.
- g. The last screen should have a tick in the register database as an ODBC source, and the system data source should be selected.
- h. Click Done to create the service and ODBC driver. It asks whether to test the connection.
- i. Click Test Connection.
- j. Enter the db2admin as user and its associated password. You are then successfully connected.

## DB2 tablespaces

- Using a Windows NT machine with DB2 client installed, create a database alias as outlined above in step (1). Start the DB2 command line processor found in:

Start->Programs->DB2 for Windows NT

Issue the following commands:

```
db2=>connect to <data base alias>  
      user <username> using <password>
```

Where <data base name> is the alias name made above, <username> is an authorized user, and <password> is the password.

```
db2=>create tablespace <tablespace name>  
      managed by system using  
      ('<tablespace name>')
```

where <table space name> is the name of the tablespace to be made. For example, the following line shows how to create one of the tablespaces required by MQSeries Integrator:

```
db2=>create tablespace FORMATTER_INDEX managed by system using  
      ('FORMATTER_INDEX')
```

Repeat this for all required tablespaces.

Use the quit command to exit the DB2 command line program:

```
db2=>quit
```

Now the database is ready for the installation of the Formatter and Rules tables (see *Installing Database Schema* in the *MQSeries Integrator Version 1.1 Installation and Configuration Guide*). After these are created, the MQSeries Integrator GUIs can connect to the database using the ODBC option in the DBMS section.

Now go to “Installing the database schema (all databases)” on page 133.

---

## Oracle tablespaces

For information on creating an Oracle database, refer to the Oracle installation documentation.



To create Oracle tablespaces:

1. Create a dedicated Oracle instance where the MQSeries Integrator database resides. NEONET is the default Oracle instance name used in the inst\_db.cmd file.
2. Create the following tablespaces in the Oracle database:

Table	Minimum size
TOOLS	1 MB
TEMP	10 MB
FORMATTER_DATA	20 MB
FORMATTER_INDEX	20 MB
RULES_DATA	20 MB
RULES_INDEX	20 MB

### Configuring Oracle7

There are three steps in the configuration of Oracle7 before MQSeries Integrator can be installed. These are:

1. Create a new database to contain the rules and formats.
2. Configure a client connection to the new database.
3. Create tablespaces within the database for the MQSeries Integrator tables.

These three steps must be performed using the 'oracle' user account that belongs to the dba group.

#### Step 1: creating the database

Use one of the following methods:

1. Ask the DBA to create a new database with SID of MQSeries Integrator
2. On the UNIX machine on which the Oracle7 server is installed, run the oraInst program:

```
$ <ORACLE_HOME>/oraInst/oraInst
```

and choose options to create a new database. This works only if the Oracle installer is installed. The oraInst program is in the \$ORACLE\_HOME/oraInst directory.

3. Use the default database that is created when Oracle is installed. This is usually called sid1 or sid.
4. Create a database manually. This is beyond the scope of this document, but is explained in detail in the Oracle7 online documentation.

## Oracle tablespaces

For UNIX, after the database is created, add the following line to `/etc/oratab` if it does not already exist:

```
<SID of data base>:<Oracle home directory>:Y
```

If it already exists, ensure the end character is Y. If it is N, change it to Y. To start the database execute:

```
$ dbstart
```

To stop the database (and all other databases on the machine) execute:

```
$ dbshut
```

### Step 2: configuring a client connection

The two configuration files that Oracle uses to connect a client to a server are `tnsnames.ora` and `listener.ora`. The `tnsnames.ora` file is found in the `<ORACLE_HOME>/network/admin` directory.

- On HP-UX and AIX<sup>®</sup> systems, `tnsnames.ora` file might have been placed in the `/etc` directory. This file is used to define service names. This definition links a service name to the host name of a server, the SID of a database on that server, and the port on which the listener on the server is listening for client connections.
- On the Windows NT version of Oracle7 client, is a utility called SQL Net Easy (Start->Oracle for Windows NT). This modifies the `tnsnames.ora` file found in the `<oracle_home>\network\admin` directory.

The best way to create the file on UNIX is to use SQL Net Easy on Windows NT, and then copy the `tnsnames.ora` file across to the `<ORACLE_HOME>/network/admin` directory on the UNIX machine using `ftp`.

Here is an example of a `tnsnames.ora` entry:

```
#  
# An example tnsnames.ora file  
#  
  
<Service name> =  
(DESCRIPTION =  
  (ADDRESS =  
    (COMMUNITY = tcpcom.world)  
    (PROTOCOL = TCP)  
    (HOST = <Machine name>)  
    (PORT = <Port number>)  
  )  
  (CONNECT_DATA =  
    (SID = <SID of database>)  
  )  
)
```

```
#
# End of tnsnames.ora file
#
```

| Where <Machine name> is replaced by the name of the server machine,  
| <Port number> is the port on which the listener is listening for connections,  
| and <SID of database> is the SID of the database that the service connects to.  
| <Service name> can be any name, but must be unique within the  
| tnsnames.ora file.

The listener.ora file contains the information required for the system to start listener processes that allow clients to connect to the database. This must contain the port address where the listener listens for client connections, the SID of the database, and the address of the Oracle home directory. This file should be created when a database is created by the installer; if not, the file must be configured manually. To check if any listeners are running, run the Oracle command:

```
$ <oracle_home>/bin/lsnrctl status
```

If there is no listener running for your database, add the following lines to the listener.ora file:

```
#
# Sample listener.ora file
#
# To start listener run: lsnrctl start mylsnr
# To stop listener run: lsnrctl stop mylsnr
# To check whether the listener is working run: lsnrctl status
mylsnr
#
mylsnr=
  (ADDRESS_LIST =
    (ADDRESS= (PROTOCOL= IPC)(KEY= <Service name>))
    (ADDRESS= (PROTOCOL= TCP)(Host= <Machine name>))
  (Port= <Port number>))
  )

SID_LIST_mylsnr=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=<SID of data base>)
      (ORACLE_HOME=<Oracle home directory>)
      (PRESPAWN_MAX=10)
    )
  )

STARTUP_WAIT_TIME_mylsnr = 0
CONNECT_TIMEOUT_mylsnr = 10
TRACE_LEVEL_mylsn = OFF
```

## Oracle tablespaces

```
#  
# End of listener.ora file  
#
```

Where <SID of database>, <Machine name>, and <Port number> correspond to the same entries as in tnsnames.ora. The word mylsnr can be changed to any word except LISTENER. The listener can now be started using the command:

```
| <oracle_home>/bin/lsnrctl start mylsnr
```

and stopped with the command:

```
| <oracle_home>/bin/lsnrctl stop mylsnr
```

There should now be a tnsnames.ora file on the client machine and a listener running on the server machine. Copy the tnsnames.ora file to the server so that you can connect to the database while on the server. To test the connection, run the svrmgrm utility (found in the <Oracle home>/bin directory). Enter the value system as the user name and its password and the service name created above. If it says:

```
ORA-12154: TNS: could not resolve service name
```

try the following:

1. Verify the tnsnames.ora is in the correct directory
2. Verify the correct service name is in the tnsnames.ora file
3. Verify there are no missing brackets or other typing errors

If the service name is resolved, but when the client attempts to connect to the database, the following error is returned:

```
ORA-12203: TNS: unable to connect to destination
```

try the following:

1. Verify the database is running (see section one above)
2. Verify the listener is running on the same port as the tnsnames.ora service and that the listener has the correct SID entry (run: lsnrctl status mylsnr)

If the connection is successful, the database is now configured for client connections from any machine that has the tnsnames.ora entry given above. This should be on the systems that MQSeries Integrator server and the MQSeries Integrator GUI client are installed on.

