MQSeries® Integrator for Windows NT®

Installation Guide

Version 2.0.2

GC34-5600-03

MQSeries® Integrator for Windows NT®

Installation Guide

Version 2.0.2

GC34-5600-03

Note!

Before using this information and the product it supports, be sure to read the general information under "Appendix D. Notices" on page 101.

Fourth edition (April 2001)

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.

Contents

Figures

Tables

About this book

This book explains how to plan for, install, and verify an installation of MQSeries Integrator for Windows NT Version 2.0.2.

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

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

“Chapter 3. Installing MQSeries Integrator” on page 23 provides detailed installation information for MQSeries Integrator for Windows NT Version 2.0.2.

“Chapter 4. Getting Started with MQSeries Integrator” on page 37 explains how you complete product setup, using a basic 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.

“Chapter 5. A more complex configuration” on page 69 describes how to enhance the basic broker domain.

The appendixes cover the configuration established by the default installation options and details of servicing and removing the product.

A glossary and bibliography are also provided.

Who this book is for

This book is for administrators of systems on which MQSeries Integrator Version 2.0.2 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 following:

- Windows NT or Windows 2000 system facilities
- MQSeries for Windows NT or Windows 2000 administration facilities

Terms used in this book

All references to MQSeries Integrator are to MQSeries Integrator Version 2.0.2 unless otherwise stated.

All references in this book to Windows NT are also applicable to Windows® 2000 unless otherwise stated.

MQSeries Integrator components that are installed and operated on Windows NT can also be installed and operated on Windows 2000.

`mqsi_root` is used to refer to the product's home directory.

All new terms introduced in this book are defined in "Glossary of terms and abbreviations" on page 105.

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 116).
- DB2®: a general term to encompass IBM DB2 Universal Database® Enterprise Edition, Connect Enterprise Edition, and Extended Enterprise Edition.

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 (GC34–5600–03)

The changes are summarized below:

- This edition covers Windows 2000 and Windows NT, although the latter name is used throughout as a general term to cover both platforms.
- There is a new table of supported databases (Table 2 on page 10).
- Information about Microsoft Data Access Component (MDAC) Version 2.5 has been added. See "Software requirements" on page 7.
- MQSeries Integrator for Windows NT is enabled for national language support. See "National language support" on page 11.
- Information about customizing DB2 has been added. See "Customizing the database" on page 46.
- NEONRules and NEONFormatter Support for MQSeries Integrator is installed automatically if the broker, Configuration Manager or Control Center components are installed. See "Installation procedure" on page 25
- The discussion of migration issues has been removed from this book. Refer to the *MQSeries Integrator Administration Guide* for guidance on migration.
- The information on using NEONFormatter and NEONRules nodes has been moved to the *MQSeries Integrator Administration Guide*.
- Windows NT Server Version 4.0, running on an iSeries 400 Integrated xSeries Server, is mentioned as a suitable platform for MQSeries Integrator for Windows NT.

Minor editorial changes have been made throughout the book.

Changes for the previous edition (GC34–5600–02)

The changes are summarized below:

- Information on migration from MQSeries Integrator Version 1 and MQSeries Publish/Subscribe was updated. See ″Migration considerations″.

Minor editorial changes have been made throughout the book.

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 2.0.2 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 different machines (that might 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 **input** 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 **output** 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 format that are *self-defining*.

You can group related message templates 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.

The message header for predefined messages usually identifies which message template to use. This enables the message flows to 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 has comprehensive online help. It includes the *MQSeries Integrator Tour*, which provides an online overview of the MQSeries Integrator product, its components, and the Control Center interface. 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 by providing:

1. Introductory information that you can read, with links to MQSeries information on the worldwide web.
2. An 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.

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 12
   c. "Product components" on page 13
   d. "Security considerations" on page 18
   e. "Migration considerations" on page 22
2. Installation:

   When you have decided which components you want to install, follow the guidance in "Chapter 3. Installing MQSeries Integrator" on page 23. 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 37.

   A basic configuration is used to illustrate the tasks needed and the results of the steps taken. A set of basic 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 running the specified operating system, all the corresponding supporting software shown below, and any associated applications unmodified. This machine could be an Integrated xSeries Server installed in an iSeries 400® (AS/400).

- Any communications hardware supporting NetBIOS, SNA LU 6.2, SPX, and TCP/IP.
- A minimum of 512 Megabytes (MB) of RAM to support runtime 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 256MB 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	195
Configuration Manager	195
Control Center	307
User Name Server	193
Online documentation	110
Samples and SDK	4
TOTAL	675

If you install the Broker, Configuration Manager or Control Center, an installation of NEONRules and NEONFormatter Support for MQSeries Integrator is launched. This requires 75MB of disk space and is included in the TOTAL given in Table 1 above.

Each component (except Online documentation, and Samples and SDK) includes a set of common files. The values in Table 1 reflect this. If you install

all the components, the total disk space used is less than the sum of the individual components, as shown in the total value in the table. This is also true if you install more than one component.

After installation, you need to 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), additional space is required.

You also need between 150MB (for a broker or full installation) and 300MB (for a custom installation) of additional space for temporary files. 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 the MQSeries Integrator installation, you need an additional 420MB of disk space.

Software requirements

Note: Minimum supported levels are given. 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 for Microsoft Windows NT Version 4.0 is not supported.

  The installation program checks that the version of operating system installed is at least Microsoft Windows NT Version 4.0 with Service Pack 5. If it is 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 Download Center at this Web site:

  `http://support.microsoft.com/directory/`

- MQSeries for Windows NT, V5.1 or later.

  This must be at service level Corrective Service Diskette (CSD) 4 or later.[1]

  Note: MQSeries for Windows NT and Windows 2000, V5.2 is supplied with the product.

  Read the `readme.txt` file, supplied on the product CD, 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 13.

  The installation program checks that MQSeries for Windows NT, V5.1 or later is installed, and that it is at the correct service level. If you do not have the MQSeries components you need, you are recommended to install these before you continue with MQSeries Integrator installation.

  Note: MQSeries for Windows NT, V5.0 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 5.0 with Service Pack 1 or later.

    This is available from the Download Center at this Web site:

    `http://www.microsoft.com`

    Note: You require Internet Explorer Version 5 if you intend using NEONRules and NEONFormatter Support.

  – 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.

[1] If you plan to establish an integrated broker network that includes both MQSeries Integrator and MQSeries Publish/Subscribe, refer to the MQSeries Integrator Administration Guide.

- Microsoft Data Access Components (MDAC) Version 2.5

  MQSeries Integrator Version 2.0.2 uses ODBC drivers that require MDAC Version 2.5. The MQSeries Integrator Version 2.0.2 installation program checks the level of MDAC installed on your system and issues a warning if you need to update your system. MDAC Version 2.5 is provided on the supplemental CD supplied with the product. It is also available from the Microsoft Web site at `http://www.microsoft.com`.
- 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 7.1, which is provided in the MQSeries Integrator product package.

  DB2 has no additional prerequisite products, but it does require an additional 420MB 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 database are only supported using DB2.

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.
- 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™ Plug-in 1.1.2_001

    This is available on the MQSeries Integrator CD.
  - Lotus® Media

    This is available on the MQSeries Integrator CD.
- Application programming support

  The following software compilers are supported:
  - IBM C and C++ Compiler Version 3.6.4
  - IBM VisualAge® for C++ for Windows Version 4.0
  - Microsoft Visual C++ for Windows 95 and Windows NT Version 6.0
  - Micro Focus Object COBOL for Windows NT Version 3.3 or Version 4.0
  - Micro Focus COBOL Version 3.1J for Windows NT
  - IBM VisualAge COBOL for Windows NT Version 2.1
  - IBM VisualAge COBOL Enterprise Version 2.2
  - IBM VisualAge e-business for Windows Version 1.01
  - 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 Basic for Windows Version 4.0 (16 bit) or 5.0 (32 bit)

Database summary

Table 2 lists the databases supported.

Table 2. Supported databases for brokers and user data

Database	AIX	HP-UX	Sun Solaris	Windows NT	Windows 2000
DB2[1] [2]	6.1[3] 7.1[3]	7.1[3]	6.1[3] 7.1[3]	6.1[3] 7.1[3]	6.1[3] 7.1[3]
Microsoft SQL Server	not applicable	not applicable	not applicable	6.5 plus SP5a 7.0 plus SP1	7.0 plus SP1 2000
Oracle[1]	8.1.6 8.1.7	8.1.6 8.1.7	8.1.6 8.1.7	8.1.6 8.1.7	8.1.6 8.1.7
Sybase	12	not supported	12	12	12

Table 2. Supported databases for brokers and user data (continued)

Database	AIX	HP-UX	Sun Solaris	Windows NT	Windows 2000

Notes:

1. DB2 6.1 and DB2 7.1 on all supported operating systems, and Oracle 8.1.6 and Oracle 8.1.7 **on Sun Solaris only**, are the only DBMS supported by MQSeries Integrator that permit 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. You must use DB2 for the configuration and message repository databases maintained by the Configuration Manager. No other database is supported for this purpose.
3. Please check the `Readme.txt` file for your product to check if a Fixpack is required.

License information

Under the terms of the MQSeries Integrator Version 2.0.2 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 2.0.2 is enabled for national language support. The user interface and message catalogs are provided in the following languages:

- Brazilian Portuguese
- French
- German
- Italian
- Japanese
- Korean
- Simplified Chinese
- Spanish
- Traditional Chinese
- US English

MQSeries Integrator Version 2.0.2 can process and construct messages in any code page for which MQSeries supports conversion to and from Unicode, on all operating systems. Supported code pages are listed in the *MQSeries Application Programming Reference*.

National language support

Note: The NEONRules and NEONFormatter Support nodes (NEONMap, NEONRulesEvaluation, and NEONTransform, and the superseded NEONFormatter and NEONRules) and graphical user interfaces are supplied in US English only

MQSeries Integrator interacts with MQSeries installed in any supported language. All languages for the MQSeries messaging products are included on the MQSeries server CD supplied with MQSeries Integrator.

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

DB2 Version 7.1 is fully NLS-enabled and is provided in all supported languages.

For further information about changing language settings, refer to the *MQSeries Integrator Administration Guide*.

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 basic broker domain” on page 37 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 69, and refer to the *MQSeries Integrator Introduction and Planning* book for planning tasks, and the *MQSeries Integrator Administration Guide* for implementation tasks.

When you install, you select a *setup type* that defines the components to install. There are three setup types:

- **Full**

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

  The basic configuration used to illustrate post-installation configuration, activation, and testing assumes a full installation on one system.
- **Broker only**

In this type of installation, the broker is the only component that is installed. You are likely to choose the Broker only setup if you are setting up a broker domain distributed across many different physical systems.

