MS81: MQSeries® internet pass-thru
Version 1.1

Document:	MS81 SCRIPT
Issued:	March 2001
Revision Date:	March 2001
Previous Revision Date:	December 2000
Next Review:	As required

MQSeries Development
IBM United Kingdom Laboratories
Hursley Park
Winchester, SO21 2JN
England

Take Note!

Before using this manual and the product it supports, be sure to read the general information under "Notices" on page 6.

Second Edition, March 2001

This edition applies to Version 1.1 of MS81: MQSeries internet pass-thru (program number 5639-L92) and to all subsequent releases and modifications until otherwise indicated in new editions.

Changes made to this second edition are indicated by vertical bars to the left of the changes.

A form for reader's comments is provided at the back of this publication.

Contents

Figures

Notices

The following paragraph does not apply in any 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 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.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe any of the intellectual property rights may be used instead of the IBM product. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, USA.

The information contained in this document has not be submitted to any formal IBM test and is distributed AS IS. The use of the information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Trademarks

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

AIX
FFST
First Failure Support Technology
IBM
IBMLink
MQSeries
SupportPac

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

Preface

What is internet pass-thru?

IBM® MQSeries internet pass-thru:

- Is an MQSeries base product extension that can be used to implement messaging solutions between remote sites across the Internet.
- Makes the passage of MQSeries channel protocols into and out of a firewall simpler and more manageable by tunneling the protocols inside HTTP or acting as a proxy.
- Operates as a standalone service that can receive and forward MQSeries message flows. The system on which it runs does not have to host an MQSeries queue manager.
- Helps you to provide business-to-business transactions using MQSeries.
- Enables existing, unchanged MQSeries applications to be used through a firewall
- Provides a single point of control over access to multiple queue managers.

In this book, MQSeries internet pass-thru is often termed “MQIPT” for convenience.

Who this book is for

This book is for systems designers, technical MQSeries administrators, and firewall and network administrators.

What you need to know to understand this document

You need a good understanding of:

- The administration of MQSeries queue managers and message channels, as described in *MQSeries System Administration* and *MQSeries Intercommunication*
- The way that firewalls are implemented
- Internet protocol routing/networking
- The IBM Network Dispatcher for load balancing and enhanced availability.

Prerequisites

This release of internet pass-thru runs on these platforms:

- Windows NT® V4.0, with Service pack 6
- Windows 2000
- Sun Solaris V2.7
- AIX® V4.3.3
- HP-UX 11

The JDK should be at level 1.3.0, or later compatible release, to benefit from the improved performance of more recent JDK releases. Earlier releases of JDK, back to 1.1.8, also work with internet pass-thru.

The only supported network protocol is TCP/IP.

The Administration Client help requires Adobe Acrobat Reader.

Accessibility information

The Administration Client GUI has been built with accessibility in mind. It is straightforward to perform all of the available functions without using a mouse, by using keyboard equivalents. You can navigate round the screen by using tab, shift-tab, ctrl-tab, and the cursor keys in the standard manner. The equivalent to pressing buttons can be achieved by first selecting the button and then pressing the enter key.

Menu options can be reached either by combinations of tab and cursor keys or by using the accelerator keys, which are available for all the options. For example, the GUI can be closed by selecting first alt-f, then alt-q (File->Quit). Once a menu item has been reached, it can be activated by using the enter key.

You can navigate around the tree by using the cursor keys. In particular, the right and left cursor keys can be used to open or close an MQIPT node, allowing the routes to be either shown or hidden.

Selected checkboxes can have their states changed by using the space key. Fields can be selected for editing by using the enter key.

Look and feel

Ideally the GUI should adopt the look and feel of the environment. As this is not always possible, you may provide a configuration file to tailor the look and feel of the GUI to suit your needs. The configuration file is called "custom.properties" and should be placed in the `bin` subdirectory.

Use this configuration file to configure the following:

- The foreground color - the color of the text
- The background color
- The font of the text
- The style of the text - whether plain, bold, italic, or bold and italic

A sample configuration file "customSample.properties" has been provided, which contains comments showing how it can be changed. You are encouraged to copy this file to `bin/custom.properties` and to make the required changes.

Bibliography

You will also find the following publications useful:

- *MQSeries Intercommunication*, SC33-1872
- *MQSeries System Administration*, SC33-1873
- *MQSeries Clients*, GC33-1632

These books provide information about the definition of MQSeries channels and their attributes - in particular, the definition of CONNAME.

The MQSeries publications are available at:

`http://www.ibm.com/software/ts/mqseries/library/manualsa/index.htm`

Summary of changes

Changes to this edition of the manual are indicated by vertical bars to the left of the changes. The changes reflect enhancements in this version of MQSeries internet pass-thru. The enhancements include:

- The addition of AIX, HP-UX, and Windows 2000 as platforms for MQIPT.
- The addition of HTTP proxy support.
- The addition of Secure Socket Layer (SSL) support.
- The ability of MQIPT to communicate with another external MQIPT or MQSeries server through a SOCKS proxy.
- The use of an Administration Client GUI to make administration of one or more MQIPTs easier.
- The addition of support for the IBM Network Dispatcher.
- Minor improvements to tracing.
- Minor improvements to the `mqiptAdmin` command.

Chapter 1. Introduction to MQSeries internet pass-thru

MQSeries internet pass-thru is an extension to the base MQSeries product. MQIPT runs as a standalone service that can receive and forward MQSeries message flows, either between two MQSeries queue managers or between an MQSeries client and an MQSeries queue manager. MQIPT enables this connection when the client and server are not on the same physical network.

One or more MQIPTs can be placed in the communication path between two MQSeries queue managers, or between an MQSeries client and an MQSeries queue manager. The MQIPTs allow the two MQSeries systems to exchange messages without needing a direct TCP/IP connection between the two systems. This is useful if the firewall configuration prohibits a direct TCP/IP connection between the two systems.

MQIPT listens on one or more TCP/IP ports for incoming connections, which can carry either normal MQSeries messages, MQSeries messages tunneled inside HTTP, or encrypted using Secure Sockets Layer (SSL). It can handle multiple concurrent connections.

The MQSeries channel that makes the initial TCP/IP connection request is referred to as the “caller”, the channel to which it is attempting to connect as the “responder”, and the queue manager that it is ultimately trying to contact as the “destination queue manager”.

The anticipated uses of MQIPT are:

- MQIPT can be used as a channel concentrator, so that channels from or to multiple separate hosts can appear to a firewall as if they are all from or to the MQIPT host. This makes it easier to define and manage firewall filtering rules.

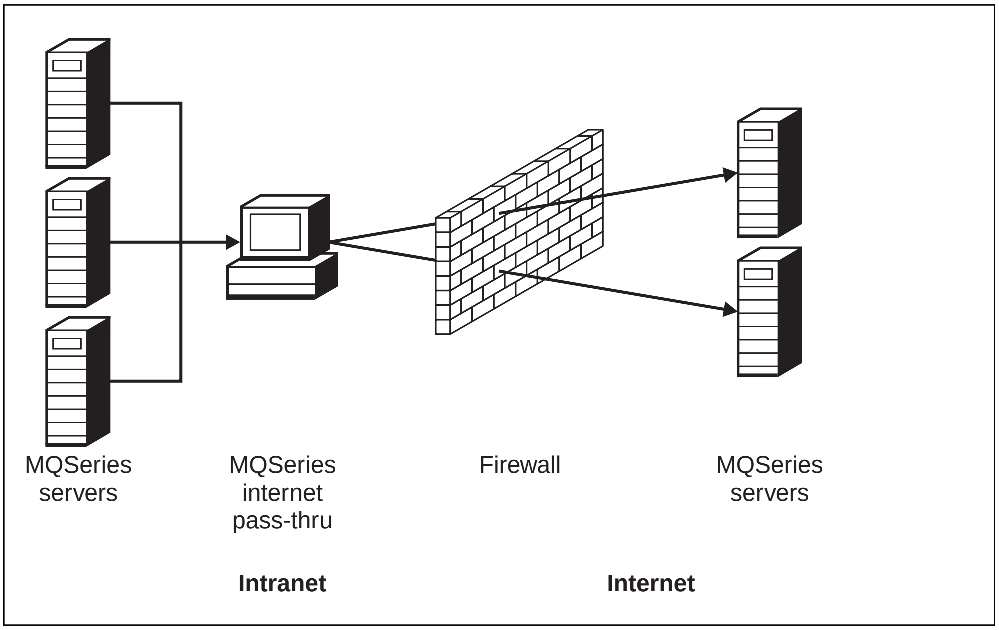


Figure 1. Example of MQIPT as a channel concentrator

- If MQIPT is placed in the firewall's "demilitarized zone" (DMZ), on a machine with a known and trusted internet protocol (IP) address, MQIPT can be used to listen for incoming MQSeries channel connections which it can then forward to the trusted intranet; the inner firewall must allow this trusted machine to make inbound connections. In this configuration, MQIPT prevents external requests for access from seeing the true IP addresses of the machines in the trusted intranet. Thus, MQIPT provides a single point of access.

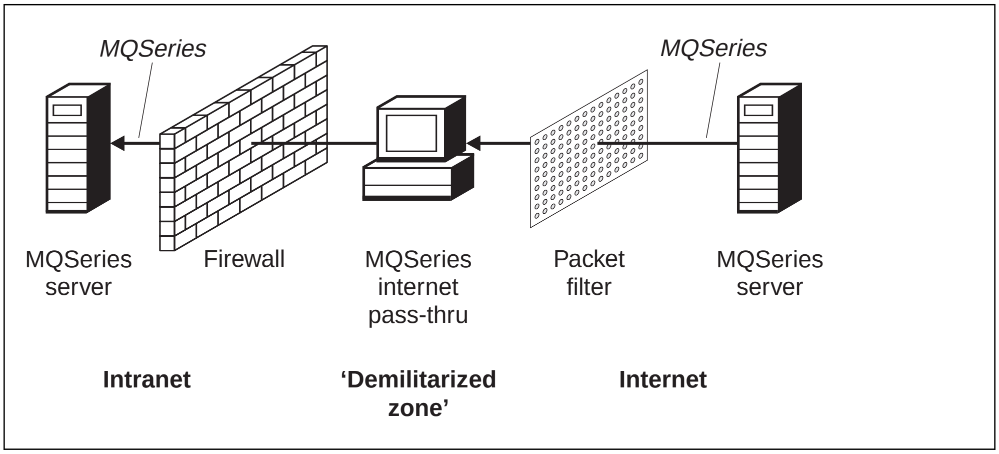


Figure 2. Example of MQIPT with a "demilitarized zone"

- If two MQIPTs are deployed inline, they can communicate using HTTP or SSL. The HTTP tunneling feature enables requests to be transmitted through firewalls, by the use of existing HTTP proxies. The first MQIPT inserts the MQSeries protocol into HTTP and the second extracts the MQSeries protocol from its HTTP wrapper and forwards it to the destination queue manager.