### Step 3: creating tablespaces

To create tablespaces on Windows NT, use either the sqlplus plus33 command line or the svrmgrm utility. On Windows NT Oracle 7.3.3 and later, the utility Storage Manager that can be used instead of svrmgrm.

For sqlplus, issue the following commands:

```

| $ <oracle_home>/bin/sqlplus
|   Enter user-name:sys
|   Enter password:<sys_password>
|   SQL>connect sys/<sys_password>@<Service name>
|   SQL>create tablespace FORMATTER_DATA datafile 'FORMATTER_DATA.one'
|         size 20M;
|   SQL>create tablespace FORMATTER_INDEX datafile 'FORMATTER_INDEX.one'
|         size 20M;
|   SQL>create tablespace RULES_DATA datafile 'RULES_DATA.one'
|         size 20M;
|   SQL>create tablespace RULES_INDEX datafile 'RULES_INDEX.one'
|         size 20M;
|   SQL>create tablespace TOOLS datafile 'TOOLS.one'
|         size 10M;
|   SQL>alter tablespace TEMP add datafile 'TEMP_INCREASE_SIZE.one'
|         size 1M;
|   SQL>quit

```

For the svrmgrm utility, do the following:

- ```

| $ <oracle_home>/bin/svrmgrm

```
1. Enter the sys user, sys password, service name and select connect as NORMAL
  2. Select tablespace folder
  3. Click on the tablespace menu and select Create Tablespace
  4. In the name box add the tablespace name, for example:  
FORMATTER\_DATA
  5. Click New next to the Datafiles box
  6. Enter a name for the data file, for example:  
FORMATTER\_DATA.one
  7. Put 20 in new file size and select MB from the side menu
  8. Click OK to create the file
  9. Finally click Create to make the tablespace

Repeat steps 1 to 9 for FORMATTER\_INDEX, RULES\_DATA, and RULES\_INDEX.

Repeat steps 5 to 9 for a data file TEMP using a filesize of 10 MB.

Repeat steps 5 to 9 for a data file TOOLS using a filesize of 1 MB.

The database is now ready for the MQSeries Integrator install script called `inst_db.sh` (`inst_db.bat` on Windows NT). The use of this is described in the *MQSeries Integrator Version 1.1 Installation and Configuration Guide*.

Now go to “Installing the database schema (all databases)” on page 133.

## Oracle tablespaces

### Configuring Oracle8

There are three steps in the configuration of Oracle8 before MQSeries Integrator can be installed. These are:

1. Create a new database to contain the rules and formats.
2. Configure a client connection to the new database.
3. Create tablespaces within the database for the MQSeries Integrator tables.

These three steps must be performed using the 'oracle' user account that belongs to the dba group.

#### Step 1: creating the database

Use one of the following methods:

1. Ask the DBA to create a new database with SID of MQSI
2. On the UNIX machine on which the Oracle8 server is installed, run the orainst program:

```
$ <ORACLE_HOME>/orainst/orainst
```

and choose options to create a new database. This works only if the Oracle installer is installed. The orainst program is in the \$ORACLE\_HOME/orainst directory.

3. Use the default database that is created when Oracle is installed. This is usually called sid1, (ORCL on Windows NT).
4. On Windows NT, run the Oracle Database Assistant, found in:  
Start->Programs->Oracle for Windows NT
5. Create a database manually. This is beyond the scope of this document, but is explained in detail in the Oracle8 online documentation.

For UNIX, after the database is created, add the following line to /etc/oratab if it does not already exist:

```
<SID of data base>:<Oracle home directory>:Y
```

If it already exists, ensure the end character is Y. If it is N, change it to Y. To start the database execute:

```
$ dbstart
```

To stop the database (and all other databases on the machine) execute:

```
$ dbshut
```

#### Step 2: configuring a client connection

The two configuration files that Oracle uses to connect a client to a server are tnsnames.ora and listener.ora.

- On UNIX the tnsnames.ora file is found in the <ORACLE\_HOME>/network/admin directory, and on Windows NT the tnsnames.ora is found in the <ORACLE\_HOME>/net80/admin directory.

On HP-UX and AIX systems, `tnsnames.ora` might have been placed in the `/etc` directory. This file is used to define service names. This definition links a service name to the host name of a server, the SID of a database on that server, and the port on which the listener on the server is listening for client connections.

- On the Windows NT version of Oracle8 client, there is a utility program called Oracle Net8 Assistant (Start->Oracle for Windows NT). This modifies the `tnsnames.ora` file that can be found in the `<oracle_home>\network\admin` directory.

The best way to create the file on UNIX is to use Oracle Net8 Assistant on Windows NT, and then copy the `tnsnames.ora` file across to the `<ORACLE_HOME>/network/admin` directory on the UNIX machine using ftp.

Here is an example of a `tnsnames.ora` entry:

```
#
# An example tnsnames.ora file
#

<Service name> =
  (DESCRIPTION =
    (ADDRESS =
      (COMMUNITY = tcpcom.world)
      (PROTOCOL = TCP)
      (HOST = <Machine name>)
      (PORT = <Port number>)
    )
    (CONNECT_DATA =
      (SID = <SID of database>)
    )
  )

#
# End of tnsnames.ora file
#
```

where `<Machine name>` should be replaced by the name of the server machine, `<Port number>` is the port on which the listener is listening for connections and `<SID of database>` is the SID of the database that the service connects to. `<Service name>` can be any name but must be unique within the `tnsnames.ora` file.

The `listener.ora` file contains the information required for the system to start listener processes that allow clients to connect to the database. This must contain the port address where the listener listens for client connections, the SID of the database, and the address of the Oracle home directory. This file should be created when a database is created by the installer; if not, the file must be configured manually. To find if any listeners are running run the Oracle command:

## Oracle tablespaces

```
| <Oracle_home>/bin/lsnrctl status
```

If there is no listener running for your database, add the following lines to the listener.ora file:

```
#
# Sample listener.ora file
#
# To start listener run: lsnrctl start mylsnr
# To stop listener run: lsnrctl stop mylsnr
# To check whether the listener is working run: lsnrctl status
myslnr
#

myslnr=
  (ADDRESS_LIST =
    (ADDRESS= (PROTOCOL= IPC)(KEY= <Service name>))
    (ADDRESS= (PROTOCOL= TCP)(Host= <Machine name>))
  (Port= <Port number>))
)

SID_LIST_myslnr=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=<SID of data base>)
      (ORACLE_HOME=<Oracle home directory>)
      (PRESPAWN_MAX=10)
    )
  )
)

STARTUP_WAIT_TIME_myslnr = 0
CONNECT_TIMEOUT_myslnr = 10
TRACE_LEVEL_myslnr = OFF

#
# End of listener.ora file
#
```

Where <SID of database>, <Machine name>, and <Port number> correspond to the same entries as in tnsnames.ora. The word mylsnr can be changed to any word except LISTENER. The listener can now be started using the command:

```
| <oracle_home>/bin/lsnrctl start mylsnr
```

and stopped with the command:

```
| <oracle_home>/bin/lsnrctl stop mylsnr
```

There should now be a tnsnames.ora file on the client machine and a listener running on the server machine. Also copy the tnsnames.ora file to the server so that you can connect to the database while on the server. To test the



connection, run the sqlplus utility (found in the <Oracle home>/bin directory). Enter the value system as user name followed by its password. At the command prompt, type:

```
sqlplus>connect sys/<sys password>@<service name>
```

If the following message is returned:

```
ORA-12154: TNS: could not resolve service name
```

try the following:

1. Verify the tnsnames.ora is in the correct directory
2. Verify the correct service name is in the tnsnames.ora file
3. Verify There are no missing brackets or other typing errors

If the service name is resolved but when the client attempts to connect to the database, the following error is returned:

```
ORA-12203: TNS: unable to connect to destination
```

try the following:

1. Verify the database is running (see step one above)
2. Verify the listener is running on the same port as the tnsnames.ora service and that the listener has the correct SID entry (run: lsnrctl status mylsnr)

If the connection is successful, the database is now configured for client connections from any machine that has the tnsnames.ora entry given above. This should be on the systems that MQSeries Integrator server and the MQSeries Integrator GUI client are installed on.