- **Custom**

  A custom installation allows you to choose the components you want to install. You select one or more components from a list of valid choices. Selecting all components results in a full installation. You can use a Custom setup to add additional components to an already installed system.

  Note: If you choose to install a new component, or replace an existing component, after you have installed a CSD for MQSeries Integrator, a warning dialog is displayed indicating that you must re-install 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 two secondary components. A set of common files is also installed with each major component.

The dependencies of each of these components are described in the sections below.

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 37.

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.

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, Sybase 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.

NEONRules and NEONFormatter Support is installed automatically when the broker option is installed.

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.

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.

NEONRules and NEONFormatter Support is installed automatically when the Configuration Manager option is installed.

The Control Center

You can install and use the Control Center on any number of systems in your broker domain subject to your license agreement.

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.

NEONRules and NEONFormatter Support is installed automatically when the Control Center option is installed.

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

You can install two optional components:
- Samples and Software Developers’ Kit (SDK)
- Online documentation

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 sample applications, and working examples of the plug-in extensions.

- Sample applications

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

- Sample programs

  These are supplied in C and Java. Both source and executable files are supplied.

  Before you run the sample programs in your broker domain, you must ensure that MQSeries connectivity is available. You should check the details of the operating systems and application programming languages supported by MQSeries clients, and by applications that are local to queue managers.

  You can copy and modify the sample programs to create your own applications, or you can add sections of code to your existing applications to exploit MQSeries Integrator functionality.

  The verification programs **Scribble**, **Postcard**, and **Soccer** are provided to help you test your initial installation. These are described in “Running the predefined verification applications” on page 57.

  The set of programs that make up the **Soccer** 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 nodes. 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.

Online Documentation

MQSeries Integrator documentation is provided in Portable Document Format (PDF) for online viewing using Adobe Acrobat Reader.

PDF files for books that have been translated are also provided. For more details about how to access these files, see the “Bibliography” on page 113.

A searchable library in PDF, which provides a cross book index and search facility, is also provided in US English only.

You can access the documentation without installing it, from the \Docs subdirectory in the root directory of the supplemental 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 Version 2.0.2 cross-platform publications" on page 113.

MQSeries Integrator Tour

The MQSeries Integrator Tour (described in "The Control Center" on page 3), provides animated sequences that depends on the following plug-in products:

- Java Plug-in 1.1.2_001
- Lotus Media

You are recommended to install both plug-in products if you intend to use the Tour on this system. They are supplied on the MQSeries Integrator CD, and you can choose to install them after you have installed the MQSeries Integrator components.

Additional products

DB2: If a supported version of DB2 is not installed already on your system, you can choose to install DB2 Version 7.1 as part of the product installation, if you install the Broker or Configuration Manager components.

NEONRules and NEONFormatter Support: NEONRules and NEONFormatter Support is installed if you install the Broker, Control Center or the Configuration Manager components.

Tivoli® Management support: You can install the files required by the Tivoli product from the primary product CD. For details, read the supplied PDF and the readme.txt file supplied on the primary product CD.

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 3 on page 20.