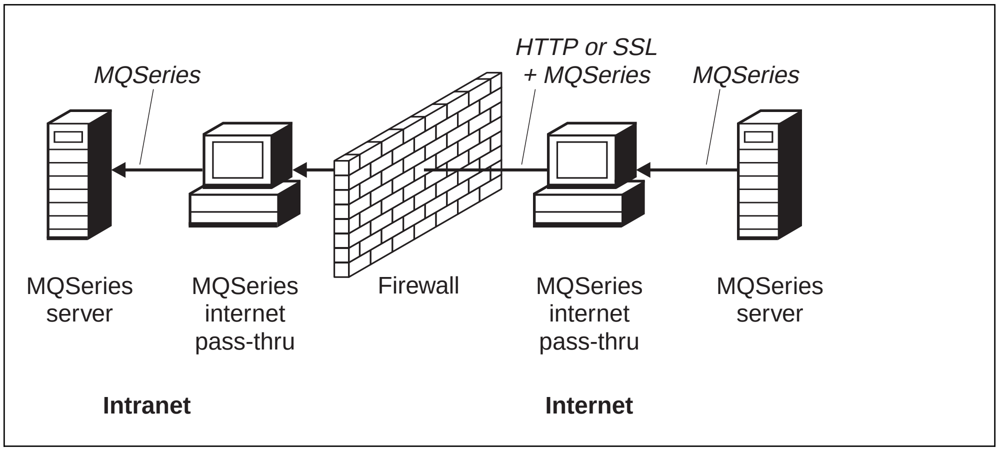


Figure 3. Example of MQIPT and HTTP tunneling

- Similarly, requests can be encrypted before transmission through firewalls. The first MQIPT encrypts the data and the second decrypts it using SSL before sending it to the destination queue manager.

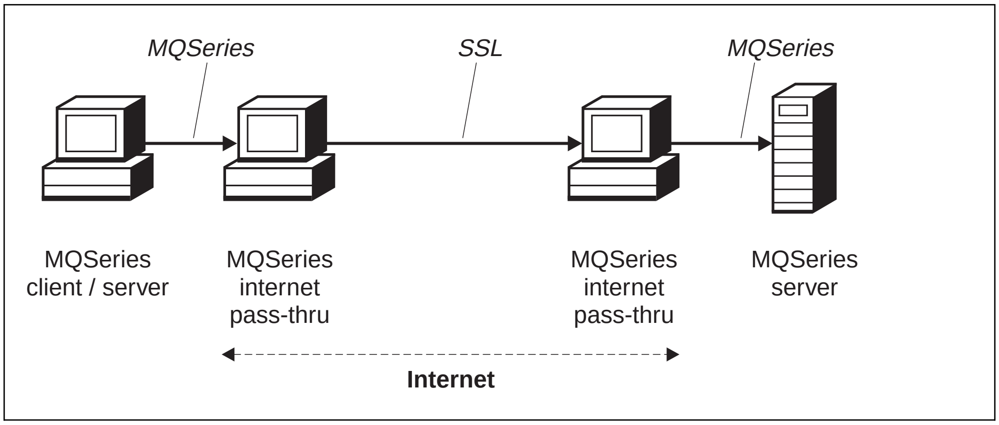


Figure 4. Example of MQIPT and SSL

MQIPT holds data in memory as it forwards it from its source to its destination. No data is saved on disk (except for memory paged to disk by the operating system). The only time MQIPT accesses the disk explicitly is to read its configuration file and to write log and trace records.

The full range of MQSeries channel types can be made through one or more MQIPTs. The presence of MQIPTs in a communication path has no effect on the functional characteristics of the connected MQSeries components, but there might be some impact on the performance of message transfer.

MQIPT can be used in conjunction with MQSeries Publish/Subscribe or the MQSeries Integrator message broker.

Figure 5 on page 16 shows all the possible configurations for MQIPTs in an MQSeries topology. In the figure, note that the HTTP proxy, SOCKS proxy, and MQIPT machines beyond the firewall on the “Outbound connections” side represent the possibility of multiple machines chained together on the internet. For example, an MQIPT machine could communicate through one or more SOCKS or HTTP proxy machines, or further MQIPT machines, before reaching its target.

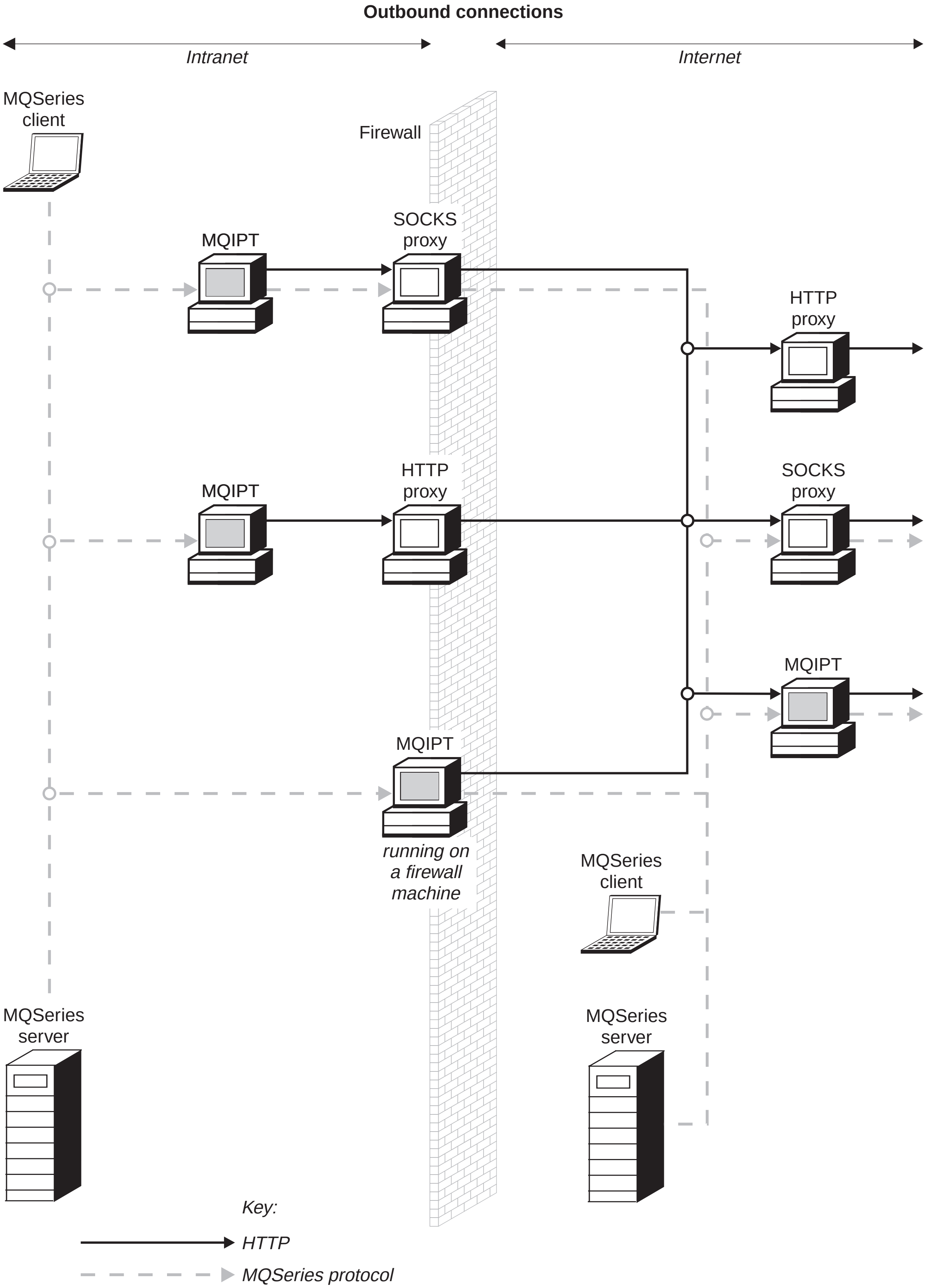


Figure 5 (Part 1 of 2). MQSeries topology showing possible MQIPT configurations

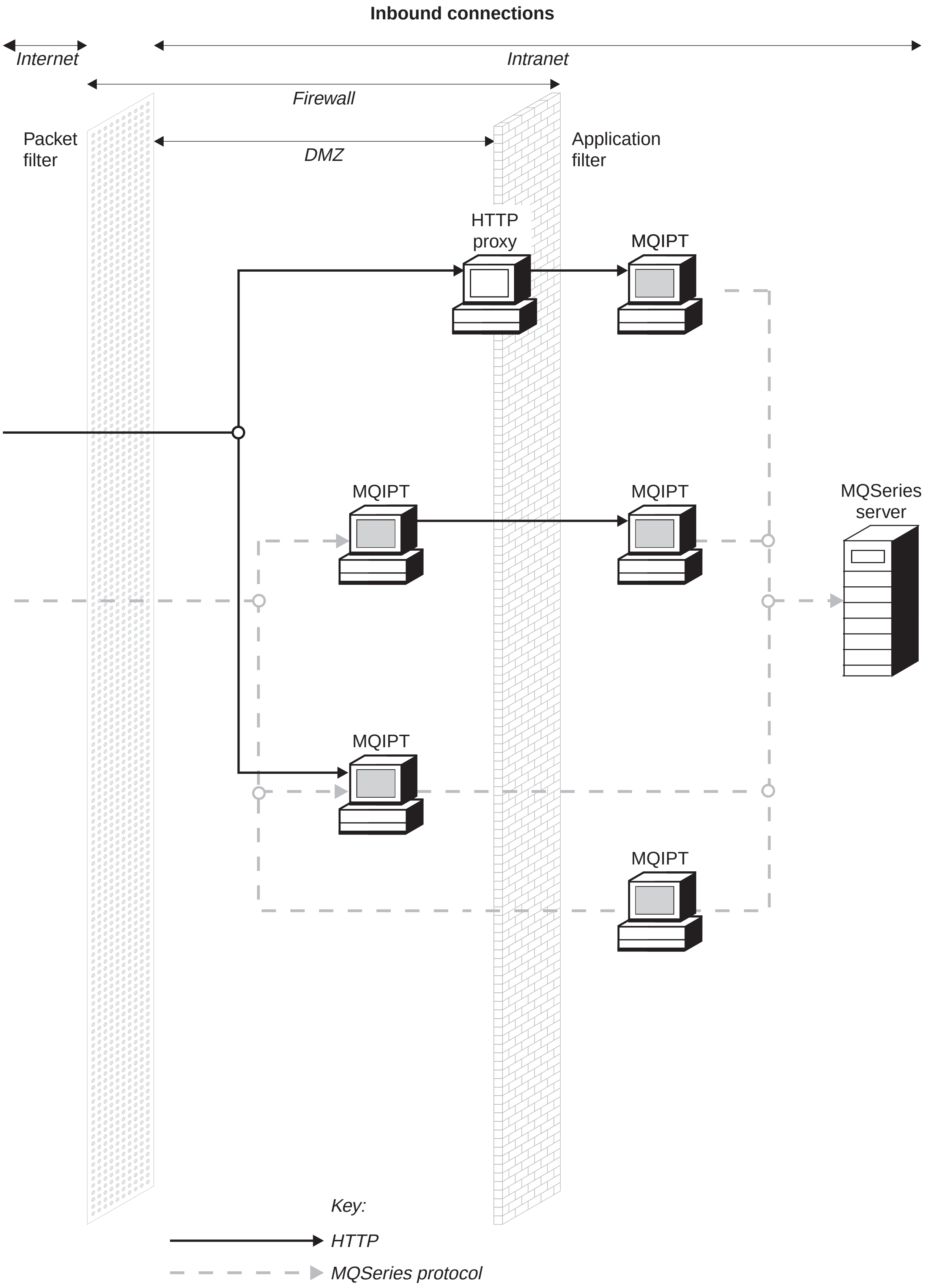


Figure 5 (Part 2 of 2). MQSeries topology showing possible MQIPT configurations

Chapter 2. How internet pass-thru works

This chapter gives an overview of the way internet pass-thru works, and then describes the following items in more detail:

- “HTTP support” on page 19
- “SOCKS support” on page 19
- “SSL support for security” on page 20
- “Network Dispatcher support” on page 21
- “Supported channel configurations” on page 23
- “Normal termination and failure conditions” on page 24
- “SSL support for security” on page 20
- “Other security considerations” on page 24

Overview of how internet pass-thru works

In its simplest configuration, MQIPT acts as an MQSeries protocol forwarder. It listens on a TCP/IP port (1414 by default) and accepts connection requests from MQSeries channels. If a well-formed request is received, MQIPT establishes a further TCP/IP connection between itself and the destination MQSeries queue manager. It then passes all protocol packets it receives from its incoming connection on to the destination queue manager, and it returns protocol packets from the destination queue manager back on the original incoming connection.

No change to the MQSeries protocol (client/server or queue manager to queue manager) is involved - because neither end is directly aware of the presence of the intermediary - so new versions of the MQSeries client or server code are not required.

To use MQIPT, the caller channel must be configured to use MQIPT’s hostname and port, not the hostname and port of the destination queue manager. MQIPT does not examine the channel name; this is simply passed through to the destination queue manager. Other configuration fields, such as the userid and password in a client/server channel, are similarly passed to the destination queue manager.

MQIPT can be used to allow access to one or more destination queue managers. For this to work, there must be a mechanism to tell MQIPT which queue manager to connect to, so MQIPT uses the incoming TCP/IP port number to determine which queue manager to connect to, as described in the next paragraph.

To allow access to more than one destination queue manager, MQIPT can be configured to listen on multiple TCP/IP ports. Each listening port is mapped to a destination queue manager through an MQIPT “route”. The MQIPT administrator may define up to 100 such routes, which associate a listening TCP/IP port with the hostname and port of the destination queue manager. This means that the hostname (IP address) of the destination queue manager is never visible to the originating channel. Each route can handle multiple connections between its listening port and destination, each condition acting independently.

HTTP support

As an option, MQIPT can be configured so that the data packets it forwards are encoded as HTTP requests. MQIPT supports HTTP tunneling with or without chunking.

Because today’s MQSeries channels do not accept HTTP requests, a second MQIPT is required to receive the HTTP requests and convert them back into normal MQSeries protocol packets. The second MQIPT strips off the HTTP header to convert the incoming packet back into a standard MQSeries protocol packet, before passing it on to the destination queue manager.

When using HTTP tunneling without chunking, an HTTP reply is sent back to the first MQIPT for each HTTP request. This reply can be the response from the destination queue manager or a dummy acknowledgement. If either MQSeries system has to send a chain of successive MQSeries protocol packets (as happens when transferring a large message), several HTTP request/reply pairs are used to transfer the data. To achieve this, MQIPT inserts additional request or reply flows.

When using HTTP tunneling with chunking, only the first packet is wrapped in an HTTP header. Middle and last packets have chunking headers. This arrangement removes the wait for a dummy acknowledgement from the second MQIPT, and thus offers slightly better performance than that provided by HTTP tunneling without chunking.

When HTTP is being used between two MQIPTs, the TCP/IP connection on which the HTTP requests and replies are flowed is kept open for the lifetime of the message channel. The MQIPTs do not close the TCP/IP connection between request/reply pairs.

If two MQIPTs are communicating through HTTP, it is possible that an HTTP request might stay outstanding for an extended period. An example is in a requester/server channel, when the server side is waiting for new messages to arrive on its transmission queue. The MQSeries channel protocol provides a “heartbeat” mechanism, which requires the waiting end periodically to send heartbeat messages to its partner (the default channel heartbeat period is 5 minutes) and MQIPT uses this heartbeat as the HTTP reply. Do not disable this channel heartbeat, or set it to an excessively high value, to avoid causing problems with timeouts in some firewalls.

SOCKS support

When making outbound connections through a firewall, an application can be SOCKS-enabled, so that all connections are made through a SOCKS proxy, thereby enabling a control point of exit through the firewall.

You configure MQIPT to use a SOCKS proxy by passing the host name or address and port number of the SOCKS proxy as parameters when MQIPT is started. For more information, see “Starting internet pass-thru from the command line” on page 29.

SSL support for security

You can configure MQIPT to use Secure Socket Layer (SSL) communications with another MQIPT. SSL V3.0 has been implemented, using Public Key Cryptography Standards (PKCS) #12 tokens stored in key ring files (with file types of .p12 or .pfx), containing X509.V3 certificates.

Various cryptographic algorithms are supported; you specify them by using SSL cipher suites. These cipher suites are supported:

SSL_RSA_EXPORT_WITH_RC4_40_MD5
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
SSL_RSA_WITH_NULL_MD5
SSL_RSA_WITH_NULL_SHA
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
SSL_DH_anon_WITH_RC4_40_MD5
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
SSL_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_DES_CBC_SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA
SSL_DH_anon_WITH_RC4_128_MD5
SSL_DH_anon_WITH_DES_CBC_SHA

An MQIPT can act as an SSL client, an SSL server, or both, and each MQIPT route can have its own set of SSL properties. See “Route section reference information” on page 53 for more details.

The key ring file on the SSL server side contains its server certificate and trusted CA (Certificate Authority) for client authentication. The key ring file on the SSL client side contains its client certificate and list of trusted CAs for server authentication. A key ring file can optionally contain a list of CRLs (Certification Revocation Lists).

See the `mqiptSample.conf` file for an example of an SSL client-server configuration.

Both SSL server and SSL client can be independently configured to accept certificates only from known identities, by using the “distinguished name” properties - for example, common name (CN=) or organization (O=).

Trust settings

As mentioned above, a key ring file contains a personal certificate including the signer certificate or chain of signer certificates. To enable authentication when a connection is being made, a certificate needs a trust setting. There are two levels of trust:

Trust as peer
Means that only this certificate may be trusted, but not any certificate signed by this certificate.

Trust as CA
Means that any certificate signed by this certificate may be trusted.

So, for server authentication, the key ring file on the client side must contain a trusted certificate. If using trust as peer, the server must present the same certificate. If using trust as CA, the server must present a certificate signed by this trusted certificate.

For client authentication, the same trust settings are true for the certificates in the server key ring file.

Finally, an alternative is to use self-signed certificates similar to those in the sample key ring files provided with MQIPT.

A utility, KeyMan, can be used to define trust settings as a certificate in a key ring file. For more information about KeyMan, see:

`http://www.alphaworks.ibm.com/tech/keyman`

Testing SSL

To test an SSL connection between two MQIPTs, a sample key ring file is provided, called `sslSample.pfx`. It contains a self-signed certificate and the trusted CA certificate.

This test key ring file can be used by both SSL client and SSL server sides of MQIPT, but must be used only for test purposes. You must create and maintain your own key ring files when running in a production environment.

The file `sslSample.pwd` contains the password to open the PKCS#12 token stored in the test key ring file.

Certificates and certificate management technologies are available from a number of vendors, including:

- RSA Security (www.rsasecurity.com)
- Entrust Technologies (www.entrust.com)
- Verisign (www.verisign.com)
- iPlanet (www.iplanet.com)

Network Dispatcher support

MQIPT can be used with the IBM Network Dispatcher to provide enhanced availability and load balancing across many servers by the use of custom advisors. This section assumes that you are familiar with Network Dispatcher and custom advisors.

Two custom advisors are supplied with MQIPT; they can be found in the `lib` subdirectory. Follow the instructions in the *Network Dispatcher User's Guide* (GC31-8496) for installing custom advisors. Figure 6 on page 22 shows an example of the use of the Network Dispatcher for monitoring port address 1414 for MQIPT. Note that each MQIPT must have the same configuration file.