### Step 3: creating table spaces

To create tablespaces on Windows NT, use either the sqlplus plus80 command line or the Oracle Storage Manager utility.

For the sqlplus, issue the following commands:

```
| $ <oracle_home>/bin/sqlplus
|   Enter user-name:sys
|   Enter password:<sys_password>
|   SQL>connect sys/<sys_password>@<Service name>
|   SQL>create tablespace FORMATTER_DATA datafile 'FORMATTER_DATA.one'
|           size 20M;
|   SQL>create tablespace FORMATTER_INDEX datafile 'FORMATTER_INDEX.one'
|           size 20M;
|   SQL>create tablespace RULES_DATA datafile 'RULES_DATA.one'
|           size 20M;
|   SQL>create tablespace RULES_INDEX datafile 'RULES_INDEX.one'
|           size 20M;
|   SQL>create tablespace TEMP datafile 'TEMP.one'
```

## Oracle tablespaces

```
|      size 10M;  
|      SQL>create tablespace TOOLS datafile 'TOOLS.one'  
|      size 1M;  
|      SQL>quit
```

For the Oracle Storage Manager, do the following:

1. Enter the sys user, sys password, service name and select connect as NORMAL
2. Select tablespace folder
3. Click on the tablespace menu and select Create Tablespace
4. In the name box add the tablespace name, for example:  
FORMATTER\_DATA
5. Click Add next to the Datafiles box
6. Enter a name for the data file, for example:  
FORMATTER\_DATA.one
7. Put 20 in new file size and select MB from the side menu
8. Click OK to create the file
9. Finally click Create to make the tablespace

Repeat steps 1 to 9 for FORMATTER\_INDEX, RULES\_DATA, and RULES\_INDEX.

Repeat steps 1 to 9 for a data file TEMP using a filesize of 10 MB.

Repeat steps 1 to 9 for a data file TOOLS using a filesize of 1 MB.

The database is now ready for the MQSeries Integrator install script called `inst_db.sh` (`inst_db.bat` on Windows NT). The use of this is described in the *MQSeries Integrator Version 1.1 Installation and Configuration Guide*.

Now go to “Installing the database schema (all databases)” on page 133.

---

## User-Defined Segments (for Sybase and SQL Server 6.5)

Both Sybase and Microsoft SQL Server databases must have user-defined segments created before you can install the database schema. User-defined segments provide a mapping from the database tables and indexes to the underlying disk space on which the database resides.

You might want to place the user-defined segments on different physical disks to balance I/O and avoid disk-access bottlenecks. You should separate data and index segments by placing them on different disks, or controllers, or both. This optimizes index and data access parallelism.

To create user-defined segments:

Create the following user-defined segments in the database that will contain MQSeries Integrator tables and stored procedures:

- FORMATTER\_DATA
- FORMATTER\_INDEX
- RULES\_DATA
- RULES\_INDEX

The commands necessary to perform this are in the following vendor documentation: *SQL Server Reference Manual* for Sybase

The commands are:

- sp\_addsegment
- sp\_dropsegment
- sp\_extendsegment

*Microsoft Transact SQL Reference, System Stored Procedures* for SQL Server

The commands are:

- sp\_addsegment
- sp\_dropsegment
- sp\_extendsegment

---

## Setting up the server

Use the following information to set up your server.

### Sybase

There are six pieces of information you need to access your Sybase Server database from MQSeries Integrator. These are:

1. User ID
2. User password
3. Server name
4. Server communication protocol (for example, TCP)
5. Server address
6. Database name

The first step is to create a database for MQSeries Integrator to use. This can be done using the `isql` command line utility, either on the machine on which Sybase Server is running, or remotely via a Sybase Client.

To start `isql`, type the following command:

```
isql -U<username> -P<password> -S<servername>
```

## Setting up the server

The default servername is the name of the machine on which the server was installed. The default System Administrator ID is username "sa" with no password (enter "-P" on its own, with no subsequent password value). You are recommended to set a password, both for security reasons, and because the MQSeries Integrator Rules Engine does not accept a null password value. This can be done with the following commands:

1. `sp_password <old password>,<new password>`
2. `go`

Where `<old password>` is entered as "NULL" if no password is currently set.

You can either use this System Administrator ID to create the database and access it from MQSeries Integrator, or you can create other usernames. MQSeries Integrator requires that the username it logs on to the Sybase Server with must be the owner of the database it is intended to access. By default, the owner of a database is the user that created it.

If you encounter problems starting `isql`, the Sybase Server might not be running. If you suspect that it is not, change to the `/install` directory off the Sybase root directory, and execute the

```
"RUN_<servername>"
```

file, where `<servername>` is the name of your server.

Now create the database using the following `isql` commands:

1. `create database <database name> on <device name> = <size in MB>`
2. `go`

Unless you have a particular preference, enter "master" for `<device name>`. Ten MB should be a sufficient size for most applications of MQSeries Integrator, but if you require more precise calculation details, see the *MQSeries Integrator Version 1.1 Installation and Configuration Guide*.

For the MQSeries Integrator `inst_db.sh` script to run, you must create the user-defined segments described in the *MQSeries Integrator Version 1.1 Installation and Configuration Guide*. To do this, use the `sp_addsegment` command:

1. `sp_addsegment <segment name>,<database name>,<device name>`
2. `go`

Repeat this for all segments. Again, unless you have a preference, use "master" for the device.

The database can be accessed directly from the machine on which the Sybase Server resides, but unless you are performing a Windows NT-only installation,

you will need to connect to it remotely with the MQSeries Integrator user interfaces from a Windows NT machine.

First run the `dscp` command line utility on the Server machine, and enter the following commands:

```
>> open interfaces
>> read <servername>
```

This displays information about the server specified by `<servername>`, including the communications protocol, address, and, if appropriate, port number on which it is listening. You must know this information to configure the client.

On the Windows NT machine, run the Sybase `Dsedit` utility from the Start menu. Open the `InterfacesDriver` directory service and add a Server Object with the name and communication protocol/address of your server. Remember, if appropriate (for example, with TCP), to include the port number on which the server is listening - `<address>,<port>` is the correct format for entering this in the Network Address field. Exit the `Dsedit` utility.

You now have all the information that MQSeries Integrator requires to access its Sybase Server database. Ensure that the Sybase Server is running and continue with MQSeries Integrator installation.

### SQL Server 6.5

The four pieces of information you must know to access your Microsoft SQL Server database from MQSeries Integrator are:

1. User ID
2. User password
3. Server name
4. Database name

The default system administrator ID for MSSQL Server is "sa" with no password. You can use this ID or create another for use with MQSeries Integrator, but you are recommended to set the password, both for security reasons and because the MQSeries Integrator Rules Engine does not accept a null value for the password field. To change a user password, you can either use the Enterprise Manager, or the `isql` command line utility.