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
```

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 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.

  You should define these groups exactly as they appear in the list above. For example, from Windows NT User Manager for Domains, select `Groups->New Global Group` and enter **Domain mqbrkrs** in the `Group Name` field and so on for each group.
- 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, the installation also adds each global group to the appropriate local group in the local domain.

  If you intend to install on a primary domain controller, you are recommended to install on the domain controller first, so that the domain groups are created for you. They are then added automatically to the local groups that are created during installation on a workstation.
- If you install MQSeries Integrator on a workstation member of a trusted domain, MQSeries Integrator does not recognize the trusted domain, and cannot add the global groups to the local groups. You must complete this step manually.
- 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, the installation also adds each global group to the appropriate local group in the local domain. The installation program does not detect the trusted domain and therefore cannot add the

global groups of the trusted security domain to the local groups. If you want to add the trusted security domain global groups to the local groups instead of, or in addition to, the primary security global groups, you need to do this manually.

When you define a new user ID to your security domain, you need to assign global authority to the user ID by assigning it to the appropriate domain group.

Table 3 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 3. Summary of authorizations

User is...	Local domain (SALONE)	Primary Domain (PRIMARY)	Trusted domain (TRUSTED)
Installing	• Member of **Administrators**	Not applicable.	Not applicable.
Uninstalling	• Member of **Administrators**	Not applicable.	Not applicable.
Creating broker, Configuration Manager, User Name Server	• Must be a user ID defined in SALONE • Member of **Administrators**	• Must be a user ID defined in PRIMARY • Member of SALONE\Administrators	• Must be a user ID defined in TRUSTED • Member of SALONE\Administrators
Starting broker, Configuration Manager, User Name Server	• Member of **Administrators**	Not applicable.	Not applicable.
Running User Name Server (service user ID)	• Must be a user ID defined in SALONE • Member of **mqbrkrs**	• Must be a user ID defined in PRIMARY • Member of PRIMARY\Domain mqbrkrs	• Must be a user ID defined in TRUSTED • Member of TRUSTED\Domain mqbrkrs
Running Configuration Manager (service user ID)	• Must be a user ID defined in SALONE • Member of **mqbrkrs** • Member of **mqm**	• Must be a user ID defined in PRIMARY • Member of PRIMARY\Domain mqbrkrs • Member of SALONE\mqm (see note 1)	• Must be a user ID defined in TRUSTED • Member of TRUSTED\Domain mqbrkrs • Member of SALONE\Domain mqm (see note 2)
Running broker (service user ID) (see note 5)	• Must be a user ID defined in SALONE • Member of **mqbrkrs**	• Must be a user ID defined in PRIMARY • Member of PRIMARY\Domain mqbrkrs	• Must be a user ID defined in TRUSTED • Member of TRUSTED\Domain mqbrkrs

Table 3. Summary of authorizations (continued)

User is...	Local domain (SALONE)	Primary Domain (PRIMARY)	Trusted domain (TRUSTED)
Running Control Center (see note 3)	• Must be a user ID defined in SALONE (see note 4) For example, SALONE\User1 is valid, PRIMARY\User2 and TRUSTED\User3 are not • Member of one or more of **mqbrasgn**, **mqbrdevt**, **mqbrops**, **mqbrtpic**	• Must be a user ID defined in PRIMARY (see note 4) For example, PRIMARY\User2 is valid, SALONE\User1 and TRUSTED\User3 are not. • Member of one or more of PRIMARY**Domain mqbrasgn**, PRIMARY**Domain mqbrdevt**, and so on.	• Must be a user ID defined in TRUSTED (see note 4) For example, TRUSTED\User3 is valid, SALONE\User1 and PRIMARY\User2 are not. • Member of one or more of TRUSTED**Domain mqbrasgn**, TRUSTED**Domain mqbrdevt**, and so on.
Running publish/subscribe applications	• Must be a user ID defined in SALONE For example, SALONE\User1 is valid, PRIMARY\User2 and TRUSTED\User3 are not.	• Must be a user ID defined in PRIMARY For example, PRIMARY\User2 is valid, SALONE\User1 and TRUSTED\User3 are not.	• Must be a user ID defined in TRUSTED For example, TRUSTED\User3 is valid, SALONE\User1 and PRIMARY\User2 are not.

Notes:

1. If you are using a primary domain, you can also:
   - Define the user ID in the domain PRIMARY.
   - Add this ID to the group **PRIMARY\Domain mqm**.
   - Add the **PRIMARY\Domain mqm** group to the group **SALONE\mqm**.
2. If you are using a trusted domain, you can also:
   - Define the user ID in the domain TRUSTED.
   - Add this ID to the group **TRUSTED\Domain mqm**.
   - Add the **TRUSTED\Domain mqm** group to the group **SALONE\mqm**.
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 **mqm** by MQSeries. MQSeries Integrator installation overrides this restriction and gives read access for this subdirectory to all users.
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.
5. The broker can be run as an MQSeries trusted application. If it is, security requirements are changed. See the *MQSeries Integrator Administration Guide* for full details.

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

Migration considerations

If you plan to migrate from one of the following versions of MQSeries Integrator, it is recommended that you refer to the *MQSeries Integrator Administration Guide* to fully understand the implications of migration and any actions you should take before you install MQSeries Integrator Version 2.0.2:

- MQSeries Integrator Version 1
- MQSeries Integrator Version 1.1
- MQSeries Integrator Version 2.0
- MQSeries Integrator Version 2.0.1

Chapter 3. Installing MQSeries Integrator

This chapter describes how to install MQSeries Integrator for Windows NT.

It covers the following:

- "Delivery media"
- "Preparing for installation" on page 24
- "Installation procedure" on page 25
- "System update information" on page 29
- "Setting up for LAN installation" on page 29
- "What to do if something goes wrong during installation" on page 30

Delivery media

The MQSeries Integrator for Windows NT Version 2.0.2 package includes the following:

- MQSeries Integrator for Windows NT Version 2.0.2 (primary product) CD-ROM.

  This contains:

  – MQSeries Integrator for Windows NT Version 2.0.2

  – Documentation for MQSeries Integrator for Windows NT Version 2.0.2

  – Tivoli interface support files and documentation
- MQSeries Integrator for Windows NT Version 2.0.2 'Supplement' CD-ROM.

  This includes:

  – MQSeries for Windows NT, V5.1 CSD 6

    The minimum supported level is MQSeries for Windows NT, V5.1 CSD 4. You do not need to install CSD 6.

  – The IBM DB2 Universal Database Client for Windows NT.

    The Administration and Runtime Clients are included in all available national languages.

  – Additional product service (MDAC Version 2.5).

    Read the `Readme.txt` file on the primary product CD for the latest details of the service levels required.
- MQSeries for Windows NT and Windows 2000, V5.2 CD-ROM.

  This product is provided in all available national languages.
- MQSeries Integrator for Windows NT Version 2.0.2 'DB2 for Windows 7.1' CD-ROM (English and EMEA Languages)

Delivery media

- MQSeries Integrator for Windows NT Version 2.0.2 'DB2 for Windows 7.1' CD-ROM (English and AP Languages)
- MQSeries Clients CD.

  Contains clients for all platforms in all available national languages.
- The following hardcopy installation documentation:
  - This book
  - *MQSeries for Windows NT and Windows 2000 Quick Beginnings*

For details about prerequisite products 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 you install MQSeries Integrator for Windows NT, you should read the `Readme.txt` file, which you can find in the root directory of the CD. This contains the latest information available.

You can also find the full product documentation in PDF format on the supplemental CD.

When you install MQSeries Integrator, you must log on with a user ID that has **Administrator** authority on the machine you are using. You cannot complete the installation without this authority.

You are recommended to use a logon ID of less than eight characters. If your logon ID has more than eight characters, you will see an error message (BIP8638W) 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 you start the installation. For more details, see “Using Windows NT primary or trusted security domains” on page 19.

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 29.

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.

Note: If a previous MQSeries Integrator Version 2.0 installation exists already on your system, a custom installation is launched and you will see a window with all the previously installed components selected.

Prerequisite software

You are advised to check the full details of the prerequisites for each component in “Software requirements” on page 7.

The installation program checks for the presence of the prerequisite software required by the components you have chosen to install. If a prerequisite is not found, the details of the product are displayed in a window. At this point, you are recommended to click **Cancel** to terminate the installation, and to install (or upgrade) the product (or products) listed. You may click **Next** to continue with the installation, but you must install or upgrade the products listed before you use MQSeries Integrator.

Installation procedure

The product takes between ***5*** and ***30*** minutes to install, depending on which components are installed, whether DB2 is installed, and the speed of your processor.

Installation procedure

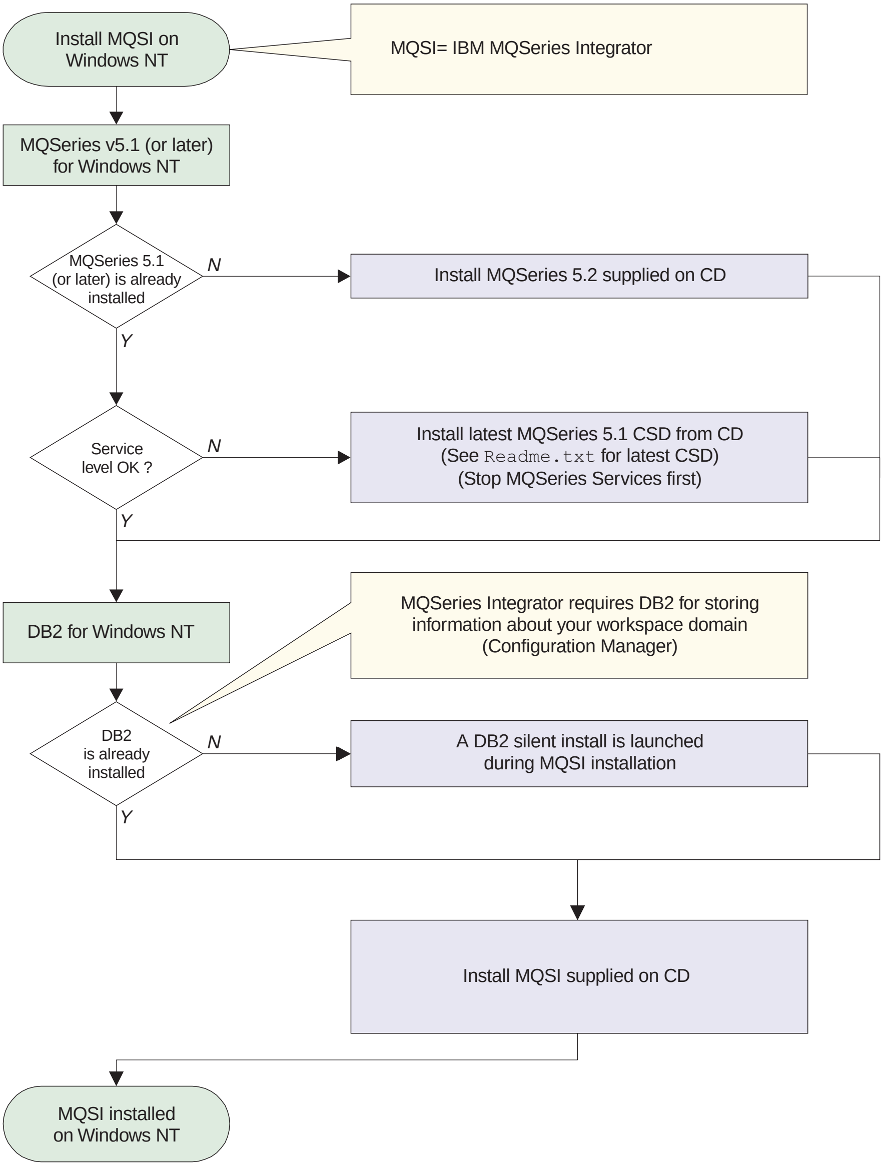

Install MQSI on Windows NT
MQSI= IBM MQSeries Integrator
MQSeries v5.1 (or later) for Windows NT
MQSeries 5.1 (or later) is already installed
N
Install MQSeries 5.2 supplied on CD
Y
Service level OK ?
N
Install latest MQSeries 5.1 CSD from CD (See Readme.txt for latest CSD) (Stop MQSeries Services first)
Y
DB2 for Windows NT
MQSeries Integrator requires DB2 for storing information about your workspace domain (Configuration Manager)
DB2 is already installed
N
A DB2 silent install is launched during MQSI installation
Y
Install MQSI supplied on CD
MQSI installed on Windows NT

Figure 1. MQSeries Integrator for Windows NT installation procedure

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 chosen to install the Broker component or the Configuration Manager component, the installation program checks for the presence of a database:
  - If a version of DB2 is not found, the window *DB2 information* is displayed. This window states the restrictions of use for this version of the DB2 product. To accept these restrictions select **Yes**. If you cannot accept these conditions, select **No**. If you select **Yes**, a silent install of DB2 will be launched later and you will be prompted to insert the appropriate CD for your language.

    You will be prompted for a user ID and a password for the DB2 administrator. DB2 has 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. If the user ID does not exist, message BIP8639W is displayed asking you to confirm that you want to create this user ID.

    You must enter a password for the user ID you have specified.

    Guidelines on choosing a user ID and password are given in the Help available from this window.

    If the checks on this user ID and password fail, you can continue with MQSeries Integrator installation, but DB2 installation might fail. If it does, correct the error (reported in message BIP8632W) and install DB2 by running `Setup.exe` from the root directory on the appropriate DB2 CD for your language.

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

    If you are installing on a backup domain controller, you will see a warning dialog with instructions for the additional steps you must take to install DB2 correctly. The dialog recommends that you terminate installation at this point, take the necessary action, and then restart the installation. If you choose to continue, DB2 installation will fail. In this case, you should take the necessary action after MQSeries Integratorinstallation, 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.
  - 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.
- If you have chosen to install the Broker, the Configuration Manager or the Control Center components, an installation of NEONRules and

NEONFormatter Support is launched after MQSeries Integrator has installed. If the NEONRules and NEONFormatter Support installation fails, you can install it later from the product CD. See “What to do if something goes wrong during installation” on page 30.

- After you have chosen the components to install, the installation program checks if you have MDAC Version 2.5 installed on your system and issues a warning if you need to upgrade your system. To install MDAC Version 2.5, you can either run the MDAC setup program (use the appropriate language version of the setup program for your installation) from the supplemental CD supplied with the product or you can download MDAC Version 2.5 from the Download Center at this Web site: `www.microsoft.com`.
- 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.

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:

- Review the release notes (`Readme.txt`). This option is selected by default.
- 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.
- Install the plug-in Lotus Media Version 2.0 Player. This is required by the MQSeries Integrator Tour. When the Select plug-in panel for Lotus Media is displayed, edit the default path for the Select plug-in folder to delete the extra `"C:`.

For details of how to complete the post-installation steps that are required, see “Chapter 4. Getting Started with MQSeries Integrator” on page 37.

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 re-shippable 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 37 for details).

System update information

The installation program updates 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 77.
- Environment variables have been created, or modified, as shown in “Environment variables” on page 78.
- The Windows NT registry has been updated to reflect your installation. A summary of the changes is shown in “Windows NT registry” on page 79.
- The principals for MQSeries Integrator security, listed in “Principals” on page 85, have been created.
- DB2 and NEONRules and NEONFormatter Support update system resources. For details refer to your DB2 and NEON documentation.

For details of how to complete the post-installation steps that are required, see “Chapter 4. Getting Started with MQSeries Integrator” on page 37.

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:

- Make the MQSeries Integrator CD-ROM drive shareable.
- Copy the product files from the CD 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.
  4. If you are installing DB2, create a new folder within the folder you created in step 1. Copy the contents of the appropriate DB2 CD (supplied in the product package) to this new folder.

Note: Each DB2 CD requires 600 MB of space.
For example:

```
md: m:\instmqsi\DB2
xcopy x:\*.*m:\instmqsi\DB2 /e
```

where e is the CD-ROM drive that contains the DB2 CD.

5. Give all licensed users access to the folder that now contains the CD image (in this example, the `m:` drive). You must also add the users to the appropriate Windows NT groups (for example, the **Administrators** group).
6. Make the drive shareable using Windows NT Explorer.
7. 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: \\mqsint\instmqsi
   ```

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

   Notes:

   a. If your shared installation directory contains spaces (for example, `mqsi install`, you must enclose it in quotes on the `net use` command.
   b. 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.
8. Change to the installation directory (in this example, `x:`). Type **Setup** and press Enter.
9. 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 product 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 116).

- 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 any corrective action to take.
- Review the MQSeries Integrator installation log, `MQSI2.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 was installed by the installation program, check the DB2 log file `mqsi2db2.log` in the `\winnt` directory. 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 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 any corrective action to take.
- If NEONRules and NEONFormatter Support installation has failed, check the NEON documentation to determine the cause of failure. To install NEONRules and NEONFormatter Support manually, run the `setup.exe` file specified in the `autorun.inf` file in the `neon52` folder on the product CD.
- If any other error is displayed, check the Windows NT online help.

When you have identified and corrected the error, run the installation program again. If the problem persists, you are advised to follow the steps for manual uninstallation (described in “Manually uninstalling MQSeries Integrator” on page 91) to ensure that your system is in a consistent state before you run the installation again.

If you have checked the possible sources of error listed above and cannot resolve the problem, contact your IBM Support Center. See “Contacting your IBM Support Center” on page 94 for further information.

Silent installation

MQSeries Integrator Version 2.0.2 supports silent installation, where no user interaction is expected. Silent installation is typically used where you need to install multiple copies of a software program on many, identical machines. Such machines may be remote to the installer.

Note: A silent installation of MQSeries Integrator Version 2.0.2 does not launch an installation of NEONRules and NEONFormatter Support. You will need to install the latter manually, if it is required. To do this, run the `setup.exe` file specified in the `autorun.inf` file in the `neon52` folder on the product CD.

Silent installation

A silent installation requires a response file. This is a text file that contains coded responses for the dialog screens that are displayed during a normal installation. It is possible to create a response file manually, but the recommended method is to **record** a response file.

The following sections describe how to record a response file and run a silent installation.