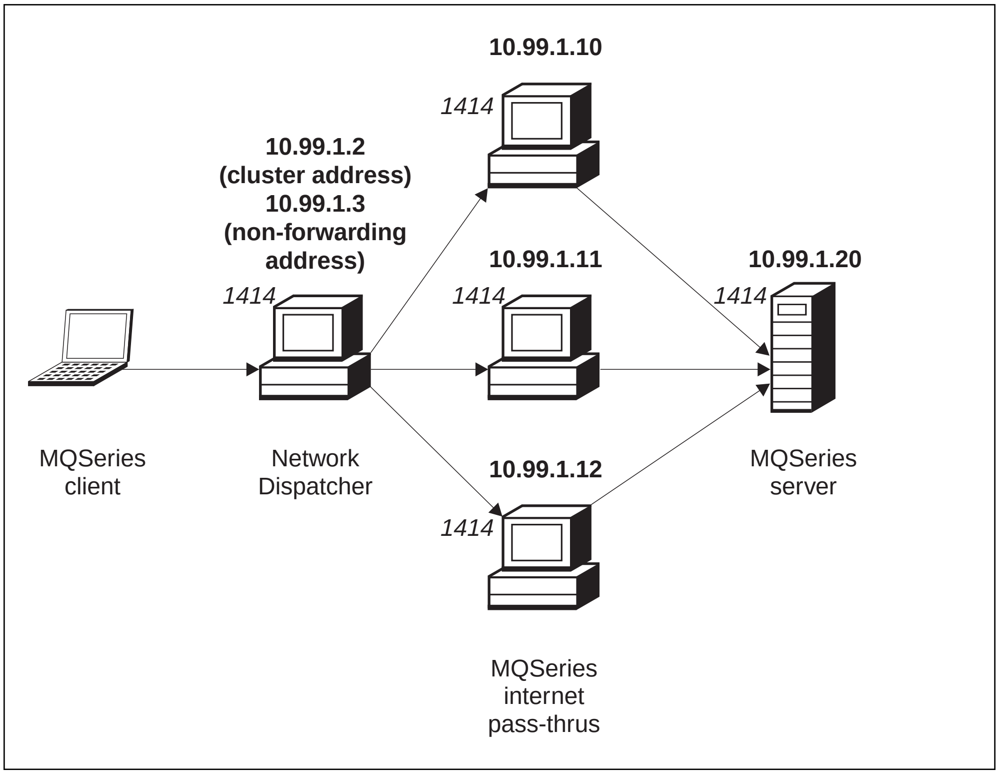


Figure 6. Using the Network Dispatcher with MQIPT

Follow the instructions in Chapter 5 of the *Network Dispatcher User's Guide* for configuring the dispatcher component to define port 1414 and the load-balanced server machines. You can use either the menu options of the Administration Client or the “ndcontrol” line mode command. For example:

```
ndcontrol port add 10.99.1.2 : 1414
ndcontrol server add 10.99.1.2 : 1414 : 10.99.1.10
ndcontrol server add 10.99.1.2 : 1414 : 10.99.1.11
ndcontrol server add 10.99.1.2 : 1414 : 10.99.1.12
```

The route definition in the MQIPT configuration file would look like this:

```
[route]
ListenerPort=1414
Destination=10.99.1.20
DestinationPort=1414
NDAdvisor=true
```

You can start (and stop) a custom advisor only from the command line. For example:

```
ndcontrol advisor start mqipt_normal 1414
```

This command starts the MQIPT advisor in “normal” mode, in which the base advisor performs its own timings to calculate the weighting factors of each MQIPT. To use the MQIPT advisor in “replace” mode, add this line to the MQIPT route definition:

```
NDAdvisorReplaceMode=true
```

You must also start the `mqipt_replace` custom advisor instead of `mqipt_normal`. For example:

```
ndcontrol advisor start mqipt_replace 1414
```

When using an advisor to monitor an SSL listener port (that is, it has `SSLServer=true` in the `mqipt.conf` configuration file), you must place a "trigger" file in the working directory of the Network Dispatcher. This "trigger" file has a specific name, relating to the route being monitored. For example, if route 1414 has `SSLServer=true`, a file called `mqipt1414.ssl` must be placed in the `c:\winnt\system32` directory (on Windows NT). See the `mqipt1414Sample.ssl` file for more information.

Supported channel configurations

All MQSeries channel types are supported, but configuration is restricted to TCP/IP connections. To an MQSeries client or queue manager, MQIPT appears as if it is the destination queue manager. Where channel configuration requires a destination host and port number, the MQIPT host name and listener port number are specified.

Client/server channels

MQIPT listens for incoming client connection requests, and then forwards them (either using HTTP tunneling, SSL, or as standard MQSeries protocol packets). If MQIPT is using HTTP tunneling or SSL it forwards them on a connection to a second MQIPT. If it is not using HTTP tunneling, it forwards them on a connection to what it sees as the destination queue manager (although this could in turn be a further MQIPT). Once the destination queue manager has accepted the client connection, packets are relayed between client and server.

Sender/receiver

If MQIPT receives an incoming request from a sender channel, it forwards it to the next MQIPT or destination queue manager in exactly the same way as for client connection channels. The destination queue manager validates the incoming request and starts the receiver channel if appropriate. All communications between sender and receiver channel (including security flows) are relayed.

Requester/server

This combination is handled in the same manner as the types above. Validation of the connection request is performed by the server channel at the destination queue manager.

Requester/sender

The ‘callback’ configuration could be of use if the two queue managers are not allowed to establish direct connections to each other, but are both allowed to connect to MQIPT and to accept connections from it.

Server/requester and server/receiver

These are handled by MQIPT just like the Sender/Receiver configuration.

Normal termination and failure conditions

When MQIPT detects closure (either normal or abnormal) of an MQSeries channel, it propagates the channel closure. If the administrator closes down a route through the MQIPT, all channels going through that route are closed.

MQIPT provides an optional idle time-out facility. If MQIPT detects that a channel has been idle for a period of time exceeding the timeout, it performs an immediate shutdown on the two connections in question.

The two MQSeries systems at either end of the channel observe these abnormal termination conditions either as network failures, or as termination of the channel by their partner. The channels in question are then able to restart and recover (if the failure happens during a protocol in-doubt period) just as they would do if there were no MQIPTs being used.

Other security considerations

If you choose not to use SSL, MQIPT allows channel security flows, so that MQSeries channel exits can be used to provide security over the entire channel from end to end.

MQIPT has several additional functions that help a designer build a secure solution:

- If there are many clients in an internal network all trying to make outgoing connections, they can all go through an MQIPT located inside the firewall. The firewall administrator then has to grant external access only to the MQIPT machine.
- MQIPT can connect only to queue managers for which it has been explicitly configured in its configuration file.
- MQIPT provides a connection log. When enabled, this facility logs all connections, successful or otherwise, detailing the host from which the connection was made, and the responding hostname. It also logs connection timeouts and disconnects.
- MQIPT verifies that the messages it receives and transmits are valid and conform to the MQSeries protocol. This helps prevent MQIPTs being used for security attacks outside of the MQSeries protocol.
- It allows channel exits to run their own end-to-end security protocols.
- MQIPT allows you to restrict the total number of incoming connections. This helps protect a vulnerable internal queue manager from denial-of-service attacks.

You must protect the MQIPT’s configuration file, `mqipt.conf`, because this file controls access to the internal hosts, and you must prevent unauthorized access to the command port (if it is enabled) because such access allows an external person to shut down MQIPT.

Chapter 3. Upgrading from the previous version

To upgrade MQIPT from Version 1.0 to Version 1.1, follow these steps:

1. Take a copy of the configuration file `mqipt.conf` and save it in a different location from the MQIPT home directory.
2. Stop MQIPT by running the command:

   ```
   mqiptAdmin -stop
   ```
3. If you have installed MQIPT as a service, you must remove it before uninstalling MQIPT:

   ```
   mqiptService -remove
   ```
4. Run the uninstallation program for MQIPT.
5. After you have installed MQIPT V1.1, copy the saved configuration file back to the MQIPT home directory. The file is compatible with V1.1. The new `mqiptSample.conf` file shows you the new properties you might want to use.
6. You are advised to use the new MQIPT Administration GUI to manage changes to MQIPT. The configuration file from V1.0 is compatible with the new GUI.

New configuration options

The following properties are new in Version 1.1:

AccessPW
HTTPProxy
HTTPProxyPort
NDAdvisor
NDAdvisorReplaceMode
SSLClient
SSLClientCipherSuites
SSLClientConnectTimeout
SSLClientDN_C
SSLClientDN_CN
SSLClientDN_L
SSLClientDN_O
SSLClientDN_OU
SSLClientDN_ST
SSLClientKeyRing
SSLClientKeyRingPW
SSLServer
SSLServerAskClientAuth
SSLServerCipherSuites
SSLServerDN_C
SSLServerDN_CN
SSLServerDN_L
SSLServerDN_O
SSLServerDN_OU
SSLServerDN_ST
SSLServerKeyRing
SSLServerKeyRingPW

For reference information about all the properties, see “Configuration reference information” on page 49.

Chapter 4. Installing internet pass-thru on Windows NT or Windows 2000

This chapter describes how you install MQIPT on a Windows NT or Windows 2000 system:

Downloading and installing the files

MQIPT is downloaded from the MQSeries SupportPac Web page, at:

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

Follow the instructions for downloading.

Open a command prompt and unpack `ms81_nt.zip` into a temporary directory. Run the `setup.exe` and follow the online instructions.

MQIPT must be installed by a user with Administrator authority.

MQIPT contains the files shown in the following table and the files for the Administration Client GUI, shipped as a separately installable feature, shown in the next table.

File	Purpose
Readme.txt	Latest information not included in the publications
mqiptSample.conf	Sample configuration file
sslSample.pfx	Test key ring file
sslSample.pwd	Password file for test key ring file
lib\MQipt.jar	Contains runtime, class, and property files
lib\ADV_mqipt_normal.class	Network Dispatcher advisor for “normal” mode
lib\ADV_mqipt_replace.class	Network Dispatcher advisor for “replace” mode
lib\mqipt1414Sample.ssl	Sample trigger file for Network Dispatcher advisor
bin\mqipt.bat	Shortcut for running MQIPT from the command line
bin\mqiptadmin.bat	Shortcut for stopping MQIPT and refreshing file information
bin\mqiptservice.exe	For adding or removing MQIPT to or from the Windows Service Control Manager
bin\mqiptservice118.exe	For adding or removing MQIPT to or from the Windows Service Control Manager, if using JDK at a level earlier than 1.2.0
bin\mqiptVersion.bat	Displays the version number of MQIPT
doc\passthru.pdf	The *internet pass-thru* manual in PDF format
doc\en_US\passthru.htm	Master file for the *internet pass-thru* manual in HTML format

The files associated with the Administration Client GUI feature are:

File	Purpose
lib\guiadmin.jar	Contains runtime, class and property files
lib\swing.jar	Contains the Java Swing library which supports the graphical user interface
bin\mqiptGui.bat	Shortcut for running the Administration Client from the command line
bin\customSample.properties	Sample file for customizing the appearance and, therefore, accessibility of the Administration Client
doc\guiadmin.pdf	Online help file

The installer updates the system CLASSPATH environment variable with the location of the `MQipt.jar`, `swing.jar`, and `guiadmin.jar` files.

Setting up internet pass-thru

Before starting MQIPT for the first time, copy the sample configuration file, mqiptSample.conf, to mqipt.conf. See Chapter 8, “Administering and configuring internet pass-thru” on page 44 for configuration and administration information.

Starting internet pass-thru from the command line

Open a command prompt and change directory to the bin directory and run mqipt. For example:

```
c:
cd \mqipt\bin
mqipt ..
```

You can also start MQIPT from the Windows Start -> Programs menu.

Running the mqipt script without any options uses a default location of “.” for the configuration file (mqipt.conf). To specify a different location:

```
mqipt <directory name>
```

To route all outbound connections through a SOCKS proxy, specify the host name and port address on:

```
mqipt .. <socksHostName <socksPort>>
```

The default socksPort is 1080.