In Enterprise Manager, you must first register your server using the `Server->Register Server` menu option. The name of your Server defaults to the name of your machine, but you can call up a list of active Servers to select from. Use the default system administrator ID to register at first, then change this, if necessary, when you change passwords or create new users. To change a user password, select the `Manage->Logins` menu option, choose the required user from the drop-down list, and type the new password.

## Setting up the server

Alternatively, start isql using the following command:

```
isql -U<username> -P<password> -S<servername>
```

Using the same values for <username>, <password>, and <servername> as if you were registering your Server in Enterprise Manager. If no password is currently set, type "-P" on its own, without a subsequent value. The commands to change the logged in user password are:

1. sp\_password <old password>,<new password>
2. go

Where <old password> should be entered as "NULL" if no password is currently set.

You must now create a database for MQSeries Integrator to use. Again, this can be done either in Enterprise Manager or with the isql command line utility.

In Enterprise Manager, select the:

Manage->Databases

menu option, then create a new database. Ten MB should be large enough for most applications of MQSeries Integrator, but you might want to consult the *MQSeries Integrator Version 1 Installation and Configuration Guide* for more precise calculation details. You will have to specify a data device on which the database is to reside. Unless you have a particular preference, accept the default option of "master" provided you have made this device large enough to accommodate it.

From isql, the commands are:

1. create database "<database name>" on<device name> = <size in MB>
2. go

Where the values of <device name> and <size in MB>are as discussed above.

For the MQSeries Integrator inst\_db.sh script to run, you must create the user-defined segments described in the *MQSeries Installation and Configuration Guide*. To do this, use the sp\_addsegment command:

```
sp_addsegment <segment name>, <device name>  
go
```

Repeat this for all segments. Again, unless you have a preference, use "master" for the device.

The username that MQSeries Integrator uses to access the database must be the database owner. By default, the owner of a particular database is the user who created it. If you are going to use another username to log in from

MQSeries Integrator, you must make sure that this username is aliased to the database owner. To do this, select the Manager->Logins menu options in Enterprise Manager and select the username that MQSeries Integrator will be using to log in. Place check marks in the appropriate "permit" boxes, and enter "dbo" as an alias for those databases that the MQSeries Integrator user will be accessing.

You now have all four pieces of information required by MQSeries Integrator to access its Microsoft SQL Server database. Ensure that the Microsoft SQL Server is running and continue with MQSeries Integrator installation.

---

### Installing the database schema (all databases)

The `inst_db` script creates the necessary tables and stored procedures in the MQSeries Integrator database. The script sends the commands from the files in the `install.sql` directory.

To install the database schema on Oracle, you need the SYS user id and password.

To install the database schema:

1. At the command line prompt, change to the `install.sql` directory. For example:

```
cd c:\Program Files\IBM MQSeries Integrator 2.0.1\install.sql
```

2. To build the MQSeries Integrator schema, type one of the following:

- **DB2**

- a. Open a Windows NT Command Prompt window and type the following:

```
db2cmd
```

- b. The `db2cmd` command opens another Command Prompt window. Switch to that window and run the following command:

```
inst_db.cmd <username> <password> <database alias>
```

- **Oracle**

**Note:** If you are using Oracle8, you must set the environment variable `SQLPLUS` to plus 80. For example:

```
set SQLPLUS=Plus80
```

- Enter the following command:

```
inst_db.cmd SYS <SYS password> <Service Name>
```

**Note:** Oracle `inst_db` script creates a `MQSI110` user account in the database. This must be used when creating or modifying `NEONRules` and `NEONFormats`.

## Installing the database schema

- **SQL Server**

- Enter the following command:

```
inst_db.cmd <username> <password> <servername> <dbname>
```

**Note:** If you have not defined a password for your database username, use two single quotes to specify the password. For example:

```
inst_db.cmd sa ' ' mycomputer
```

3. As the script runs, answer the prompts and look for errors.
4. When the script completes the instantiation, a verification message appears.
5. For installation details, look at the `inst_db.log` file located in the `c:\temp` directory

**Note:** In the `inst_db.log` file, the error "table or view does not exist" does not indicate a problem with database instantiation. The database is successfully instantiated if this is the only error you receive.

---

## Editing the database connection file

Some MQSeries Integrator executables connect to the database using the database connection file `sqlsvses.cfg`. This file contains entries for DBMS sessions that detail the server name, user id, password, and database name that a particular session uses. Executables search the `sqlsvses.cfg` file for a given session name and attempt to connect to the MQSeries Integrator database (for example, `msgtst` searches for `new_format_demo`).

A sample `sqlsvses.cfg` file that is commented out is provided in the `bin` directory. Uncomment the section that applies to your DBMS type. You must edit the sample file with your site-specific information. This file enables certain NEONRules and NEONFormatter executables to connect to the database. For more information, refer to the *MQSeries Integrator Version 1.1 System Management Guide*.

To edit the database connection file:

1. Change to the `bin` directory
2. In the `bin` directory, locate the text file `sqlsvses.cfg`
3. In the `sqlsvses.cfg` file, edit the following:

### DB2

```
<sessionname>:<dbalias>:<username>:<password>:
```

For example:

```
new_format_demo:dodge:neonuser:neonpwd:  
rules:dodge:neonuser:neonpwd:
```



### Oracle

```
<sessionname>:<servicename>:<username>:<password>:
```

For example:

```
new_format_demo:dodge::neonuser:neonpwd:  
rules:dodge:neonuser:neonpwd:  
nnfie:dodge:neonuser:neonpwd:  
nnrmie:dodge:neonuser:neonpwd:
```

The default for both the Oracle username and password is NEONET.

### SQL Server

```
<sessionname>:<servername>:<username>:<password>:<dbname>
```

For example:

```
new_format_demo:dodge:neonuser:neonpwd:TESTDATABASE  
rules:dodge:neonuser:neonpwd:TESTDATABASE
```

---

## Migrating rules and formats

This section explains how to migrate the data from an MQSeries Integrator Version 1.0 database to an MQSeries Integrator Version 2.0.1 database. The NNRie and NNFie utilities are used to migrate rules and formats.

### Notes:

1. This migration procedure assumes that you created rules and formats in MQSeries Integrator Version 1.0.
2. This procedure also assumes that the database to which you are migrating rules and formats is a clean database. If you are migrating to a database that already contains data, data conflicts might occur.
3. This procedure is only required for rules and formats from Version 1.0. Rules and formats from Version 1.1 do not need to be migrated.
4. Reference is made to executable files having a suffix of 40. This is the case only if the Version 1.0 code has been removed from your machine.

## NNRie overview

Use the NEONRules Import/Export Utility, NNRie, to export existing rules from an MQSeries Integrator database and import them to an MQSeries Integrator Version 2.0.1-compatible database.

Using NNRie, you can:

- Export a single rule identified by its corresponding application name, message type, and rule name.
- Export a single subscription identified by its corresponding application name, message type, and subscription name.

## Migrating rules and formats

- Export entire rule sets, rules, and subscriptions identified by corresponding application group and message type names.
- Export all message types and their rule sets identified by the message types application group name.
- Export all application groups and their associated message types and rules.

This program creates an export file that can be interchanged between platforms. All application groups and their associated message types and rules should be exported. The exported file can then be imported to the database using NNRie.