Recording a response file

Note: If you have a previous installation of MQSeries Integrator, you **must** do the following before you run `setup` to record a response file:

- Stop any brokers
- Stop the Configuration Manager and User Name Server
- Close the Control Center

To record a response file, run the `setup` command with the parameters `-noinst`, `-r` and `-f1`. For example:

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

where `x` is the target drive for the response file being recorded.

The parameters `-noinst`, `-r` and `-f1` are explained in Table 4.

Table 4. Parameters for silent installation and response files

Parameter	Explanation
-noinst	Indicates that setup will record a response file only and will not perform a real installation (that is, the registry is not updated and no files are copied). You might see a warning message telling you that a previous install has been detected and that reinstall might overwrite the configuration files. You can safely ignore this warning.
-r	Indicates that setup will record a response file.
-nomsgbox	Indicates that message-box output for warnings should be ignored. This must be placed before the -s flag.
-s	Indicates that setup is silent.

Table 4. Parameters for silent installation and response files (continued)

Parameter	Explanation
-lnnnn	Indicates a national language support message catalog, where *nnnn* is one of the following: • 0007 (German) • 0009 (Spanish) • 0010 (Italian) • 0011 (Japanese) • 0012 (Korean) • 0404 (Traditional Chinese (Taiwan)) • 0406 (French) • 0416 (Portuguese (Brazilian)) • 0804 (Simplified Chinese (PRC))
-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.

If MQSeries Integrator is not installed already on your system, you will step through all the dialog screens that you would see during a normal installation and your responses are recorded.

An example response file for MQSeries Integrator Version 2.0.2 is shown below:

```
[InstallShield Silent]
Version=v5.00.000
File=Response File
[File Transfer]
OverwriteReadOnly=NoToAll
[DlgOrder]
Dlg0=ID_DLG_WELCOME-0
Count=5
Dlg1=MbShowInfoList-0
Dlg2=SdComponentDialogEx-0
Dlg3=MbStartCopy-0
Dlg4=MbFinish-0
[ID_DLG_WELCOME-0]
Result=1
[MbShowInfoList-0]
Result=1
[SdComponentDialogEx-0]
Component-type=string
Component-count=6
Component-0=Broker
Component-1=Configuration Manager
```

```
Component-2=Control Center
Component-3=User Name Server
Component-4=Online Documentation
Component-5=Samples and SDK
RememberComponents=127
Result=1
[MbStartCopy-0]
Result=1
[MbFinish-0]
Result=1
bOpt1=0
bOpt2=0
bOpt3=0
bOpt4=0
bOpt5=0
bOpt6=0
```

This example shows a full installation where DB2 is already installed.

Running a silent installation

To run a silent installation, run the `setup.exe` file for MQSeries Integrator from a command line, using the command:

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

where `<location>` is the location of the install image and `<parameters>` are selected from those listed in Table 4 on page 32.

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

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

where `x` is the drive and directory from which the response file will be read, and `y` is the directory to which the log will be written.

Detecting errors from a silent install

To verify that a silent installation was successful, you should examine the log specified on the `-f2` flag on the setup command to check that the ResultCode=0.

The log for a successful installation should look like the example below:

```
[InstallShield Silent]
Version=v5.00.000
File=Log File
[Application]
Name=IBM MQSeries Integrator Version 2.0.2
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 installation of MQSeries Integrator Version 2.0.2 was successful. Non-zero result code values that might be returned by InstallShield are shown in Table 5.

Table 5. 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

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 is not valid on this machine. For a normal (non-silent) installation, this would not cause the installation to fail; the installation does not proceed until the user chooses a valid target directory. For a silent installation, the response file value is fixed and the silent installation fails.

The non-zero return codes that you might find in MQSI2.log are listed in Table 6.

Table 6. 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 or 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
-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 setup of a basic configuration, and shows how you can create a basic scenario to confirm that the product has been installed successfully.

The chapter has two sections:
- "Configuring a basic broker domain"
- "Verifying your installation" on page 54

Note: The steps given in this chapter are based on an example for Windows NT. If you are using Windows 2000, the steps you need to take will differ slightly from those described here. Refer to your Windows 2000 documentation for more information.

Configuring a basic broker domain

This section takes you through the tasks 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 69.

There are six tasks:
1. "Designing a basic configuration" on page 40
2. "Setting up user IDs and groups" on page 42
3. "Creating and connecting to the databases" on page 43
4. "Setting up database authorizations" on page 45
5. "Configuring your broker domain" on page 46
6. "Starting your broker domain" on page 51

Figure 2 on page 38 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 use it to check off the tasks as you complete them. You must complete all the steps in each task.

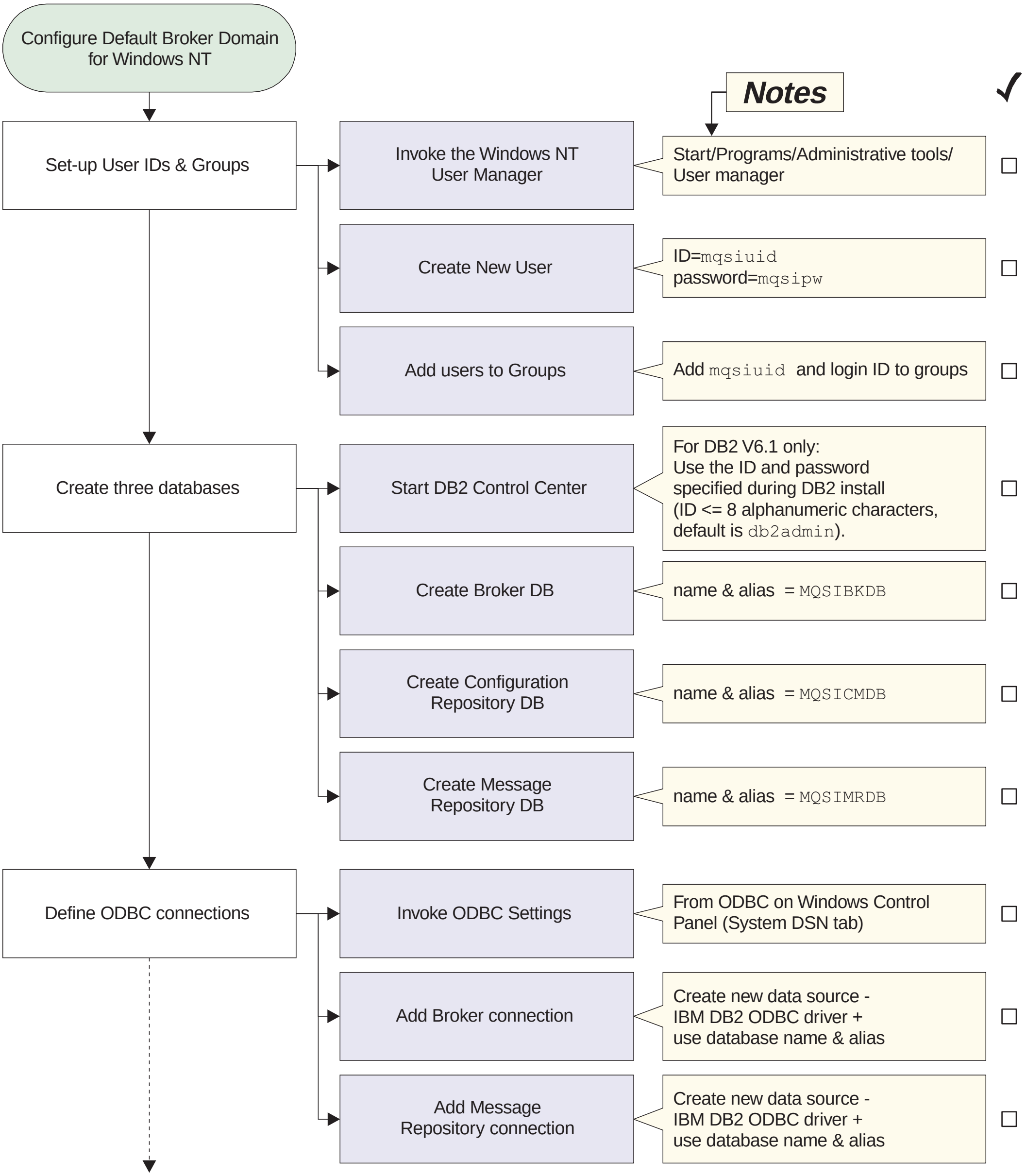
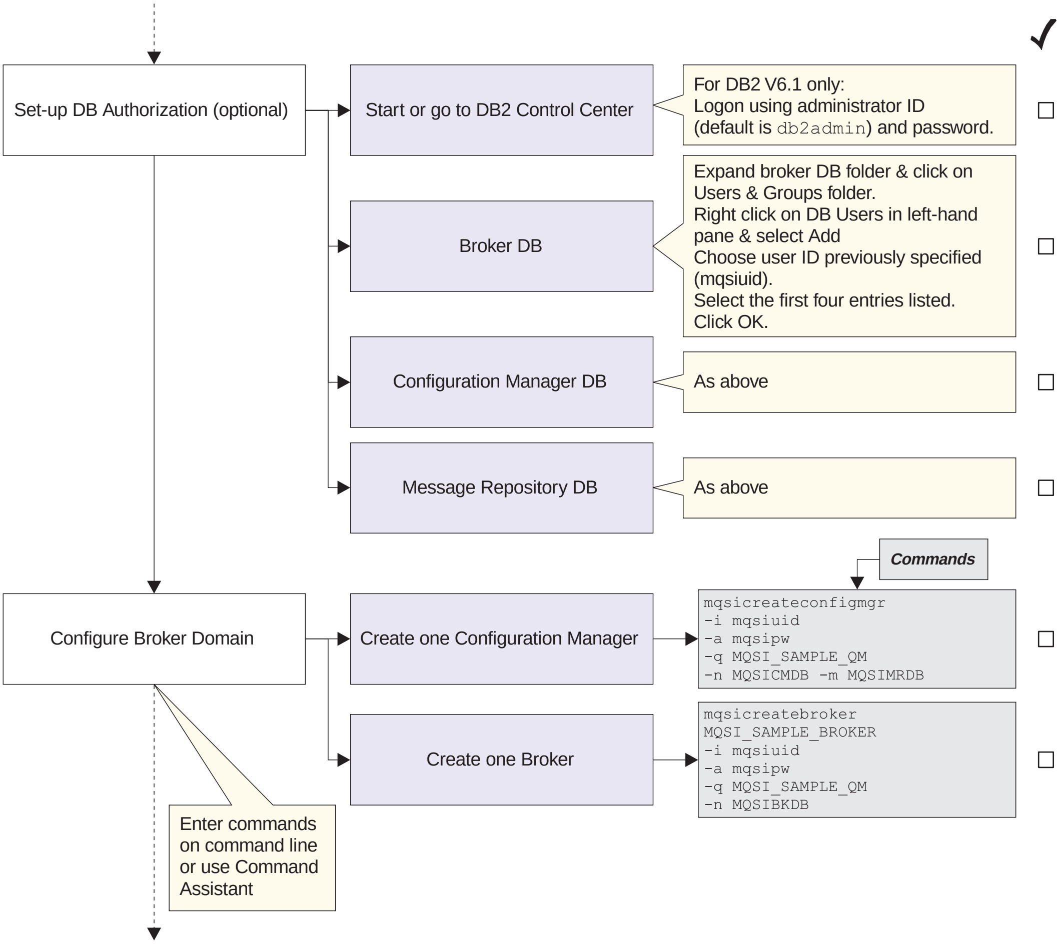


Figure 2. Configuring a basic broker domain (Part 1 of 3)



For DB2 V6.1 only:
Logon using administrator ID
(default is `db2admin`) and password.

Expand broker DB folder & click on Users & Groups folder.
Right click on DB Users in left-hand pane & select Add
Choose user ID previously specified (mqsiuid).
Select the first four entries listed.
Click OK.

As above

As above

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

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

Enter commands on command line or use Command Assistant

Figure 2. Configuring a basic broker domain (Part 2 of 3)

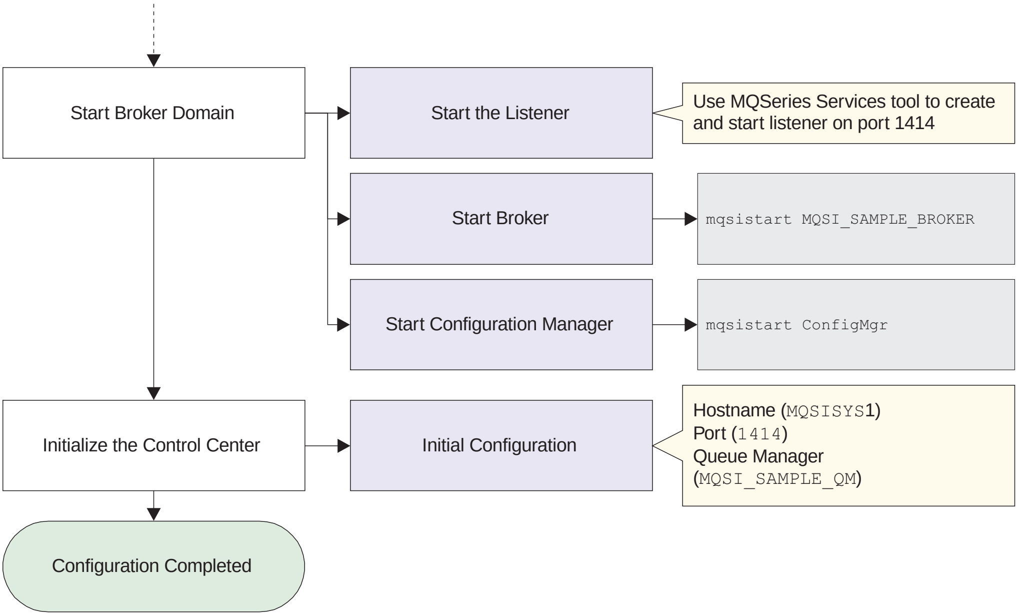


Figure 2. Configuring a basic broker domain (Part 3 of 3)

Most of these tasks 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. This guide describes how to use the Command Assistant screens and gives the commands in full. You can issue the commands using whichever method you prefer.

The Command Assistant and the configuration commands are described in the *MQSeries Integrator Administration Guide*. It also provides additional reference and guidance material, and describes the actions to take should any errors occur whilst you are completing the tasks described here.

Designing a basic configuration

Before you start to define any resources, review the assumptions made about the basic 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.

This chapter makes the following assumptions:

- You have completed a **Full** installation. All components are therefore installed, and can be created, on a single machine.
- The product is installed on a system with the TCP/IP hostname MQSISYS1. If your system has a different hostname, remember to replace MQSISYS1 with the hostname of your system in the steps in this chapter.
- The default MQSeries port 1414 is available. If port 1414 is already being used by another MQSeries queue manager, you need to assign an alternative port value.
- 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 a basic 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' (*mqsiuid*), 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*.

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 69 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

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

Note: If you choose to create a new User ID, do not give it the same name as your machine, otherwise the following error occurs: `BIP8026 Unable to start the component`.

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.

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. If you decide to use a different (new or existing) user ID, be sure to use it wherever the sample user ID is shown in the following tasks.

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 `mqsiuid`, with password `mqsipw`.

   Note that the maximum length for the `mqsiuid` field is 12 characters.

Step 3. Add the new user ID to the groups required for running the components (see Table 3 on page 20 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 **mqbrtpic**.

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.

Creating and connecting to the databases

Note

On a DB2 system, you need to bind the db2cli package to the configuration-manager database. To do this, open a DB2 Command Line Processor window and carry out the following procedure:

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

   Where `c:\` is the drive on which you installed DB2.
3. Connect and reset.

Take the following steps:

Step 1. Start the DB2 Control Center from the Start menu (*Start->Programs->IBM DB2 v7.1->Control Center*).

DB2 Version 7.1 does not prompt for a user ID and password. DB2 Version 6.1 prompts you to enter the user ID and password you specified for the DB2 Administrator when DB2 was installed. The default user ID is `db2admin`.

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 Wizard*.

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.

Enter a name and an alias for each database you create. You can accept the default values for all the remaining options. Click **Finish**. A confirmation message indicating successful completion of the create command appears at the bottom of the window.

Step 3. Define the ODBC connections.

a. From the Windows NT Start menu, select *Start->Settings->Control Panel*.

b. Within the Control Panel, double-click the ODBC icon (labelled *ODBC* or *Data Sources (ODBC)*).

c. Click the *System DSN* tab.

d. 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

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 task if you choose to specify your DB2 administrator ID and password for the data source 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.

For DB2 Version 6.1, sign on with the user ID and password you specified for “Creating and connecting to the databases” on page 43.

Step 2. For each database you created in “Creating and connecting to the databases” on page 43:

a. Expand the Object tree until you find the database. Left-click *User and Group Objects*.

b. Right-click *DB Users* in the right-hand pane and select *Add* from the popup menu. The Add User notebook opens.

c. Select the user ID `mqsiuid` (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*.

The `ServiceUserID` used for the create commands (the default is `mqsiuid`) must have the following authorities for all the databases you have created for MQSeries Integrator:

- Connect database
- Create tables
- Create packages
- Register functions to execute in database manager's process.

d. Click **OK** to grant the authority. The dialog is closed.

Step 3. Close the DB2 Control Center.

Customizing the database

DB2 Version 7.1 requires a Database Heap Size of at least 900 pages (4KB). To change the Heap Size:

Step 1. Start the DB2 Control Center.

Step 2. For each database you have created in "Creating and connecting to the databases" on page 43:

a. Expand the Object tree in the DB2 Control Center until you find the database. Right-click and select *Configure...*.

b. Select the *Performance* tab and the *Database Heap Size (dbheap)* parameter. Set this parameter to 900 in the value box displayed.

c. Click **OK**.

Refer to the *MQSeries Integrator Administration Guide* and your database documentation for more information or assistance for this task.

Configuring your broker domain

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

Work through the following steps:

Step 1. Creating the Configuration Manager.

Step 2. Creating the broker.

These steps are described using the Command Assistant. When you enter values in the entry fields, you will see the command that is generated displayed 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
```