Messages will appear on the console showing the status of MQIPT. If an error occurs, see “Problem determination” on page 59. The following messages are an example of MQIPT successfully starting:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2001 All Rights Reserved
MQCPI001 MQSeries internet pass-thru Version 1.1 starting
MQCPI004 Reading configuration information from c:\mqipt\mqipt.conf
MQCPI008 Listening for control commands on port 1881
MQCPI011 The path c:\mqipt\logs will be used to store the log files
MQCPI006 Route 1418 has started and will forward messages to :
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....using MQ protocols
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file c:\mqipt\KeyMan.pfx
MQCPI038 ......distinguished name(s) CN=*Doe O=IBM OU=* L=* ST=* C=*
```

The following subdirectories of the mqipt home directory are created automatically when MQIPT is invoked for the first time:

- A "logs" directory in which the connection log is kept
- An "errors" directory in which any First Failure Support Technology™ (FFST™) and trace records are written

Starting the Administration Client from the command line

Open a command prompt and change directory to the `bin` directory and run `mqiptGui`. For example:

```
c:
cd \mqipt\bin
mqiptGui
```

To allow the Administration Client to connect outwards through a firewall to an MQIPT using a SOCKS proxy, specify the host name or address and port number:

```
mqiptGui <socksHostName <socksPort>>
```

The default `socksPort` is 1080.

The status of the Administration Client is shown by messages appearing in the Administration Client's main window.

Using a Windows service control program

A separate service control program, `mqiptservice.exe`, is provided to allow MQIPT to be managed and started as a Windows service.

`mqiptservice.exe` takes the following command-line arguments:

mqiptservice -install *path*
Installs and registers the service, so that it appears on the Windows services panel as a manual service. Go to the services panel and change the setting to “automatic” to make MQIPT start automatically when the system starts. You have to reboot Windows after installing this service. The path parameter, which must be supplied, is the fully-qualified path to the directory containing the `mqipt.conf` configuration file. Put quotes around the path name if the name contains blanks.

mqiptservice -remove
Removes the service, making it disappear from the services panel.

mqiptservice ?
Displays US English help messages listing the valid arguments.

Specifying both install and remove on the same command causes an error.

Windows internally invokes the `mqiptservice` program with no arguments. If you call it from the command line with no arguments, the program times out and returns with an error.

When the MQIPT service is started, all active MQIPT routes start up. When it is stopped, all routes are subjected to immediate shutdown.

Notes:

1. If you have a release of JDK earlier than 1.2.0, use `mqiptservice118.exe` instead of `mqiptservice.exe`.
2. It is assumed that the system PATH environment variable contains the location of the JNI runtime libraries. For JDK 1.20 and above, this file (`jvm.dll`) can be found in the `classics` subdirectory of the JDK. For older releases of the JDK, file `javai.dll` can be found in the `bin` subdirectory.

Uninstalling internet pass-thru as a Windows service

You uninstall MQIPT as a service by first stopping it from the Windows services panel. Then open a command prompt, go to MQIPT's `bin` subdirectory, and type:

```
mqiptservice -remove
```

Note: If you have a release of JDK earlier than 1.2.0, use

```
mqiptservice118 -remove
```

.

Uninstalling internet pass-thru

Before uninstalling MQIPT from your system, remove it as a Windows Service, as described above. Then run the uninstall process from the Windows Start menu.

Chapter 5. Installing internet pass-thru on Sun Solaris

This chapter describes how you install MQIPT on a Sun Solaris system:

Downloading and installing the files

MQIPT is downloaded from the MQSeries SupportPac Web page, at:

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

Follow the instructions for downloading.

Log in as `root`, uncompress and unpack `ms81_sol.tar.Z` into a temporary directory. Run the `pkgadd` command, as in this example:

```
login root
cd /tmp
uncompress -fv ms81_sol.tar.Z
tar xvf ms81_sol.tar
pkgadd -d . mqipt
```

The example assumes that `ms81_sol.tar.Z` is in the /tmp directory.

MQIPT contains the files shown in the following table, including the files for the Administration Client GUI.

File	**Purpose**
Readme.txt	Latest information not included in the publications
mqiptSample.conf	Sample configuration file
sslSample.pfx	Test key ring file
sslSample.pwd	Password file for test key ring file
lib/MQipt.jar	Contains runtime, class, and property files
lib/ADV_mqipt_normal.class	Network Dispatcher advisor for “normal” mode
lib/ADV_mqipt_replace.class	Network Dispatcher advisor for “replace” mode
lib/mqipt1414Sample.ssl	Sample trigger file for Network Dispatcher advisor
bin/mqipt	Shortcut for running MQIPT from the command line
bin/mqiptAdmin	Shortcut for stopping MQIPT and refreshing file information
bin/mqiptVersion	Display the version number of MQIPT
bin/mqiptService	For installing MQIPT so that it starts automatically at system startup.
bin/mqiptEnv	Defines the location of the `mqipt.jar` file and is used only by the other scripts.
doc/en_US/passthru.htm	Master file for the *internet pass-thru* manual in HTML format
lib/mqiptGui.jar	Contains runtime, class and property files for the Administration Client GUI
lib/swing.jar	Contains the Java Swing library which supports the graphical user interface for the Administration Client GUI
bin/mqiptGui	Shortcut for running the Administration Client GUI from the command line
bin/customSample.properties	Sample file for customizing the appearance and, therefore, accessibility of the Administration Client
doc/guiadmin.pdf	Online help file

Setting up internet pass-thru

Before starting MQIPT for the first time, copy the sample configuration file, `mqiptSample.conf`, to `mqipt.conf`. See Chapter 8, “Administering and configuring internet pass-thru” on page 44 for configuration and administration information.

Starting internet pass-thru from the command line

Log in as `root` and change directory to the `bin` directory. For example:

```
cd /opt/mqipt/bin
mqipt ..
```

Running the `mqipt` script without any options uses a default location of “.” for the configuration file (`mqipt.conf`). To specify a different location:

```
mqipt <directory name>
```

To route all outbound connections through a SOCKS proxy, specify the host name and port address on

```
mqipt .. <socksHostName <socksPort>>
```

The default `socksPort` is 1080.

Messages will appear on the console showing the status of MQIPT. If an error occurs, see “Problem determination” on page 59. The following messages are an example of MQIPT successfully starting:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2001 All Rights Reserved
MQCPI001 MQSeries internet pass-thru Version 1.1 starting
MQCPI004 Reading configuration information from /opt/mqipt/mqipt.conf
MQCPI008 Listening for control commands on port 1881
MQCPI011 The path /opt/mqipt/logs will be used to store the log files
MQCPI006 Route 1418 has started and will forward messages to :
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....using MQ protocols
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file /opt/mqipt/KeyMan.pfx
MQCPI038 ......distinguished name(s) CN=*Doe O=IBM OU=* L=* ST=* C=*
```

The following subdirectories of the `mqipt` home directory are created automatically when MQIPT is invoked for the first time:

- A "logs" directory in which the connection log is kept
- An "errors" directory in which any First Failure Support Technology (FFST) and trace records are written

Starting internet pass-thru automatically

To start MQIPT automatically when the system is started, run the `mqiptService` script. For example:

```
cd /opt/mqipt/bin
mqiptService -install
```

To prevent MQIPT from starting automatically:

```
cd /opt/mqipt/bin
mqiptService -remove
```

Starting the Administration Client from the command line

Open a command prompt and change directory to the `bin` directory and run `mqiptGui`. For example:

```
cd /opt/mqipt/bin
mqiptGui
```

To allow the Administration Client to connect outwards through a firewall to an MQIPT, specify the host name or address and port number:

```
mqiptGui <socksHostName <socksPort>>
```

The default `socksPort` is 1080.

The status of the Administration Client is shown by messages appearing in the Administration Client's main window.

Uninstalling internet pass-thru

Before uninstalling MQIPT from your system, prevent it from starting automatically, as described in “Starting internet pass-thru automatically” on page 34. Log in as `root` and run the `pkgrm` command:

```
pkgrm mqipt
```

Chapter 6. Installing internet pass-thru on AIX

This chapter describes how you install MQIPT on an AIX system:

- "Downloading and installing the files"
- "Setting up internet pass-thru" on page 37
- "Starting internet pass-thru from the command line" on page 37
- "Starting internet pass-thru automatically" on page 38
- "Starting the Administration Client from the command line" on page 39
- "Uninstalling internet pass-thru" on page 39

Downloading and installing the files

MQIPT is downloaded from the MQSeries SupportPac Web page, at:

http://www.ibm.com/software/ts/mqseries/downloads

Follow the instructions for downloading.

Log in as root, uncompress and unpack ms81_aix.tar.Z into a temporary directory. Run the installp command, as in this example:

```
cd /tmp
uncompress -fv ms81_aix.tar.Z
tar xvf ms81_aix.tar
installp -d . -a mqipt-RT
```

The example assumes that ms81_aix.tar.Z is in the /tmp directory.

MQIPT contains the files shown in the following table, including the files for the Administration Client GUI.

File	Purpose
Readme.txt	Latest information not included in the publications
mqiptSample.conf	Sample configuration file
sslSample.pfx	Test key ring file
sslSample.pwd	Password file for test key ring file
lib/MQipt.jar	Contains runtime, class, and property files
lib/ADV_mqipt_normal.class	Network Dispatcher advisor for “normal” mode
lib/ADV_mqipt_replace.class	Network Dispatcher advisor for “replace” mode
lib/mqipt1414Sample.ssl	Sample trigger file for Network Dispatcher advisor
bin/mqipt	Shortcut for running MQIPT from the command line
bin/mqiptAdmin	Shortcut for stopping MQIPT and refreshing file information
bin/mqiptVersion	Display the version number of MQIPT
bin/mqiptService	For installing MQIPT so that it starts automatically at system startup.
bin/mqiptEnv	Defines the location of the `mqipt.jar` file and is used only by the other scripts.
doc/en_US/passthru.htm	Master file for the *internet pass-thru* manual in HTML format
lib/mqiptGui.jar	Contains runtime, class and property files
lib/swing.jar	Contains the Java Swing library which supports the graphical user interface
bin/mqiptGui	Shortcut for running the Administration Client from the command line
bin/customSample.properties	Sample file for customizing the appearance and, therefore, accessibility of the Administration Client
doc/guiadmin.pdf	Online help file

Setting up internet pass-thru

Before starting MQIPT for the first time, copy the sample configuration file, `mqiptSample.conf`, to `mqipt.conf`. See Chapter 8, “Administering and configuring internet pass-thru” on page 44 for configuration and administration information.

Starting internet pass-thru from the command line

Log in as `root` and change directory to the `bin` directory. For example:

```
cd /usr/opt/mqipt/bin
mqipt ..
```

Running the mqipt script without any options uses a default location of "." for the configuration file (mqipt.conf). To specify a different location:

```
mqipt <directory name>
```

To route all outbound connections through a SOCKS proxy, specify the host name and port address on

```
mqipt .. <socksHostName <socksPort>>
```

The default socksPort is 1080.