### NNFie overview

The NEONFormatter Import/Export Utility, NNFie, is used to export existing formats from an MQSeries Integrator Version 1 database and to import the formats to an MQSeries Integrator database.

Using NNFie, you can:

- Export existing formats into a file that can be interchanged between platforms.
- Import the file into the database. NNFie can import a file created by exporting from an MQSeries Integrator formats database.

Additional information regarding NNFie is available in the *MQSeries Integrator Version 1.1 System Management Guide*.

### Migrating your rules and formats

When you migrate rules and formats to a Version 2 database, you are recommended to use two separate databases. You are not recommended to rebuild the schema after export and importing into the same database.

Verify the following before migrating rules and formats:

- There is enough disk space to hold the output file. This file can be re-directed anywhere the system supports.
- The target database has been instantiated.

The following steps are required to migrate your rules and formats from a Version 1.0 database to a Version 2 database.

1. Set up your environment
2. Check the consistency of your Version 1.0 database using the Consistency Checker version for the database from which you are exporting
3. Export your data from the Version 1.0 database
4. Modify your environment for import
5. Import your formats into the Version 2 database

6. Check the consistency of your Version 2 database using the Consistency Checker

### Setting up the environment

#### The sqlsvses.cfg file

Create the sqlsvses.cfg file. This file is used by the import and export utilities to create a session with the database server. Place this file in a the bin directory where the NNFIe and NNRIe utilities are located.

**Note:** One sqlsvses.cfg can be created and used for both NNFIe, which migrates formats, and NNRIe, which migrates rules.

#### Oracle

The sqlsvses.cfg file must contain:

```
nnfie:<dbms instance>:<user>:<password>:  
nnrmie:<dbms instance>:<user>:<password>:
```

#### Sybase and SQL Server

The sqlsvses.cfg file must contain:

```
nnfie:<dbms instance>:<user>:<password>:<database>  
nnrmie:<dbms instance>:<user>:<password>:<database>
```

#### DB2

The sqlsvses.cfg file must contain:

```
nnfie:<database name or alias>:<user>:<password>:  
nnrmie:<database name or alias>:<user>:<password>:
```

#### Environment variables

**Oracle:** Set the following:

```
ORACLE_SID=<servername>  
ORACLE_HOME=<your Oracle database directory>
```

**Sybase and SQL Server:** Set the following:

```
SYBASE=<your sybase database directory>
```

**DB2:** Set the following:

```
DB2INSTANCE=<your DB2 instance name>
```

### Checking consistency

For NEONRules and NEONFormatter, run the Consistency Checker that matches the version of the database from which you are exporting. The following examples use the suffix 110 to match a Version 1.10 or Version 2.0.1 database. Use the suffix 40 to match a Version 1.0 database. You might want to direct the output to a file, as shown in the examples below, rather than standard output because the output from the scripts could be substantial.

## Migrating rules and formats

### Oracle

- **UNIX**

```
formatcc110.ksh <user> <password> <instance> >formatcc110.log  
rulecc110.ksh <user> <password> <instance> >rulescc110.log
```

- **Windows NT**

```
formatcc110.cmd <user> <password> <instance> >formatcc110.log  
rulecc110.cmd <user> <password> <instance> >rulescc110.log
```

### Sybase and SQL Server

- **UNIX**

```
formatcc110.ksh <user> <password> <dbms instance> <database>  
>formatcc110.log  
rulescc110.ksh <user> <password> <dbms instance> <database> >rulescc110.log
```

- **Windows NT**

```
formatcc110.cmd <user> <password> <dbms instance> <database>  
>rulescc110.log  
rulescc110.cmd <user> <password> <dbms instance> <database> >rulecc110.log
```

### DB2

- **UNIX**

```
formatcc110.ksh <user> <password> <database name or alias>  
>formatcc110.log  
rulescc110.ksh <user> <password> <database name or alias> >rulescc110.log
```

- **Windows NT**

```
formatcc110.cmd <user> <password> <database name or alias>  
>formatcc110.log  
rulescc110.cmd <user> <password> <database name or alias> >rulescc110.log
```

---

## Setting the system environment variable

The definitions of the rules and formats in the database are identified in the MQSIRuleng.mpf configuration file. Every broker in which you deploy a message flow that includes the NEONRules or NEONFormatter message processing nodes requires access to these definitions.

Since these definitions are not distributed through your broker network in the same way as the formats and rules defined by the MQSeries Integrator Version 2.0.1 Control Center, you must check that:

- The system on which the broker is installed has client access to the system on which the database is installed.
- The file identified in the MQSI\_PARAMETERS\_FILE environment variable contains the correct information to connect to the database. See the *MQSeries Integrator Version 1.1 System Management Guide* for more details about the contents of this file.

---

### Exporting rules and formats

To export formats from your 1.0 database, run NNFie and to export rules, run NNRie, using the version that matches the version of the database from which you are exporting.

The data is exported to files named NEONet.fie and NEONet.rie.

DBMS platforms, MQSeries Integrator Version 1.0 database::

```
NNFie10 -e NEONet.fie -s nnfie  
NNRie10 -e NEONet.rie -s nnrie
```

---

### Modifying your environment for import

For all DBMS platforms, modify your sqlsvses.cfg file to change the instance, username, password, and database (for Sybase and SQL Server) parameters to reflect the MQSeries Integrator Version 2.0.1 database into which you are importing rules and formats.

## Importing rules and formats

---

### Importing rules and formats into your Version 2 database

To import formats run NNFie Version 2 and to import rules, run NNRie Version 2. Make sure you have instantiated the Version 2 database before you attempt to import rules and formats.

- **UNIX**

```
NNFie -i NEONet.fie -s nnfie  
NNRie -i NEONet.rie -s nnrie
```

- **Windows NT**

```
NNFie.cmd -i NEONet.fie -s nnfie  
NNRie.exe -i NEONet.rie -s nnrie
```

---

### Using NEONRules and NEONFormatter

When you have completed these actions, follow this checklist to fully use the functions of the NEONRules and NEONFormatter nodes.

- Test your installation by starting the NEONFormatter user interface. To do this, select *Start->Programs->IBM MQSeries Integrator 2.0.1->MQSIV1.1>NEONFormatter*.

Follow the information that is described in Chapter 3 "Formatter" in the *MQSeries Integrator Version 1.1 User's Guide*. to define the function you require.

- You can then use the NEONRules user interface. To do this, click *Start->Programs->IBM MQSeries Integrator 2.0.1->MQSIV1->NEONRules*.

Follow the information that is described in Chapter 4 "Rules" in the *MQSeries Integrator Version 1.1 User's Guide* to define the function you require.

- You can then use the NEON Visual Tester to verify your NEONFormatter and NEONRules definitions.

Follow the information that is described in Chapter 5 "Visual Tester" in the *MQSeries Integrator Version 1.1 User's Guide*. (You can ignore the information about recaching Rules/Formatter in the section called "General Options".)

- Use the NEONFormatter functions to parse separate input messages into individual fields and to transform input messages into an output message with a different format.

Follow the information that is described in Chapter 3 "Formatter" in the *MQSeries Integrator Version 1.1 System Management Guide*. (You can ignore the information in the section called "Shared Libraries/DLLs".)

- Use the NEONRules functions to evaluate messages, based on your chosen criteria.

Follow the information that is described in Chapter 4 "Rules" in the *MQSeries Integrator Version 1.1 System Management Guide*. (You can ignore the information in the section called "Shared Libraries/DLLs".)