Step 1: creating the Configuration Manager

Start the Command Assistant to create the Configuration Manager (select *Start->Programs->IBM MQSeries Integrator 2.0->Command Assistant->Create Configuration Manager*). Complete the fields on the first two screens. Review and check the full command and click **Finish** on the third summary screen.

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 data source name for the message repository (flag `-m`).

Notes:

1. 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, specify them here (flags `-u` and `-p` for the configuration repository and flags `-e` and `-r` for the message repository).
2. The queue manager will be created for you if it does not already exist.

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.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
BIP8071I: Successful command completion.
```

If the command detects an error, or cannot complete, it returns an error message on the command line, or in the Windows NT Event viewer (Application View) which includes an explanation of the error and any action to take. The error might have been caused by another program with which MQSeries Integrator interacts to complete this command, so you should also check for errors from Windows NT, DB2, and MQSeries.

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 *MQSeries IntegratorConfigMgr*. It has a default start up status of manual: you can change this to automatic if required.
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 84.
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 9 on page 81.
6. The database tables required by the message repository have been set up in the database MQSIMRDB. These tables are listed in Table 10 on page 82.
7. The Windows NT registry has been updated to record the Configuration Manager creation. Registry updates are described in “Windows NT registry” on page 79.

Step 2: creating the broker

Start the Command Assistant to create the broker (select *Start->Programs->IBM MQSeries Integrator 2.0->Command Assistant->Create Broker*). Complete the fields on the first two screens. Review and check the full command and click **Finish** on the third summary screen.

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).

Notes:

1. The service user ID and password are also used as the user ID and password for broker database. If you want to use a different user ID (flag `-u`) and password (flag `-p`) for access to this database, specify them here and grant the user ID access to the database (described in “Setting up database authorizations” on page 45).
2. The queue manager has already been created by the **mqsicreateconfigmgr** command.

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.

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.
```

If the command detects an error, or cannot complete, it returns an error message on the command line, or in the Windows NT Event viewer (Application View) which includes an explanation of the error and any action to take. The error might have been caused by another program with which MQSeries Integrator interacts to complete this command, so you should also check for errors from Windows NT, DB2, and MQSeries.

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 *MQSeries Integrator 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 84.
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 8 on page 80.
5. The Windows NT registry has been updated to record the broker creation. Registry updates are described in “Windows NT registry” on page 79.

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. Instead, 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 this task.

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

1. Starting the MQSeries listener.
2. Starting the Configuration Manager.
3. Starting your broker.
4. Starting 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.

Starting the MQSeries listener

Most of the resources you need to support this basic 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.

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:

```
runmqlsr -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 being used by another queue manager), you should assign an alternative port value. The port value is set in the Listener properties dialog (Parameters tab), or as the `-p` parameter on the `runmqlsr` command. If the port is already in use, the Control Center cannot 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 which listeners are already active using MQSeries Services.

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):

```
mqsistart 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.

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.

```
mqsistart 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 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-> Control Center*). Complete the following tasks to set up the environment you need to complete the basic verification described in “Verifying your installation” on page 54.

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*.

- Complete the *Configuration Manager Connection* dialog displayed by the Control Center. This provides the information required to connect your Control Center session to the Configuration Manager. The fields are:
  - Hostname. This is blank initially. Enter the network hostname of the system on which the Configuration Manager has been created. In the simple configuration defined in this chapter, the hostname is MQSISYS1. If you are using a different hostname, enter it here.
  - Port. This is blank initially. Enter the port number on which the queue manager is listening (you defined this in “Starting the MQSeries listener” on page 51). The default port number is 1414.
  - Queue Manager name. This is blank initially. 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.
- 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 your user ID is not a member of the MQSeries Integrator groups.

  Check the groups of which your current user ID is a member. Also check that you are logged onto the same security domain as the one you were logged onto when you installed MQSeries Integrator.

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

If you want to check, or change, these settings at a later time, click *File->Connection* to display 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 57
  - "Running the Soccer application" on page 58
  - "Running the Scribble application" on page 60
  - "Running the Postcard application" on page 62
- "Building and using a basic message flow" on page 64

All the tasks illustrated here assume you have used the sample names and values when you completed the tasks in "Configuring a basic broker domain" on page 37. 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

Some preparation is required before you can run any of the verification programs.

Note: If you are not using MQSI_SAMPLE_QM as the queue manager for your broker, you need to change the broker queue manager name in the output nodes of the sample verification applications.

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 Soccer application
  - MQSI_SOCCER_PUBLICATION_QUEUE
  - 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

Now you need to work with the MQSeries Integrator resources that are used by the applications.

Step 1. Stop the Control Center and the Configuration Manager (`mqsistop 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:

```
mqsiimpexpmsgset -i -n MQSIMRDB -u mqsiuid -p mqsipw -f 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, use them here.

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

Step 3. Restart the Configuration Manager (`mqsistart 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 displays *Untitled* to show that the workspace is empty. Now import the supplied workspace import file that defines the resources used by the verification programs:

  a. Select *File->Import to workspace*. The Import dialog is displayed.

  b. 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.

c. Click **Browse** and locate the `\examples` subdirectory (in the MQSeries Integrator home directory). Change the setting in the *Files of type* box to **All files (*.*)**. Select the sample workspace import file `SamplesWorkspaceForImport`, select **Open**, and then click **Import**.

   The definitions can take a few minutes to import. When import has finished, an Imported resources box is displayed, listing the eight resources that have been imported successfully. Click **OK** to close the dialog. You will now see the sample broker (MQSI_SAMPLE_BROKER) in the topology.

d. Select the *Message Flows* view to display the following message flows:
   - default publish/subscribe
   - Version 1 Migration/compatibility
   - Soccer
   - ScribbleInversion
   - Postcard

   The default message flows are described in *MQSeries Integrator Introduction and Planning*. The *Message Flows* view also displays the *IBMPrimitives* folder.

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.

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

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 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 Domain Topology panels of the Assignments view.

Step 9. Select *File->Check in->All (Save to Shared)*. The contents of the configuration repository are updated with the new definitions and assignments and all the files are checked in to the repository.

Step 10. Now 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 will contain the following messages: BIP4045I, BIP4040I and BIP2056I.

Step 12. View the deployed configuration graphically in the *Operations view* Refresh the view. The deployed broker, execution group, message set, and message flows are 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 basic broker domain” on page 37 (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 Soccer application

The Soccer application is written in the C programming language and demonstrates a number of basic publish/subscribe features. The application is a basic 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 and executable files) 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**.

You 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 Soccer 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 Soccer works: The Soccer 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 and forwards it to the publication node. The publication can indicate that:
- 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[2] 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 Soccer 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.

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 Soccer 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 and executable files) 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*.

[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*.

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 its 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 create and specify a different queue for each one. The definitions you completed in “Creating the MQSeries resources” on page 54 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 ESQL 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 2.0.2. It is written in C with a Java end-user interface.

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 and executable files) 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*.

A dialog prompts you for a nickname to use for sending or receiving messages. Enter an alphanumeric string of up to 24 characters. The dialog also prompts 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. 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 pull down 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 54. 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 55), you can create a message set based on a C header file (like `postcardstruc.h`) by importing 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 basic message flow

This verification scenario illustrates how you define a basic 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 46. 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 basic message flow"
- "Assigning the message flow to the broker" on page 66
- "Deploying the message flow to the broker" on page 66
- "Testing the message flow" on page 67

Creating the MQSeries resources

This basic verification test requires 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 accept the default values for all the other properties of the queue.

Step 5. Repeat the last two steps to define the output queue MQSI_OUTQ. Again, you can accept the default value for all other properties.

Creating a basic message flow

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

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, all the updated resources must be checked in. 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 as follows:

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.

| In addition to using MQSeries Explorer, you can test message flows using a
| GUI based program that is supplied in MQSeries SupportPac IH03.

A basic 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 37. 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 70
- "Creating new resources" on page 70
- "Restarting your broker domain" on page 73

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 37. All the assumptions stated in "Designing a basic configuration" on page 40 are still valid, and the following additional assumptions apply:

- The User Name Server has its own queue manager, MQSI_SAMPLE_UNS_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 queue manager being used for the broker and the Configuration Manager (MQSI_SAMPLE_QM) is 1414. The port number for the queue manager being used for the User Name Server 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 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->Command Assistant ->Modify Configuration Manager* and *Modify Broker*). Complete the fields in the screen displayed.

If you prefer, you can use the following commands:

```
mqsichangeconfigmgr -s MQSI_SAMPLE_UNS_QM

mqsichangebroker MQSI_SAMPLE_BROKER -s MQSI_SAMPLE_UNS_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->Command Assistant->Create User Name Server*). Complete the fields in the screen displayed.

If you prefer, you can use the following command:

```
mqsicreateusernameserver -i mqsiuid -a mqsipw -q MQSI_SAMPLE_UNS_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_UNS_QM.
Default objects statistics : 29 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
MQSeries queue manager 'MQSI_SAMPLE_UNS_QM' started.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
The setmqaut command completed successfully.
BIP8071I: Successful command completion.
```

If the command detects an error, or cannot complete, it returns an error message on the command line, or in the Windows NT Event viewer (Application View) which includes an explanation of the error and any action to take. The error might have been caused by another program with which MQSeries Integrator interacts to complete this command, so you should also check for errors from Windows NT and MQSeries.

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_UNS_QM. These resources are detailed in “Default MQSeries resources” on page 84.
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 79.

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.

Creating MQSeries resources

You need to complete the following tasks:

Step 1. Create the channels and the transmission queue on the User Name Server's queue manager.

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_UNS_QM
```

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_UNS_QM') 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_UNS_QM') 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:

1. Starting the listeners and channels
2. Starting the User Name Server
3. Restarting your broker
4. Restarting the Configuration Manager
5. Restarting 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 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_UNS_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.

```
runmqlsr -t tcp -p 1415 -m MQSI_SAMPLE_UNS_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 `runmqlsr` 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 73.

The broker's queue manager listener is already active (you started it earlier in "Starting the MQSeries listener" on page 51). 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_UNS_QM -c MQSI_SAMPLE_UN_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 retrieves the principals information immediately. If you start the broker first, it will try periodically to contact the User Name Server to request the information. The broker cannot check ACLs until it has contacted the User Name Server and records this information in the Windows NT Event log. After the User Name Server has returned principals information to the broker, the broker stores the information in case the User Name Server is unavailable in the future.

Note that the Configuration Manager does not maintain a local record of principals information. If the Configuration Manager cannot 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 is made, Control Center users cannot work with topic ACLs (although they can complete other tasks for which they have authorization).

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->Control Center* to restart the Control Center. You can check that the broker has re-initialized 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 appendix describes the changes made by installation and configuration to the systems you have set up in your broker domain. The following sections assume that you have followed the configuration guidance given in "Configuring a basic broker domain" on page 37.

- "Directory structure"
- "Environment variables" on page 78
- "Windows NT registry" on page 79
- "Database contents" on page 80
- "Default MQSeries resources" on page 84
- "Principals" on page 85

The default product home directory is:

```
C:\Program Files\IBM\MQSeries Integrator 2.0\
```

This is written as <mqsi_root> wherever it appears in this chapter.

The default DB2 home directory is:

```
C:\SQLLIB\
```

The default NEONRules and NEONFormatter Support home directory is:

```
C:\neonsoft\
```

Directory structure

The table below lists the subdirectories created and populated within your home directory <mqsi_root> when you carry out a Full installation. It also provides a brief description of the contents. If you have not done a Full installation, 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.

Directory structure

Table 7. Directory structure after installation

Directory Name	Contents
bin	Executable files, .exe, .dll, .lil
BOOK	PDF files for MQSeries Integrator Version 2.0.2 and NEONRules and NEONFormatter Support
classes	Java class files
CmdAsst	Command Assistant files
examples	C, C++ and Java language samples for MQSeries Integrator Version 2.0.2, MQSeries Integrator Version 2.0.1 and MQSeries Integrator Version 1.1, MQSC samples
include	Header and other files for samples
lib	Shared library files
log	Error log files
licenses	License agreement and information in all supported languages
messages	Description files for messages and exceptions
mrm	Message repository files
neonsoft	NEONRules and NEONFormatter Support
primitives	Message processing node files
Register	Product registration details
Tool	Graphical utility programs
uninst	Uninstall files

If you have installed the Tour plug-in products, 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;C:\neonsoft\gui;C:\neonsoft\bin;
  ```

- LIB

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

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

- INCLUDE

  If you install the Samples and SDK 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.

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.

If NEONRules and NEONFormatter Support is installed, you need to ensure that the file `neonreg.dat`, which is identified in the NN_CONFIG_FILE_PATH environment variable, contains the correct information to locate the NEONFormatter and NEONRules database. A sample `neonreg.dat` file is supplied in the directory:

```
<mqsi_root>\examples
```

Windows NT registry

The Windows NT registry is updated by the installation. The entries depend on the choices made during installation. Entries are placed within a number of keys below the top-level key HKEY_LOCAL_MACHINE:

- SOFTWARE\IBM\MQSeriesIntegrator

  Entries 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 Version 2.0.2

  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.
- SOFTWARE\New Era of Networks

An entry is added here for NEONRules and NEONFormatter Support, which is installed as part of MQSeries Integrator.

- SYSTEM\CurrentControlSet\Services\Eventlog\Application

  Entries are added when Windows NT services are created (for example, for each broker).

If DB2 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

When you create MQSeries Integrator resources after installation, a number of database tables are created for your broker, within the database you specify in the create command. This database can be created for each set of tables, or can be shared. The database tables schema inherits its name from the Data Source User ID specified in the create command.

You can use a single database for all your brokers, providing you use the same Data Source User ID when you create the brokers.