---

## Appendix E. Notices

This information was developed for products and services offered in the United States. IBM may not offer the products, services, or features discussed in this information 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 information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

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

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

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

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make

## Notices

improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

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

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

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

IBM United Kingdom Laboratories,  
Mail Point 151,  
Hursley Park,  
Winchester,  
Hampshire,  
England  
SO21 2JN.

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 Programming License Agreement, or any equivalent agreement between us.

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.



---

## Trademarks

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

|                        |                                  |
|------------------------|----------------------------------|
| AIX                    | DB2                              |
| DB2 Universal Database | First Failure Support Technology |
| FFST                   | IBM                              |
| MQSeries               | SupportPac                       |
| VisualAge              |                                  |

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

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

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

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



---

## Glossary of terms and abbreviations

This glossary defines MQSeries® Integrator terms and abbreviations used in this book. If you do not find the term you are looking for, see the index or the *IBM® Dictionary of Computing*, New York: McGraw-Hill, 1994.

This glossary includes terms and definitions from the *American National Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute. Copies may be ordered from the American National Standards Institute, 11 West 42 Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.

### A

**Access Control List (ACL).** The list of principals that have explicit permissions (to publish, to subscribe to, and to request persistent delivery of a publication message) against a topic in the topic tree. The ACLs define the implementation of topic-based security.

**ACL.** Access Control List.

**AMI.** Application Messaging Interface.

**Application Messaging Interface (AMI).** The programming interface provided by MQSeries that defines a high level interface to message queuing services. See also *MQI* and *JMS*.

### B

**blob.** Binary Large OBject. A block of bytes of data (for example, the body of a message) that has no discernible meaning, but is treated as one solid entity that cannot be interpreted. Also written as BLOB.

**broker.** See *message broker*.

**broker domain.** A collection of brokers that share a common configuration, together with the single Configuration Manager that controls them.

### C

**callback function.** See *implementation function*.

**category.** An optional grouping of messages that are related in some way. For example, messages that relate to a particular application.

**check in.** The Control Center action that stores a new or updated resource in the configuration or message repository.

**check out.** The Control Center action that extracts and locks a resource from the configuration or message repository for local modification by a user. Resources from the two repositories can only be worked on when they are checked out by an authorized user, but can be viewed (read only) without being checked out.

**collective.** A hyperconnected (totally connected) set of brokers forming part of a multi-broker network for publish/subscribe applications.

**configuration.** In the broker domain, the brokers, execution groups, message flows and message sets assigned to them, topics and access control specifications.

**Configuration Manager.** A component of MQSeries Integrator that acts as the interface between the configuration repository and an executing set of brokers. It provides brokers with their initial configuration, and updates them with any subsequent changes. It maintains the broker domain configuration.

**configuration repository.** Persistent storage for broker configuration and topology definition.

**connector.** See *message processing node connector*.

## Glossary

**content-based filter.** An expression that is applied to the content of a message to determine how the message is to be processed.

**context tag.** A tag that is applied to an element within a message to enable that element to be treated differently in different contexts. For example, an element could be mandatory in one context and optional in another.

**Control Center.** The graphical interface that provides facilities for defining, configuring, deploying, and monitoring resources of the MQSeries Integrator network.

## D

**datagram.** The simplest form of message that MQSeries supports. Also known as *send-and-forget*. This type of message does not require a reply. Compare with *request/reply*.

**deploy.** Make operational the configuration and topology of the broker domain.

**destination list.** A list of internal and external destinations to which a message is sent. These can be nodes within a message flow (for example, when using the RouteToLabel and Label nodes) or MQSeries queues (when the list is examined by an MQOutput node to determine the final target for the message).

**distribution list.** A list of MQSeries queues to which a message can be put using a single statement.

**Document Type Definition.** The rules that specify the structure for a particular class of SGML or XML documents. The DTD defines the structure with elements, attributes, and notations, and it establishes constraints for how each element, attribute, and notation can be used within the particular class of documents. A DTD is analogous to a database schema in that the DTD completely describes the structure for a particular markup language.

**DTD.** Document Type Definition

## E

**e-business.** A term describing the commercial use of the Internet and World Wide Web to conduct business (short for electronic-business).

**element.** A unit of data within a message that has business meaning, for example, street name

**element qualifier.** See *context tag*.

**ESQL.** Extended SQL. A specialized set of SQL statements based on regular SQL, but extended with statements that provide specialized functions unique to MQSeries Integrator.

**exception list.** A list of exceptions that have been generated during the processing of a message, with supporting information.

**execution group.** A named grouping of message flows that have been assigned to a broker. The broker is guaranteed to enforce some degree of isolation between message flows in distinct execution groups by ensuring that they execute in separate address spaces, or as unique processes.

**Extensible Markup Language (XML).** A W3C standard for the representation of data.

## F

**filter.** An expression that is applied to the content of a message to determine how the message is to be processed.

**format.** A format defines the internal structure of a message, in terms of the fields and order of those fields. A format can be self-defining, in which case the message is interpreted dynamically when read.

## G

**graphical user interface (GUI).** An interface to a software product that is graphical rather than textual. It refers to window-based operational characteristics.

## I

**implementation function.** Function written by a third-party developer for a plug-in node or parser. Also known as a *callback function*.

**input node.** A message flow node that represents a source of messages for the message flow.

**installation mode.** The installation mode can be Full, Custom, or Broker only. The mode defines the components of the product installed by the installation process on Windows NT® systems.

## J

**Java™ Database Connectivity (JDBC).** An application programming interface that has the same characteristics as **ODBC** but is specifically designed for use by Java database applications.

**Java Development Kit (JDK).** A software package that can be used to write, compile, debug, and run Java applets and applications.

**Java Message Service (JMS).** An application programming interface that provides Java language functions for handling messages.

**Java Runtime Environment.** A subset of the Java Development Kit (JDK) that contains the core executables and files that constitute the standard Java platform. The JRE includes the Java Virtual Machine, core classes and supporting files.

**JDBC™.** Java Database Connectivity.

**JDK™.** Java Development Kit.

**JMS.** Java Message Service. See also *AMI* and *MQI*.

**JRE.** Java Runtime Environment.

## L

**local error log.** A generic term that refers to the logs to which MQSeries Integrator writes records on the local system. On Windows NT, this is the

Event log. On UNIX® systems, this is the syslog. See also *system log*. Note that MQSeries records many events in the log that are not errors, but information about events that occur during operation, for example, successful deployment of a configuration.

## M

**message broker.** A set of execution processes hosting one or more message flows.

**messages.** Entities exchanged between a broker and its clients.

**message dictionary.** A repository for (predefined) message type specifications.

**message domain.** The source of a message definition. For example, a domain of MRM identifies messages defined using the Control Center, a domain of NEON identifies messages created using the NEON user interfaces.

**message flow.** A directed graph that represents the set of activities performed on a message or event as it passes through a broker. A message flow consists of a set of message processing nodes and message processing node connectors.

**message flow component.** See *message flow*.

**message parser.** A program that interprets a message bitstream.

**message processing node.** A node in the message flow, representing a well defined processing stage. A message processing node can be one of several primitive types or can represent a subflow.

**message processing node connector.** An entity that connects the output terminal of one message processing node to the input terminal of another. A message processing node connector represents the flow of control and data between two message flow nodes.

**message queue interface (MQI).** The programming interface provided by MQSeries queue managers. The programming interface

## Glossary

allows application programs to access message queuing services. See also *AMI* and *JMS*.

**message repository.** A database holding message template definitions.

**message set.** A grouping of related messages.

**message template.** A named and managed entity that represents the format of a particular message. Message templates represent a business asset of an organization.

**message type.** The logical structure of the data within a message. For example, the number and location of character strings.

**metadata.** Data that describes the characteristic of stored data.

**MQI.** Message queue interface.

**MQRFH.** An architected message header that is used to provide metadata for the processing of a message. This header is supported by MQSeries Publish/Subscribe.

**MQRFH2.** An extended version of MQRFH, providing enhanced function in message processing.

**multilevel wildcard.** A wildcard that can be specified in subscriptions to match any number of levels in a topic.

## N

**node.** See *message processing node*.

## O

**ODBC.** Open Database Connectivity.

**Open Database Connectivity.** A standard application programming interface (API) for accessing data in both relational and non-relational database management systems. Using this API, database applications can access data stored in database management systems on a variety of computers even if each database management system uses a different data storage

format and programming interface. ODBC is based on the call level interface (CLI) specification of the X/Open SQL Access Group.

**output node.** A message processing node that represents a point at which messages flow out of the message flow.

## P

**plug-in.** An extension to the broker, written by a third-party developer, to provide a new message processing node or message parser in addition to those supplied with the product. See also *implementation function* and *utility function*.

**point-to-point.** Style of messaging application in which the sending application knows the destination of the message. Compare with *publish/subscribe*.

**POSIX.** Portable Operating System Interface For Computer Environments. An IEEE standard for computer operating systems (for example, AIX® and Sun Solaris).

**predefined message.** A message with a structure that is defined before the message is created or referenced. Compare with *self-defining message*.

**primitive.** A message processing node that is supplied with the product.

**principal.** An individual user ID (for example, a log-in ID) or a group. A group can contain individual user IDs and other groups, to the level of nesting supported by the underlying facility.

**property.** One of a set of characteristics that define the values and behaviors of objects in the Control Center. For example, message processing nodes and deployed message flows have properties.

**publication node.** An end point of a specific path through a message flow to which a client application subscribes. A publication node has an attribute, subscription point. If this is not

specified, the publication node represents the default subscription point for the message flow.

**publish/subscribe.** Style of messaging application in which the providers of information (publishers) are decoupled from the consumers of that information (subscribers) using a broker. Compare with *point-to-point*. See also *topic*.

**publisher.** An application that makes information about a specified topic available to a broker in a publish/subscribe system.

## Q

**queue.** An MQSeries object. Message queuing applications can put messages on, and get messages from, a queue. A queue is owned and maintained by a queue manager. Local queues can contain a list of messages waiting to be processed. Queues of other types cannot contain messages: they point to other queues, or can be used as models for dynamic queues.

**queue manager.** A system program that provides queuing services to applications. It provides an application programming interface (the MQI) so that programs can access messages on the queues that the queue manager owns.

## R

**retained publication.** A published message that is kept at the broker for propagation to clients that subscribe at some point in the future.

**request/reply.** Type of messaging application in which a request message is used to request a reply from another application. Compare with *datagram*.

**rule.** A rule is a definition of a process, or set of processes, applied to a message on receipt by the broker. Rules are defined on a message format basis, so any message of a particular format will be subjected to the same set of rules.

## S

**self-defining message.** A message that defines its structure within its content. For example, a message coded in XML is self-defining. Compare with *pre-defined message*.

**send and forget.** See *datagram*.

**setup type.** The definition of the type of installation requested on Windows NT systems. This can be one of **Full**, **Broker only**, or **Custom**.

**shared.** All configuration data that is shared by users of the Control Center. This data is not operational until it has been deployed.

**signature.** The definition of the external characteristics of a message processing node.

**single-level wildcard.** A wildcard that can be specified in subscriptions to match a single level in a topic.

**subscriber.** An application that requests information about a specified topic from a publish/subscribe broker.

**subscription.** Information held within a publication node, that records the details of a subscriber application, including the identity of the queue on which that subscriber wants to receive relevant publications.

**subscription filter.** A predicate that specifies a subset of messages to be delivered to a particular subscriber.

**subscription point.** An attribute of a publication node that differentiates it from other publication nodes on the same message flow and therefore represents a specific path through the message flow. An unnamed publication node (that is, one without a specific subscription point) is known as the default publication node.

**system log.** A generic term used in the MQSeries Integrator messages (BIPxxx) that refers to the local error logs to which records are written on the local system. On Windows NT, this is the Event log. On UNIX systems, this is the syslog. See also *local error log*.

## Glossary

### T

**terminal.** The point at which one node in a message flow is connected to another node. Terminals enable you to control the route that a message takes, depending whether the operation performed by a node on that message is successful.

**topic.** A character string that describes the nature of the data that is being published in a publish/subscribe system.

**topic based subscription.** A subscription specified by a subscribing application that includes a topic for filtering of publications.

**topic security.** The use of ACLs applied to one or more topics to control subscriber access to published messages.

**topology.** In the broker domain, the brokers, collectives, and connections between them.

**transform.** A defined way in which a message of one format is converted into one or more messages of another format.

### U

**Uniform Resource Identifier.** The generic set of all names and addresses that refer to World Wide Web resources.

**Uniform Resource Locator.** A specific form of URI that identifies the address of an item on the World Wide Web. It includes the protocol followed by the fully qualified domain name (sometimes called the host name) and the request. The Web server typically maps the request portion of the URL to a path and file name. Also known as Universal Resource Locator.

**URI.** Uniform Resource Identifier

**URL.** Uniform Resource Locator

**User Name Server.** The MQSeries Integrator component that interfaces with operating system facilities to determine valid users and groups.

**utility function.** Function provided by MQSeries Integrator for the benefit of third-party developers writing plug-in nodes or parsers.

### W

**warehouse.** A persistent, historical datastore for events (or messages). The **Warehouse** node within a message flow supports the recording of information in a database for subsequent retrieval and processing by other applications.

**wildcard.** A character that can be specified in subscriptions to match a range of topics. See also *multilevel wildcard* and *single-level wildcard*.

**wire format.** This describes the physical representation of a message within the bit-stream.

**W3C.** World Wide Web Consortium. An international industry consortium set up to develop common protocols to promote evolution and interoperability of the World Wide Web.

### X

**XML.** Extensible Markup Language.



---

# Index

## A

Adobe Acrobat Reader xi, 16  
applications 4  
    AMI 4  
    MQI 4  
    point-to-point 4  
    publish/subscribe 4  
applying maintenance 109

## B

broker 1, 13  
    changing 80  
    creating 56  
    MQSeries Integrator Version 2.0,  
        migration 24  
    MQSeries resources 85  
    restarting 87  
    starting 61  
    stopping 80  
broker domain 41  
    assigning message set 66  
    assumptions 44, 79  
    checking configuration 66  
    complex configuration 79  
    configuring 52  
    creating the broker 56  
    creating the Configuration  
        Manager 52  
    creating the User Name  
        Server 82  
    database authorizations 50  
    database setup 48  
    deploying configuration 66  
    environment variables 90  
    extending 80  
    MQSeries objects 95  
    saving configuration 66  
    simple configuration 41  
    starting 59, 85  
    starting channels 86  
    starting listeners 86  
    starting MQSeries listener 59  
    starting the broker 61  
    starting the Configuration  
        Manager 60  
    starting the Control Center 62  
    starting User Name Server 87  
    stopping the broker 80

broker domain 41 (*continued*)  
    stopping the Configuration  
        Manager 80  
    stopping the Control Center 80  
    system updates 89  
    user IDs 46