Table 8 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 8. 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
BMQEPUBDEST	Contains list of MQSeries Everyplace queues on which messages can be placed.
BMQEPUBMSGIN	Holds persistent MQSeries Everyplace publish/subscribe messages waiting to be processed by the broker.
BMQEPUBMSGOUT	Holds persistent MQSeries Everyplace publish/subscribe messages to be placed on an MQSeries Everyplace queue.

Table 8. Database tables for brokers (continued)

Table name	Description
BMQESTDMSGIN	Holds persistent MQSeries Everyplace non-publish/subscribe messages waiting to be processed by the broker.
BMQESTDMSGOUT	Holds persistent messages to be sent to MQSeries Everyplace destinations.
BSCADADEST	Contains list of SCADA clients to which messages can be sent.
BSCADAMSGIN	Holds persistent SCADA messages waiting to be processed by the broker.
BSCADAMSGOUT	Holds persistent messages to be sent to SCADA destinations.
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
BROKERAA	Broker process details to support recovery
BROKERAAEG	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 9 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 9. Database tables for the configuration repository

Table name	Description
CBROKER	Broker details
CBROKERCEG	Execution group to broker mapping information
CCICSADAPTER	Not used currently

Table 9. Database tables for the configuration repository (continued)

Table name	Description
CCMDFCTYPE	Not used currently
CCOLLECTIVE	Collectives details
CCOLLECTIVECBROKER	Collective to broker relationships
CDATACTXTFCTYPE	Not used currently
CDECFCTYPE	Not used currently
CDELETED	Deleted resources
CEG	Execution group details
CEGCMSGFLOW	Message flow mapping information
CEGCMSGPROJECT	Message set mapping information
CITERFCTYPE	Not used currently
CLOG	Event messages
CMSGFLOW	Message flow details
CMSGPROJECT	Message set details
CNAVTYPE	Not used currently
CNEIGHBOURS	Broker neighbor topology details
COUTSTANDING	Actions with outstanding responses
CPRIMITIVES	Primitive message processing node details
CPROPERTIES	UUID properties
CSRCADAPTER	Not used currently
CSUBSCRIBE	Subscriber information
CTGTADAPTER	Not used currently
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

Table 10 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 10. Database tables for the message repository

Table name	Description
CATEGORY_MEMBER	Category members

Table 10. Database tables for the message repository (continued)

Table name	Description
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 do not need to understand the contents of these tables.

Default MQSeries resources

When you create MQSeries Integrator components, some MQSeries resources are created for their use. Table 11 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 11. 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.
SYSTEM.BROKER.EXECUTIONGROUP.QUEUE	queue	broker	Target for messages to the broker.

Table 11. MQSeries Integrator default objects (continued)

Resource name	Type	Queue manager	Description
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 19).

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 18), 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 Samples and SDK components.

Appendix B. Uninstalling MQSeries Integrator

This appendix describes how to uninstall one or all of the MQSeries Integrator components on Windows NT.

Notes:

1. DB2 is **not** uninstalled by the MQSeries Integrator uninstall program. To uninstall DB2, run the DB2 Uninstall program, either from the Control Panel (select *Add/Remove Programs*) or from the Start menu (select *Start->IBM DB2 v7.1->Uninstall*).
2. You are not authorized to use this copy of DB2 for any purpose other than supporting MQSeries Integrator.
3. NEONRules and NEONFormatter Support is **not** uninstalled by the MQSeries Integrator uninstall program. To uninstall it, use the **New Era of Networks** install/uninstall utility, available from the Control Panel (select *Add/Remove Programs*).

If you intend to uninstall MQSeries for Windows NT, V5.1 and MQSeries Integrator, you **must** uninstall MQSeries Integrator first, since it depends on MQSeries functionality to complete successfully.

To uninstall MQSeries Integrator, you must log on using a user ID that is a member of the Windows NT **Administrators** group in the local account security domain.

Before you start

There are two uninstall options:

- *Uninstall including data*

  This option enables you to remove the product data and the product files. The following items are removed:

  – All MQSeries Integrator product directories. This includes the `\Tool\export` subdirectory where you might have stored information about your Control Center workspace. (`\Tool\Export` and `\Tool\Working` are the default directories when you select *File->Save Workspace As...* or *File->Export All in Workspace*, unless you have changed it in the Control Center Preferences).

  – 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.

  Any files you have saved in the MQSeries Integrator home directory are preserved. However, you are recommended to back up these files before you start the uninstallation.

  If you intend to recreate your configuration and workspace, you are recommended to take the steps in “Recommendations for preserving your data”.

- *Uninstall excluding data*

  If you choose this option, you will see a warning that all components and all product files in the home directory will be destroyed if you continue. Click **Next** to continue with the uninstallation.

  If you intend to recreate this data on another system, or on this system after re-installation, you are recommended to take the steps in “Recommendations for preserving your data”.

  This option does **not** remove the following items:

  - 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.
  - The contents of all tables in the configuration repository, and the contents of all tables in the message repository. If you want to remove them, run the **mqsideleteconfigmgr** 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 should 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. To check which tables have been created by MQSeries Integrator, refer to “Database contents” on page 80. Note that each broker has a set of tables listed in Table 8 on page 80.

Recommendations for preserving your data

Before you start an uninstallation, you are recommended to take the action below to preserve any data you might need if you intend to recreate your configuration and workspace.

- If you want to keep a workspace, you can do one of the following:
  - Select *File->Check In->All (Save to Shared)* to record updates in the broker and configuration repository tables. Then run *File->Save Workspace As* to save a local file. Since the option *Uninstall including data* destroys the

contents of the broker database tables, you must also create a backup of these database tables so that you can recreate them in a new installation if you choose.

Note: If you select *File->Save As...* to save the workspace, a local copy is saved in the hard disk of your 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.

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

  File->Export All in Workspace 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 produces a file that is almost 10 times bigger than a file produced by 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 **mqsiimpexpmsgset**. See the *MQSeries Integrator Administration Guide* for details of this command.

Before you uninstall, you are recommended to stop the following manually:
- The Control Center
- The Configuration Manager
- The User Name Server
- Any brokers that are running

Using the uninstall program

You can uninstall one or more of the product components as follows:

1. Run the uninstall program using one of the following methods:
   - Select MQSeries Integrator from Add/Remove Programs in the Control Panel. This is the recommended method.
   - Select Run from the Start menu or run the uninstall program directly from a command prompt. The program is called `mqsiunin.exe`, and it is located in the `\uninst` subdirectory of the MQSeries Integrator home directory. The default location is

     `C:\Program Files\IBM\MQSeries Integrator 2.0.`
   - Select *Start->Programs->IBM MQSeries Integrator 2.0->Uninstall.*
2. Select one of the following options:
   - Uninstall one or more components.
   - Uninstall MQSeries Integrator completely, including data.
   - Uninstall MQSeries Integrator, excluding data.

- Uninstall CSDs.

Notes:

a. The option *Uninstall CSDs* is not available if one of the following is true:
   - You have not installed any CSDs for 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.

b. If you have installed more than one CSD, the option *Uninstall CSDs* uninstalls the latest CSD and restores the previous CSD level. If you have installed only one CSD, this option will restore the base level of code.

c. The option *Uninstall MQSeries Integrator excluding data* leaves components you have added 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 later reinstall the product, you can access and reuse this data.

d. The option *Uninstall one or more components* displays a list of components to uninstall. This option excludes data: any files you have created are preserved; product files only are removed. If you select every component, the `\uninst` subdirectory, and its contents are also removed.

The uninstall program appends records to the log file `MQSI2.log`, created by the MQSeries Integrator installation in the Windows NT root directory.

Migration considerations

If you have a previous installation of MQSeries Integrator Version 1 that you did not uninstall before installing MQSeries Integrator Version 2, you can now re-establish 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 MQSeries Integrator 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.

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 **mqsideletebroker** 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 94 for details of this process).
4. If the Configuration Manager exists on this system, invoke the **mqsideleteconfigmgr** 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 94 for details of this process).
5. If the User Name Server exists on this system, invoke the **mqsideleteusernameserver** 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 94 for details of this process).
6. If you have successfully deleted all the components on this system, you can remove the MQSeries Integrator Version 2.0.2 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 2.0.2 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
   ```
7. If you have created any folders to store 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, you should now save these files.

8. 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
   ```

   If you cannot delete the product files because they are in use, check that the Control Center is not active. If it is, close it. If you still cannot delete the files, and you cannot identify a process that is holding the file lock, restart the system.
9. 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\IBMMQSeries Integrator 2.0
   ```

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

Unattended Removal of MQSeries Integrator Version 2.0.2 on Windows NT

MQSeries Integrator Version 2.0.2 can be uninstalled (removed) without user interaction. This process is called unattended (or silent) removal and it uses a response file. This is a text file containing values for the options you select when you uninstall MQSeries Integrator Version 2.0.2.

To create a response file you can do either of the following:
- Edit the response file (in `<mqsi_root>\uninst` or `mqsiunin.rspmqsiunin.rsp`) supplied on the IBM MQSeries Integrator Version 2 CD 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 `mqsiunin.exe` from the root directory, with the fully-qualified path to `mqsiunin` in double quotes, as follows:

```
c:\>"c:\Program Files\IBM\MQSeries Integrator 2.0\uninst\mqsiunin" -s -f1<responsefile>
```

where c:\> is the command prompt.

In this 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 `mqsiunin.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.

Uninstalling MQSeries Integrator

This is a sample of the response file that you need to create to run the Silent Uninstall.

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

Notes:

1. The uninstall ends if any locked files are found.
2. The Broker component is not 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
```

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 **mqsiimpexpmsgset** command with the -e flag set.
  - 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 7.1).
  - 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.

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 116.

You can also download CSDs from this web site.

Applying the maintenance information

To apply maintenance updates to MQSeries Integrator Version 2.0.2:

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 the Control Center.
   c. Ensure that you do not have any MQSeries Integrator Version 2.0.2 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 should:
   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 downloaded, change to the download directory and run the executable file.

      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.

The `setup.exe` file now runs.

5. To install from CD:
   a. Read the file `memo.ptf` and any `Readme.txt` files, both 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 autorun is enabled, `setup.exe` runs automatically. If not, double-click on `setup.exe` on the CD.
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 might 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 97).

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 2.0*, and click **Add/Remove**. This starts the MQSeries Integrator uninstall program, and presents the uninstall dialog (see “Appendix B. Uninstalling MQSeries Integrator” on page 87 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->Service Level*. This displays the `memo.ptf` file 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.

Checking the service level

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 116.

Appendix D. 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 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:

400
AIX
AS/400
DB2
DB2 Universal Database
First Failure Support Technology
IBM
IBMLink
Lotus
MQSeries
OS/390
SupportPac
Tivoli
VisualAge
VSE/ESA

Java, JDBC and JDK are registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Lotus is a trademark of Lotus Development Corporation in the United States, other countries, or both.