## C

CMRMUUID, Configuration  
    Manager, migration 25  
Configuration Manager 1, 13  
    changing 80  
    creating 52  
    MQSeries Integrator Version 2.0,  
        migration 24  
    restarting 87  
    starting 60  
    stopping 80  
contacting IBM 106  
Control Center 3, 14  
    initializing connection 62  
    MQSeries Integrator Version 2.0,  
        migration 23  
    restarting 87  
    starting 62  
    stopping 80

## D

database  
    authorizations 51  
    broker tables 58, 92  
    configuration repository 55, 93  
    connecting 48  
    creating 48  
    initializing 48  
    message repository 94  
    ODBC connection 49  
    tables 92  
database summary 10  
DB2  
    applying maintenance 112  
    creating the broker database 45  
    license agreement 9  
    NLS 11  
    publications xii  
    using NEONFormatter and  
        NEONRules 113, 115, 133  
delivery media 27  
directory structure 89  
disk space required 6

## E

environment variables 90

## G

getting started 41

## I

installation  
    after installation 33  
    application programming 10  
    before you start 28  
    classpath 10  
    connectivity 9  
    Control Center hardware  
        requirements 6  
    database 8  
    database disk space 6  
    delivery media 27  
    disk space requirements 6  
    error logs 34  
    errors 34  
    general hardware  
        requirements 5  
    Java Plugin 9  
    LAN 33  
    license agreement 11  
    Lotus Media 9  
    MQSeries 7  
    MQSeries Integrator Tour 9  
    MQSeries Integrator Version 1  
        databases 9  
    operating system software 7  
    preparation 28  
    prerequisites 28  
    procedure 29  
    product registration 32  
    Readme.txt 32  
    security 17  
    shared CD drive 33  
    shared LAN drive 33  
    software options 9  
    software requirements 7  
    summary of component  
        installation options 16  
    system updates 33  
    temporary disk space 7  
    verification 63  
installing on Windows NT 27

- L**
  - license agreement
    - DB2 9
    - MQSeries Integrator 11
- M**
  - message flow 2
    - input node 2
    - output node 2
    - publication node 2
  - message parsers 3
  - message sets 3
    - import 64
  - messages
    - predefined 3
    - self-defining 3
  - migration
    - concerns during installation 20
    - MQSeries Integrator Version 2.0
      - broker 24
      - CMRMUUIID column
        - name 25
      - Configuration Manager 24
      - Control Center 23
    - MQSeries
      - publications xi
      - Web site xiii
    - MQSeries client
      - Java 14
    - MQSeries configuration 83
    - MQSeries Integrator Tour 3, 14
    - MQSeries Integrator Version 1 21
      - access to rules and formats 21
      - adding new rules and
        - formats 21
      - backup 22
      - logs and log records 22
      - migrating rules and formats 21
      - MQSIgetdata.mpf 22
      - MQSIputdata.mpf 22
      - MQSIRuleng.mpf 22
      - publications xi
      - sqlsvses.cfg 22
      - uninstall 22
      - user exits 22
      - using NEON nodes 113
    - MQSeries Integrator Version 1
      - publications xi
    - MQSeries Integrator Version 2.0
      - migration 23
    - MQSeries Integrator Version 2.0.1
      - PDF library xi
      - publications x
  - MQSeries objects
    - broker 58
- MQSeries objects (*continued*)
  - Configuration Manager 55
    - connecting two queue
      - managers 83
    - MQSeries channels 85
    - MQSeries listener 85
    - User Name Server 83
  - MQSeries Publish/Subscribe
    - migration 25
    - MQSeries requirements 25
    - publications xii
  - MQSeries Workflow
    - publications xii
  - MQSI\_PARAMETERS\_FILE 21, 138
  - mqsigratebroker 24
  - mqsigrateconfigmgr 24
- N**
  - national language support 11
  - NEON database
    - checking consistency 137
    - collecting information 115
    - creating tablespaces 115
    - DB2 tablespaces 116
    - editing database connection
      - file 134
    - environment setup 113, 137
    - environment variables 137
    - exporting rules and formats 139
    - importing rules and formats 140
    - installing database schema 133
    - migrating rules and formats 136
    - modifying environment for
      - import 139
    - NNFie overview 136
    - NNRie overview 135
    - Oracle tablespaces 118
    - rules and formats 135
    - schema installation 115
    - SQL Server user-defined
      - segments 128, 131
    - Sybase user-defined
      - segments 128, 129
    - using NEONFormatter 140
    - using NEONRules 140
  - nodes, new in Version 2.0.1 24
- O**
  - ODBC connection
    - defining 49
  - online documentation 16
  - Oracle
    - creating the broker database
      - using NEONFormatter and
        - NEONRules 114, 115, 133
- P**
  - PDF library xi
  - planning
    - NLS 11
  - Postcard 71
    - how it works 73
    - running 72
  - preparation
    - for installation 28
    - for uninstall 99
  - primary domain 18
  - problems with installation 34
  - product components
    - broker 13
    - checking existence 58
    - Configuration Manager 13
    - Control Center 14
    - online documentation 16
    - primary 12
    - secondary 14
    - Tour plug-ins 16
    - User Name Server 14
  - product registration 32
  - publications
    - DB2 xii
    - MQSeries xi
    - MQSeries Integrator Version
      - 1 xi
    - MQSeries Integrator Version
      - 2.0.1 x
    - MQSeries Publish/Subscribe xii
    - MQSeries Workflow xii
- R**
  - Results Service 67
    - how it works 68
    - running 68
  - running applications 67
- S**
  - Samples
    - samples 15
    - SDK 15
  - Scribble 70
    - how it works 71
    - running 70
  - SDK 15
  - secondary components
    - samples 14
    - Software Developers' Kit
      - (SDK) 14
  - security
    - domains 17
    - global groups 18, 96
    - primary domain 18

- security (*continued*)
  - principals 17, 96
  - summary of authorizations 19
  - trusted domain 18
- service
  - applying maintenance 109
  - backup files 110
  - checking service level 111
  - DB2 112
  - restoring previous level 111
- setup types 11
  - broker only 12
  - custom 12
  - full 12
- simple message flow
  - assigning 75
  - creating 74
  - deploying 76
  - MQSeries resources 74
  - testing 76
- space requirements 6
- SQL Server
  - creating the broker database 45
  - using NEONFormatter and NEONRules 114, 128, 133
- Sybase
  - creating the broker database 45
  - using NEONFormatter and NEONRules 114, 128, 133
- system setup 5
- system updates 33

## T

- Tour plug-ins 16
- trusted domain 18

## U

- uninstallation
  - backup 100
  - files preserved 100
  - files removed 99
  - manual 102
  - migration considerations 102
  - options 101
  - unattended 104
  - uninstall program 101
- uninstalling MQSeries Integrator 99
- user IDs
  - database authorizations 51
- User Name Server 14
  - creating 82
  - MQSeries resources 84
  - starting 87

## V

- verification applications
  - Postcard 71
  - Results Service 67
  - Scribble 70
- verifying installation 63
  - applications 67
  - creating MQSeries resources 63
  - message set import 64
  - preparation 63
  - simple message flow 74
  - workspace import 64

## W

- Windows NT
  - planning for installation 5
- Windows NT registry 91
  - broker 58
  - Configuration Manager 56
  - User Name Server 83
- Windows NT service
  - broker 58
  - Configuration Manager 55
  - User Name Server 83
- workspace
  - import 65



---

## 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 44-1962-870229
  - From within the U.K., use 01962-870229
- 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 the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

GC34-5600-02