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

Tivoli is a trademark of Tivoli Systems Inc. in the United States, other countries, or both.

UNIX is a registered trademark in the United States and other countries of X/Open Company Limited.

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 respository.

check out. The Control Center action that extracts and locks a resource from the configuration or message respository 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*.

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*.

debugger. A facility on the *Message Flows* view in the Control Center that enables message flows to be debugged.

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 (DTD). 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.

external reference. A reference within a message set to a component that has been defined outside the current message set. For example, an integer that defines the length of a string element might be defined in one message set but used in several message sets.

F

field reference. A sequence of period-separated values that identify a specific field (which might be a structure) within a message tree. An example of a field reference might be something like `Body.Invoice.InvoiceNo`.

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 (JRE). 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 value that determines how the message is interpreted (parsed). The following domains are recognized:

- MRM, which identifies messages defined using the Control Center
- NEONMSG[3], which identifies messages created using the NEONFormatter user interfaces.
- XML, which identifies messages that are self-defining
- BLOB, which identifies messages that are undefined

[3] The message domain NEON is also recognized for compatibility with previous releases.

You can also create your own message domains: if you do so, you must supply your own message parser.

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 allows application programs to access message queuing services. See also *AMI* and *JMS*.

message repository. A database holding message template definitions.

message repository manager (MRM). A component of the Configuration Manager that handles message definition and control. A message defined to the MRM has a message domain set to MRM.

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.

MQe. MQSeries Everyplace.

MQI. Message queue interface.

MQIsdp. MQSeries Integrator SCADA device protocol. A lightweight publish/subscribe protocol flowing over TCP/IP.

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.

MQSeries Everyplace. A generally available MQSeries product that provides proven MQSeries reliability and security in a mobile environment.

MRM. Message Repository Manager.

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

SCADA. Supervisory, Control, And Data Acquisition.

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.

stream. A method of topic partitioning used by MQSeries Publish/Subscribe applications.

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.

Supervisory, Control, And Data Acquisition. A broad term, used to describe any form of remote telemetry system used for gathering data from remote sensor devices (for example, flow rate meters on an oil pipeline) and for the near real time control of remote equipment (for example, pipeline valves).

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*.

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.

Bibliography

This section describes the documentation available for all current MQSeries Integrator products.

MQSeries Integrator Version 2.0.2 cross-platform publications

The MQSeries Integrator cross-platform publications are:

- *MQSeries Integrator Introduction and Planning*, GC34-5599
- *MQSeries Integrator Using the Control Center*, GC34-5602
- *MQSeries Integrator Messages*, GC34-5601
- *MQSeries Integrator Programming Guide*, SC34-5603
- *MQSeries Integrator Administration Guide*, SC34-5792
- *MQSeries Integrator ESQL Reference*, SC34-5923

These books are all available in hardcopy.

You can order publications from the IBMLink™ Web site at:

`http://www.ibm.com/ibmlink`

In the United States, you can also order publications by dialing 1-800-879-2755.

In Canada, you can order publications by dialing 1-800-IBM-4YOU (1-800-426-4968).

For further information about ordering publications contact your IBM authorized dealer or marketing representative.

MQSeries Integrator Version 2.0.2 platform-specific publications

Each MQSeries Integrator product provides one platform-specific installation guide, which is supplied in hardcopy.

MQSeries Integrator for AIX Version 2.0.2
MQSeries Integrator for AIX Installation Guide, GC34-5841

MQSeries Integrator for HP-UX Version 2.0.2
MQSeries Integrator for HP-UX Installation Guide, GC34-5907

MQSeries Integrator for Sun Solaris Version 2.0.2
MQSeries Integrator for Sun Solaris Installation Guide, GC34-5842

MQSeries Integrator for Windows NT Version 2.0.2
MQSeries Integrator for Windows NT Installation Guide, GC34-5600

MQSeries Everyplace publications

If you intend to connect MQSeries Everyplace applications to message flows that include the MQSeries Everyplace message flow nodes, you will find the following publications useful:

- *MQSeries Everyplace for Multiplatforms Version 1.1 Introduction*, GC34-5843
- *MQSeries Everyplace for Multiplatforms Version 1.1 Programming Guide*, SC34-5845
- *MQSeries Everyplace for Multiplatforms Version 1.1 Programming Reference*, SC34-5846

- *MQSeries Everyplace for Multiplatforms Version 1.1 Native Client Information*, SC34-5880

You can find these books on the MQSeries Web site (see "MQSeries information available on the Internet" on page 116). Translated versions of these books are also available in some languages from the same Web site.

NEONRules and NEONFormatter Support for MQSeries Integrator publications

The following publications are supplied on the product CD in PDF format, and are installed with the Documentation component.

- *NEONRules and NEONFormatter Support for MQSeries Integrator User's Guide*
- *NEONRules and NEONFormatter Support for MQSeries Integrator System Management Guide*
- *NEONRules and NEONFormatter Support for MQSeries Integrator Programming Reference for NEONRules*
- *NEONRules and NEONFormatter Support for MQSeries Integrator Programming Reference for NEONFormatter*
- *NEONRules and NEONFormatter Support for MQSeries Integrator Application Development Guide*

These books are provided in US English only.

Softcopy books

All the MQSeries Integrator books are available in softcopy formats.

Portable Document Format (PDF)

All books in the MQSeries Integrator library are supplied in US English only in a searchable PDF library on the product CD.

You can install the library as follows:

- On AIX, invoke `install –d` and select the documentation fileset. After installation, run the command `mqsidocs`. This launches Acrobat Reader and opens the PDF package.
- On HP-UX, invoke `swinstall –d` and select `MQSI-DOCS` from the menu. After installation, run the command `mqsidocs`. This launches Acrobat Reader and opens the PDF package.
- On Sun Solaris, invoke `pkgadd –d` and select `mqsi-docs` from the menu. After installation, run the command `mqsidocs`. This launches Acrobat Reader and opens the PDF package.
- On Windows NT, select the `Online Documentation` component on a custom installation, or do a full installation. After installation, select *Start—>Programs—>IBM MQSeries Integrator 2.0—>Documentation*.

In addition, PDF files for books that have been translated are installed into the location `mqsi_root/bin/book/pdf/<locale>` (on UNIX) or `mqsi_root\bin\book\pdf\<locale>` (on Windows NT) where `<locale>` is one of the following:

- **de_DE** for German
- **en_US** for US English
- **es_ES** for Spanish
- **fr_FR** for French
- **it_IT** for Italian
- **ja_JP** for Japanese
- **ko_KR** for Korean
- **pt_BR** for Brazilian Portuguese
- **zh_CN** for Simplified Chinese
- **zh_TW** for Traditional Chinese

An index file (in HTML format) that provides a link to each book is supplied for each language. For example, the French index file is called `indexfr.htm`. The files are stored in the following directory:

- On UNIX, `<mqsi_root>/docs/`
- On Windows NT, `<mqsi_root>\bin\book`

Each index file has an entry for every book: if a particular book has not been translated into the appropriate language for that index file, a link to the English PDF is included. You can use any Web browser to view the index file. On Windows NT, you can also access the index file through the *Start* menu.

The PDF file names for the English books are shown in Table 12.

Table 12. File names of MQSeries Integrator book PDFs

Book title	File name
MQSeries Integrator for AIX Installation Guide	bipaac04.pdf
MQSeries Integrator for HP-UX Installation Guide	bipcac00.pdf
MQSeries Integrator for Sun Solaris Installation Guide	bip7ac03.pdf
MQSeries Integrator for Windows NT Installation Guide	bipyac03.pdf
MQSeries Integrator Introduction and Planning	bipyab02.pdf
MQSeries Integrator Administration Guide	bipyag04.pdf
MQSeries Integrator Using the Control Center	bipyar03.pdf
MQSeries Integrator ESQL Reference	bipyae00.pdf
MQSeries Integrator Programming Guide	bipyal02.pdf
MQSeries Integrator Messages	bipyao02.pdf

The fifth character of the file name indicates the language of the book (**a** indicates US English). You can deduce the file names of translated books by using the following substitutions for the fifth character:

- **g** for German
- **s** for Spanish
- **f** for French
- **i** for Italian
- **j** for Japanese
- **k** for Korean
- **b** for Brazilian Portuguese
- **z** for Simplified Chinese
- **t** for Traditional Chinese

PDF files can be viewed and printed using the Adobe Acrobat Reader.

If you cut and paste examples of commands from PDF files to a command line for execution, you must check that the content is correct before you press `Enter`. Some characters might be corrupted by local system and font settings.

If you need to obtain the Adobe Acrobat Reader, or would like up-to-date information about the platforms on which the Acrobat Reader is supported, visit the Adobe Systems Inc. Web site at:

```
http://www.adobe.com/
```

PDF versions of all current MQSeries Integrator books are also available from the MQSeries product family Web site at:

```
http://www.ibm.com/software/mqseries/
```

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.

For Windows NT installation tasks you might need:

- *MQSeries for Windows NT and Windows 2000 Quick Beginnings, GC34-5389*

For planning and configuration tasks you might need:

- *MQSeries MQSC Command Reference, SC33-1369*

  This book contains the syntax of the MQSC commands.
- *MQSeries System Administration, SC33-1873*

  This book supports day-to-day management of local and remote MQSeries objects.
- *MQSeries Clients, GC33-1632*

  This book describes how to install, configure, use, and manage MQSeries clients.
- *MQSeries Intercommunication, SC33-1872*

  This book describes MQSeries Intercommunication between different platforms.

For a complete list of MQSeries product publications, refer to the information on the MQSeries Web site. The URL is given in the section "MQSeries information available on the Internet").

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:

- *MQSeries Publish/Subscribe User's Guide, GC34-5269*

This book and the MQSeries Publish/Subscribe Software Development Kit (SDK) package are available on the MQSeries Web site. The URL is given in the section "MQSeries information available on the Internet").

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. The URL is given in the section "MQSeries information available on the Internet").

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*

You can download these publications from the DB2 Web site at

```
http://www.ibm.com/software/db2
```

MQSeries information available on the Internet

The MQSeries product family Web site is at:

```
http://www.ibm.com/software/mqseries/
```

By following links from this Web site you can:

- Obtain latest information about the MQSeries product family.

- Access the MQSeries books in HTML and PDF formats.
- | Obtain information about complementary offerings by following these links:
  - | IBM Business Partners
  - | Partner Offerings (within *Related links*)
- Download an MQSeries SupportPac.

Index

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 44–1962–842327
  - From within the U.K., use 01962–842327
- 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.