Messages will appear on the console showing the status of MQIPT. If an error occurs, see "Problem determination" on page 59. The following messages are an example of MQIPT successfully starting:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2001 All Rights Reserved
MQCPI001 MQSeries internet pass-thru Version 1.1 starting
MQCPI004 Reading configuration information from /usr/opt/mqipt/mqipt.conf
MQCPI008 Listening for control commands on port 1881
MQCPI011 The path /usr/opt/mqipt/logs will be used to store the log files
MQCPI006 Route 1418 has started and will forward messages to :
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....using MQ protocols
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file /usr/opt/mqipt/KeyMan.pfx
MQCPI038 ......distinguished name(s) CN=*Doe O=IBM OU=* L=* ST=* C=*
```

The following subdirectories of the mqipt home directory are created automatically when MQIPT is invoked for the first time:

- A "logs" directory in which the connection log is kept
- An "errors" directory in which any First Failure Support Technology (FFST) and trace records are written

Starting internet pass-thru automatically

To start MQIPT automatically when the system is started, run the mqiptService script to add an entry in the inittab. For example:

```
cd /usr/opt/mqipt/bin
../mqiptService -install
```

To prevent MQIPT from starting automatically and remove its entry from inittab:

```
cd /usr/opt/mqipt/bin
../mqiptService -remove
```

Starting the Administration Client from the command line

Open a command prompt and change directory to the bin directory and run mqiptGui. For example:

```
cd /usr/opt/mqipt/bin
../mqiptGui
```

To allow the Administration Client to connect outwards through a firewall to an MQIPT, specify the host name or address and port number:

```
mqiptGui <socksHostName <socksPort>>
```

The default socksPort is 1080.

The status of the Administration Client is shown by messages appearing in the Administration Client's main window.

Uninstalling internet pass-thru

Before uninstalling MQIPT from your system, prevent it from starting automatically, as described in “Starting internet pass-thru automatically” on page 38. Log in as root and run the installp command:

```
installp -u mqipt-RT
```

Chapter 7. Installing internet pass-thru on HP-UX

This chapter describes how you install MQIPT on an HP-UX system:

- “Downloading and installing the files”
- “Setting up internet pass-thru” on page 41
- “Starting internet pass-thru from the command line” on page 42
- “Starting internet pass-thru automatically” on page 42
- “Starting the Administration Client from the command line” on page 43
- “Uninstalling internet pass-thru” on page 43

Downloading and installing the files

MQIPT is downloaded from the MQSeries SupportPac Web page, at:

http://www.ibm.com/software/ts/mqseries/downloads

Follow the instructions for downloading.

Log in as root, uncompress and unpack ms81_hp11.tar.Z into a temporary directory. Run the swinstall command, as in this example:

```
login root
cd /tmp
uncompress -fv ms81_hp11.tar.Z
tar xvf ms81_hp11.tar
swinstall -s /tmp MQIPT.MQIPT-RT
```

The example assumes that ms81_hp11.tar.Z is in the /tmp directory.

MQIPT contains the files shown in the following table, including the files for the Administration Client GUI.

File	Purpose
Readme.txt	Latest information not included in the publications
mqiptSample.conf	Sample configuration file
sslSample.pfx	Test key ring file
sslSample.pwd	Password file for test key ring file
lib/MQipt.jar	Contains runtime, class, and property files
lib/ADV_mqipt_normal.class	Network Dispatcher advisor for “normal” mode
lib/ADV_mqipt_replace.class	Network Dispatcher advisor for “replace” mode
lib/mqipt1414Sample.ssl	Sample trigger file for Network Dispatcher advisor
bin/mqipt	Shortcut for running MQIPT from the command line
bin/mqiptAdmin	Shortcut for stopping MQIPT and refreshing file information
bin/mqiptVersion	Display the version number of MQIPT
bin/mqiptService	For installing MQIPT so that it starts automatically at system startup.
bin/mqiptEnv	Defines the location of the `mqipt.jar` file and is used only by the other scripts.
bin/mqiptFork	Used to launch MQIPT during system startup
doc/en_US/passthru.htm	Master file for the *internet pass-thru* manual in HTML format
lib/mqiptGui.jar	Contains runtime, class and property files for the Administration Client GUI
lib/swing.jar	Contains the Java Swing library which supports the graphical user interface for the Administration Client GUI
bin/mqiptGui	Shortcut for running the Administration Client GUI from the command line
bin/customSample.properties	Sample file for customizing the appearance and, therefore, accessibility of the Administration Client
doc/guiadmin.pdf	Online help file

Setting up internet pass-thru

Before starting MQIPT for the first time, copy the sample configuration file, `mqiptSample.conf`, to `mqipt.conf`. See Chapter 8, “Administering and configuring internet pass-thru” on page 44 for configuration and administration information.

Starting internet pass-thru from the command line

Log in as `root` and change directory to the `bin` directory. For example:

```
cd /opt/mqipt/bin
mqipt ..
```

Running the `mqipt` script without any options uses a default location of “.” for the configuration file (`mqipt.conf`). To specify a different location:

```
mqipt <directory name>
```

To route all outbound connections through a SOCKS proxy, specify the host name and port address on

```
mqipt .. <socksHostName <socksPort>>
```

The default `socksPort` is 1080.

Messages will appear on the console showing the status of MQIPT. If an error occurs, see “Problem determination” on page 59. The following messages are an example of MQIPT successfully starting:

```
5639-L92 (C) Copyright IBM Corp. 2000, 2001 All Rights Reserved
MQCPI001 MQSeries internet pass-thru Version 1.1 starting
MQCPI004 Reading configuration information from /opt/mqipt/mqipt.conf
MQCPI008 Listening for control commands on port 1881
MQCPI011 The path /opt/mqipt/logs will be used to store the log files
MQCPI006 Route 1418 has started and will forward messages to :
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....using MQ protocols
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file /opt/mqipt/KeyMan.pfx
MQCPI038 ......distinguished name(s) CN=*Doe O=IBM OU=* L=* ST=* C=*
```

The following subdirectories of the `mqipt` home directory are created automatically when MQIPT is invoked for the first time:

- A "logs" directory in which the connection log is kept
- An "errors" directory in which any First Failure Support Technology (FFST) and trace records are written

Starting internet pass-thru automatically

To start MQIPT automatically when the system is started, run the `mqiptService` script. For example:

```
cd /opt/mqipt/bin
mqiptService -install
```

This assumes that JDK 1.3 is already installed in a directory called `/opt/java1.3`. If this is not the case, edit file `mqipt.ske` and change the PATH variable to point to the location of the JDK. You must apply this change before running the `mqiptService -install` command.

When MQIPT is started as a service, it writes a `console.log` file to the `logs` subdirectory. This subdirectory is created the first time MQIPT is run, so MQIPT must be started at least once before trying to start it as a service.

To prevent MQIPT from starting automatically:

```
cd /opt/mqipt/bin
mqiptService -remove
```

Starting the Administration Client from the command line

Open a command prompt and change directory to the `bin` directory and run `mqiptGui`. For example:

```
cd /opt/mqipt/bin
mqiptGui
```

To allow the Administration Client to connect outwards through a firewall to an MQIPT, specify the host name or address and port number:

```
mqiptGui <socksHostName <socksPort>>
```

The default `socksPort` is 1080.

The status of the Administration Client is shown by messages appearing in the Administration Client's main window.

Uninstalling internet pass-thru

Before uninstalling MQIPT from your system, prevent it from starting automatically, as described in “Starting internet pass-thru automatically” on page 42. Log in as `root` and run the `swremove` command:

```
swremove MQIPT
```

Chapter 8. Administering and configuring internet pass-thru

You configure MQIPT by making changes to the configuration file `mqipt.conf`. Do this by using the Administration Client, which is the recommended way, or by using the editor of your choice. Both techniques are described here, with reference information relevant to both:

- "Using the internet pass-thru Administration Client"
- "Using internet pass-thru line mode commands" on page 48
- "Configuration reference information" on page 49

Using the internet pass-thru Administration Client

You can use the Administration Client to configure and update one or more MQIPTs. It displays global properties for an MQIPT and route-specific properties.

The only data stored locally to the Administration Client is the list of MQIPTs, in a file called `client.conf`. Global and route properties are always retrieved from the MQIPT before they are displayed in the Administration Client.

Starting the Administration Client

Start the Administration Client by using the `mqiptGui` script found in the `bin` subdirectory of MQIPT. See the installation chapter for each platform for information about starting the Administration Client.

The first time the Administration Client is started, a dialog box is displayed, prompting you for connection information to an MQIPT. The information required is:

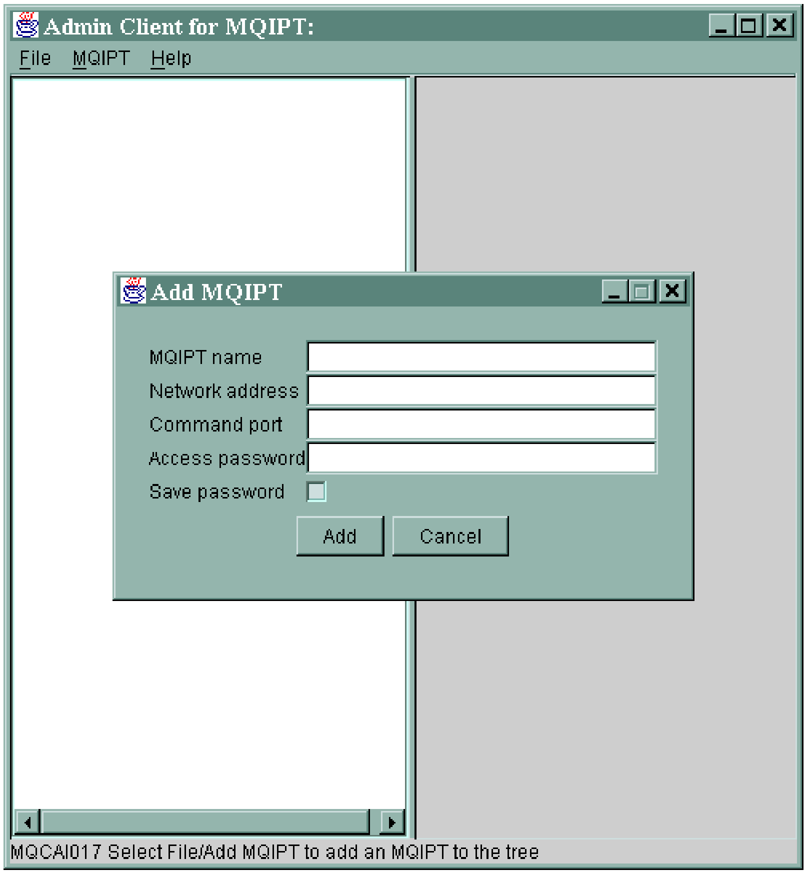
MQIPT Name
A name used to describe this MQIPT. Although this information is not essential, you are recommended to supply it.

Network Address
The address of the system on which the MQIPT resides - either a name recognized by the name server, a dotted decimal address, or localhost (if the MQIPT is on the same machine as the client).

Command Port
The number of the port on which the MQIPT is listening for commands.

Access Password
The password used when communicating with the MQIPT. Fill in this field only if password checking is in force. (Password checking is in force if the AccessPW is provided in the MQIPT configuration file and is anything other than a null string.)

Save Password
If this checkbox is left blank, the password is remembered for the duration of the session, or until the MQIPT is removed. If the checkbox is selected, the password is saved for future sessions.



MQCAI017 Select File/Add MQIPT to add an MQIPT to the tree

Figure 7. Window for first accessing an MQIPT

Administering an MQIPT

Only one MQIPT can be updated at a time, so, if another MQIPT is selected from the list, any outstanding changes must be applied before continuing. Changes made to any of the properties do not affect the MQIPT until the “Apply“ menu option is used.

Selecting an MQIPT from the list retrieves the global and route properties from the MQIPT. If the MQIPT is not running or the incorrect CommandPort has been specified, an error message is issued. Changes to the host name and CommandPort can be made from the “Connection” menu option.

Double-clicking on an MQIPT from the list displays a list of routes. Selecting a route displays its properties. You can tailor the properties to your requirements.

If you use a configuration file (`mqipt.conf`) from MQSeries internet pass-thru Version 1.0, you do not see a route name. You can add a route name by updating the Name field.

When changes are applied, the configuration file is time stamped and sent back to the MQIPT and the changes take effect immediately. Any existing comment lines are lost.

A route can be added by using the “Add Route” menu option. A set of default properties is displayed for this new route, as defined by the global properties.

The inheritance of properties

There is a hierarchy of ways in which properties of MQIPTs and routes can be set in the Administration Client:

1. Every property has a default value and if the property is not mentioned in the configuration file, or has not been specifically set by user action in the Administration Client, this default value is assumed.
2. Global properties set on MQIPTs are assumed by every route on that MQIPT unless there is specific route information to the contrary. In the configuration file, this means that properties set in the global stanza are propagated to all routes unless additional properties are set in route stanzas. Properties set by the Administration Client user on an MQIPT are propagated to all the routes unless a property is specifically set on a route.
3. Regardless of default values and global settings, any setting made against a route is sustained for that route.

File menu options

Most of the options relevant to managing the tree are shown when the File menu is selected.

Add MQIPT
Brings up the same dialog that appears when the client is first used, described in “Starting the Administration Client” on page 44.

Remove MQIPT
Removes the currently highlighted MQIPT only from the tree on the Administration Client. It does not affect the running of the MQIPT.

Save Configuration
Saves the MQIPT nodes of the tree to the Administration Client's configuration file so that they can be read back the next time it starts. Only the MQIPT nodes are saved. Global and route properties are always retrieved from the MQIPT.

Quit
Stops the Administration Client running. However, the Administration Client first checks whether the tree or the current MQIPT has changed; if either or both have, you are presented with a dialog or dialogs asking whether you wish to save the client, apply the changes to the MQIPT, or both.

MQIPT menu options

Connection
Changes an MQIPT's access parameters. The changes are reflected in the tree view. It brings up a window similar to the one described in “Starting the Administration Client” on page 44.

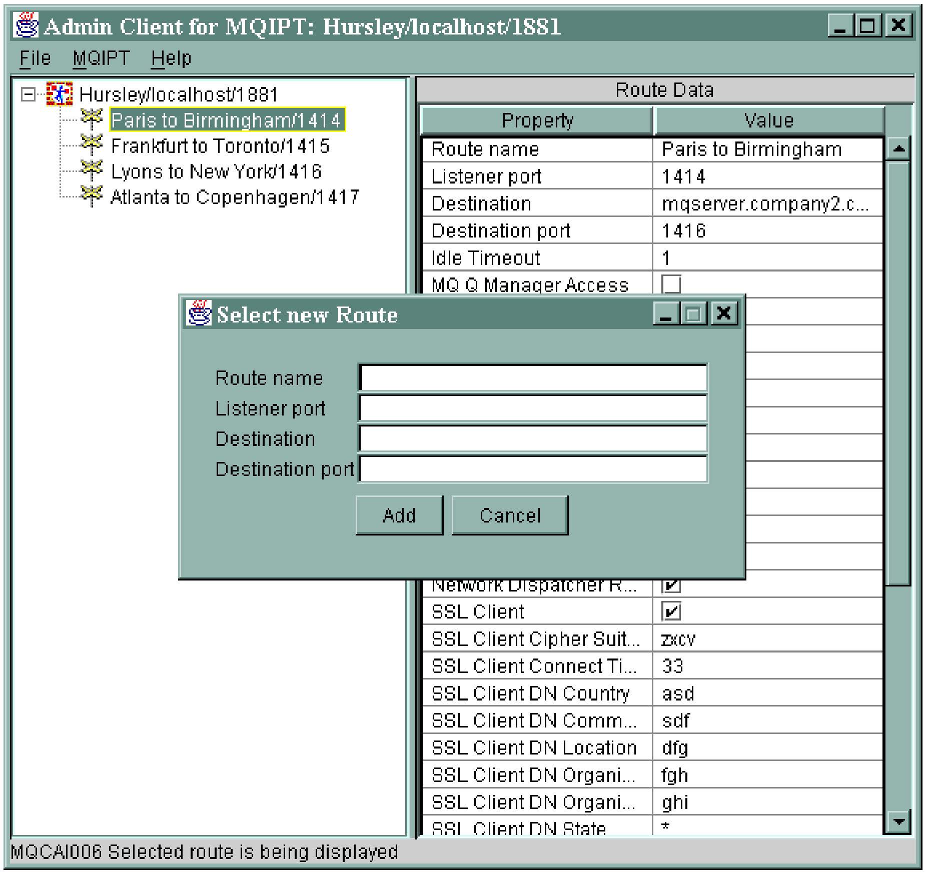
Password
Changes the password property of the remote MQIPT. This action brings up a password dialog where you are expected to make the following entries:

- **Current Password**: as a check against improper use, you must demonstrate that you know the current password before you can change it. If no password is currently in force, this field is left blank.

- **New Password**: the new password or blank if you wish to discontinue the use of passwords on this MQIPT.
- **New Password Again**: protects you against typing mistakes in the previous field by asking you to repeat the same information.
- **Save Password**: used to determine whether the new password will be saved locally, along with the other access properties of this MQIPT.

Add Route

Adds a route to the selected MQIPT. See Figure 8 for details. Each route must have a unique ListenerPort for the MQIPT.



Hursley/localhost/1881
- Paris to Birmingham/1414
- Frankfurt to Toronto/1415
- Lyons to New York/1416
- Atlanta to Copenhagen/1417

Property	Value
Route name	Paris to Birmingham
Listener port	1414
Destination	mqserver.company2.c...
Destination port	1416
Idle Timeout	1
MQ Q Manager Access	☐
Network Dispatcher R...	☑
SSL Client	☑
SSL Client Cipher Suit...	zxcv
SSL Client Connect Ti...	33
SSL Client DN Country	asd
SSL Client DN Comm...	sdf
SSL Client DN Location	dfg
SSL Client DN Organi...	fgh
SSL Client DN Organi...	ghi
SSL Client DN State	*

Route name
Listener port
Destination
Destination port

MQCAI006 Selected route is being displayed

Figure 8. Adding a route

Delete Route

Deletes the selected route from the MQIPT. The deletion does not affect the MQIPT until the “Apply” menu option is used.

Apply

When you are satisfied with the changes you have made to the MQIPT's configuration, this option sends a new configuration file to the MQIPT, which saves it. The new settings are made effective immediately.

Refresh

Reads the configuration file from the selected MQIPT and refreshes the display.

Stop

Sends a stop command to the MQIPT to tell it to stop running. After this command, you lose contact with the MQIPT. This command is ignored unless the global property RemoteShutdown is turned on.

Route information can be updated in the same way as MQIPT global information. When you change any properties of a route, you have to apply the changes before they take effect. You can do this either by selecting the “MQIPT/Apply” menu option or replying "Yes" when you are prompted about saving the configuration.

Help menu options

Help

Uses Adobe Acrobat to display information on how to use the Administration Client.

About

Briefly shows a splash window with essential information about the Administration Client.

Using internet pass-thru line mode commands

If you choose not to use the Administration Client, you can use line mode commands to administer and configure internet pass-thru.

Administering internet pass-thru using line mode commands

Using your editor of choice, change the configuration file, `mqipt.conf`, to meet your requirements. See “Configuration reference information” on page 49 for a list of the properties you can change.

If the global section of `mqipt.conf` specifies a value for `CommandPort`, MQIPT listens on this port for the following ASCII administration commands:

```
mqiptAdmin -refresh {hostname {port} }     sends the refresh command
mqiptAdmin -stop    {hostname {port} }     sends the stop command
```

The `mqiptAdmin` script is in the `bin` subdirectory.

If not provided, hostname defaults to `localhost` and port to `1881`.

STOP

MQIPT closes all connections, stops listening for incoming connections, and then exits. Using the “MQIPT/Stop” menu option of the Administration Client has the same effect. This command is ignored unless the `mqipt.conf` file specifies `RemoteShutDown=true`.

REFRESH

MQIPT re-reads `mqipt.conf`. If it finds:

- That any of the routes currently active are now marked as inactive (or are missing altogether), it closes them down and stops listening for incoming connections on those routes.
- Any routes marked active in the configuration file that it doesn’t currently have running, it starts them up.

- That the configuration parameters of a currently running route have changed, it applies the changed values to those routes. Where possible (for example, a change to the setting of trace) it does this without disruption to running connections. For some parameter changes (for example, a change to a destination), MQIPT has to close all connections before effecting the change and restarting the route.

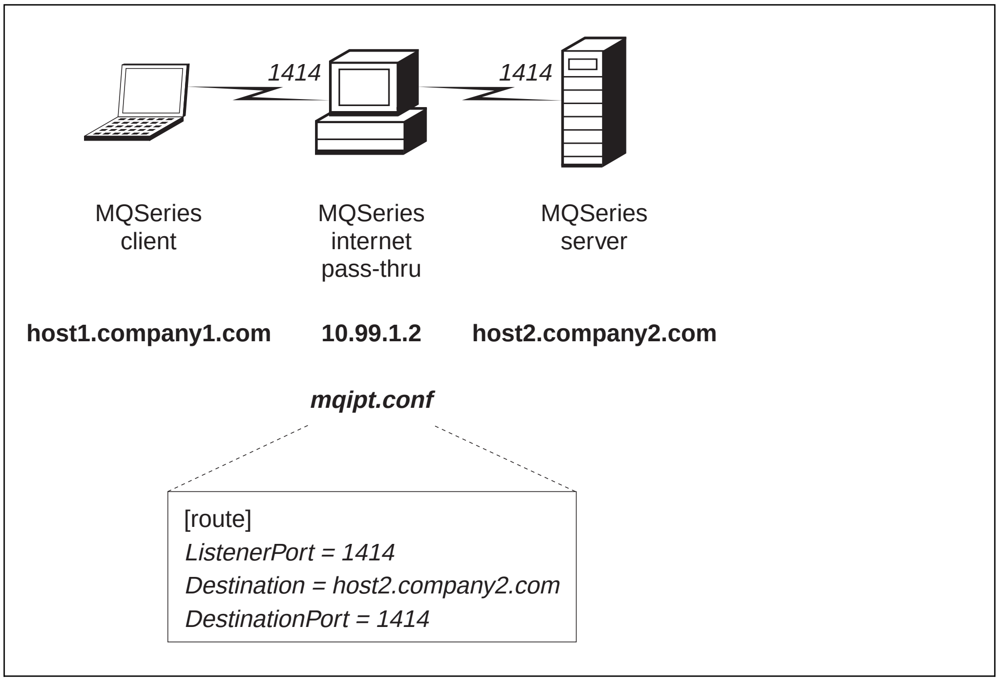
Using the “MQIPT/Apply” menu option of the Administration Client has the same effect, provided that the Administration Client has not changed any of the MQIPT's settings.

On **Windows NT and Windows 2000**, these administrative functions are also available from the Start -> Programs menu.

Configuration reference information

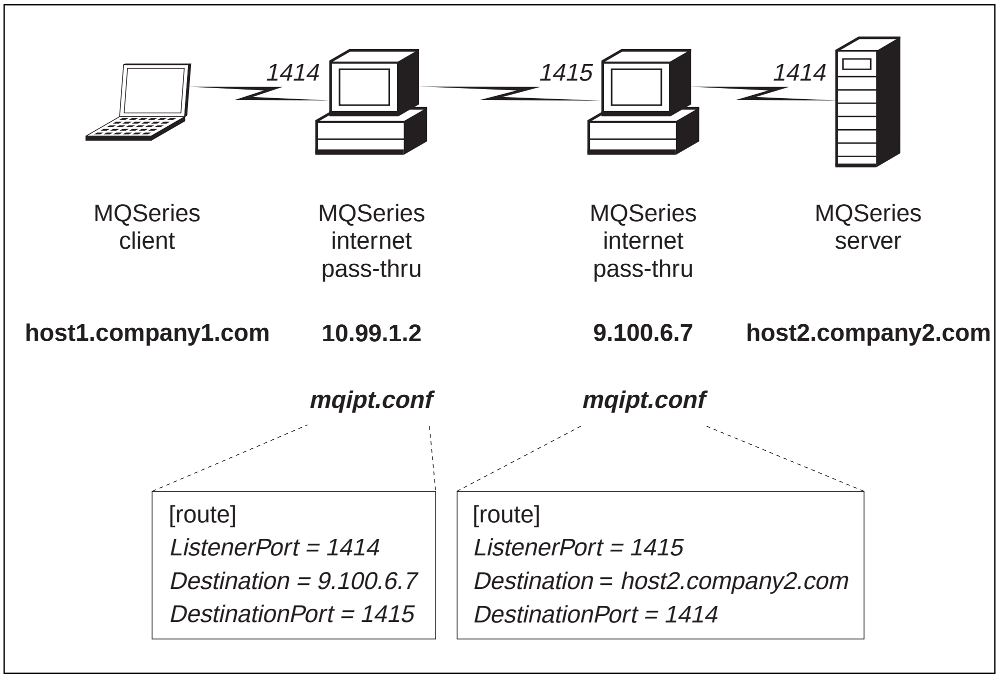
For a sample configuration, see the `mqiptSample.conf` file in the home directory of MQIPT.

The two diagrams below show how the route definitions have been defined on each MQIPT.



```
[route]
ListenerPort = 1414
Destination = host2.company2.com
DestinationPort = 1414
```

Figure 9. Example mqipt.conf definitions with one MQIPT



```
[route]
ListenerPort = 1414
Destination = 9.100.6.7
DestinationPort = 1415
```

```
[route]
ListenerPort = 1415
Destination = host2.company2.com
DestinationPort = 1414
```

Figure 10. Example mqipt.conf definitions with two MQIPTs

The `mqipt.conf` file comprises a set of sections. There is one global section, and an additional section for each route that has been defined through MQIPT. In this simple configuration, there is only one route, so the file contains two sections, one global and one route section.

Each section contains name/value property pairs. Some properties can appear only in the global sections, some can appear only in the route sections, and some may appear both in route and global sections. If a property does appear in both route and global sections, the property value in the route section overrides the global value, but only for the route in question. In this way, the global section can be used to establish the default values to be used for those properties not set in the individual route sections.

The global section starts with a line containing the characters `[global]` and ends when the first route section starts. The global section must precede all route sections in the file. Each route section starts with a line containing the characters `[route]` and ends when the next route section starts, or when the end of the configuration file is reached.

Any unrecognized keyword names (that is to say, any name/value pairs where the name is not one of the names defined in this document) are ignored. If a name/value pair appearing in a route section has a recognized name but has an invalid value (for example `MinConnectionThreads=x` or `HTTP=unsure`), that route is disabled (that is, it does not listen for any incoming connections). If a name/value pair appearing in the global section has a recognized name but has an invalid value, all routes are disabled and MQIPT does not start. Where a property is listed as taking the values `true` and `false`, any mixture of upper- and lower-case can be used.

Note that keyword names are always case sensitive; for example, **IdleTimeout** is a valid keyword, but **IdleTimeOut** is not valid and will be ignored, with no error message produced.

Summary of properties

Table 1 shows:

- All the properties
- Whether the property applies to the global section, the route section, or both
- Default values

If a property is missing from both the route section and the global section, the defaults shown in the table are used.

Table 1 (Page 1 of 2). Summary of configuration properties

Name of property	Global	Route	Default
AccessPW	yes		<blank>
CommandPort	yes		<null>[1]
ConnectionLog	yes		true
MaxLogFileSize	yes		50
RemoteShutdown	yes		false
Active	yes	yes	true
ClientAccess	yes	yes	false
Destination		yes	<null>
DestinationPort		yes	1414
HTTP	yes	yes	false
HTTPChunking[2]	yes	yes	false
HTTPProxy[2]	yes	yes	<null>
HTTPProxyPort[2]	yes	yes	8080
IdleTimeout	yes	yes	0
ListenerPort		yes	<null>
MaxConnectionThreads	yes	yes	100
MinConnectionThreads	yes	yes	5
Name		yes	<null>
NDAdvisor	yes	yes	false
NDAdvisorReplaceMode	yes	yes	false
QMgrAccess	yes	yes	true
SSLClient	yes	yes	false
SSLClientCipherSuites[3]	yes	yes	<null>
SSLClientConnectTimeout	yes	yes	30
SSLClientDN_C[3]	yes	yes	*[5]
SSLClientDN_CN[3]	yes	yes.	*[5]
SSLClientDN_L[3]	yes	yes	*[5]
SSLClientDN_O[3]	yes	yes	*[5]

Table 1 (Page 2 of 2). Summary of configuration properties

Name of property	Global	Route	Default
SSLClientDN_OU[3]	yes	yes	*[5]
SSLClientDN_ST[3]	yes	yes	*[5]
SSLClientKeyRing[3]	yes	yes	<null>
SSLClientKeyRingPW[3]	yes	yes	<null>
SSLServer	yes	yes	false
SSLServerAskClientAuth[4]	yes	yes	true
SSLServerCipherSuites[4]	yes	yes	<null>
SSLServerDN_C[4]	yes	yes	*[5]
SSLServerDN_CN[4]	yes	yes	*[5]
SSLServerDN_L[4]	yes	yes	*[5]
SSLServerDN_O[4]	yes	yes	*[5]
SSLServerDN_OU[4]	yes	yes	*[5]
SSLServerDN_ST[4]	yes	yes	*[5]
SSLServerKeyRing[4]	yes	yes	<null>
SSLServerKeyRingPW[4]	yes	yes	<null>
Trace	yes	yes	0

Notes:

1. The default is 1881 when using line mode commands.
2. Set HTTP to true for these properties to have an effect.
3. Set SSLClient to true for these properties to have an effect.
4. Set SSLServer to true for these properties to have an effect.
5. The "*" symbol represents a wildcard.

Global section reference information

The global section may contain the following properties and all the properties in “Route section reference information” on page 53, apart from ListenerPort, Destination, DestinationPort, and Name.

AccessPW

The password used when an Administration Controller sends commands to the MQIPT. If this property is not present or is set to blank, no checking takes place.

CommandPort

The TCP/IP port on which MQIPT listens for configuration commands from the `mqiptAdmin` utility or the Administration Client. You can change the command port from the Administration Client in the same way as any other property. Note that you do not change the connection properties. When you apply the new setup to the MQIPT, the Administration Client changes the connection properties automatically.

If the CommandPort property is not present, MQIPT does not listen for configuration commands. If you want to listen on the command port, you are advised to use 1881. The Administration Client does not have a default value for CommandPort, but 1881 is the default value when you use line mode commands.

ConnectionLog

Either true or false. When true, MQIPT logs all connection attempts (successful or otherwise) in the logs subdirectory and disconnection events to the file mqipt.log. The default value is true.

MaxLogFileSize

The maximum size (specified in KB) of the mqipt.log connection log file. When the mqipt.log file size increases above this maximum a backup copy mqipt.back is made, and a new file is started. Only one backup file is kept; each time the main log file fills up, any earlier backups are erased. The default value is 50, the minimum allowed value is 5.

RemoteShutDown

Either true or false. When true (and when there is a command port) MQIPT shuts down whenever a STOP command is received on the command port. The default value is false.

Route section reference information

The route section may contain the following properties:

Active

The route accepts incoming connections only if the value of Active is set to true. This means that you can temporarily shut off access to the destination, by setting Active=false, without having to delete the route section from the configuration file. If you change this property to false, the route is stopped when a REFRESH command is issued. All connections to this route are terminated.

ClientAccess

The route allows incoming client channel connections only if the value of ClientAccess is set to true. Note that potentially you can configure MQIPTs to accept client requests only, queue manager requests only, or both types of request. Use this property in conjunction with the QMgrAccess property. If you change this property to false, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

Destination

The hostname (or dotted decimal IP address) of the queue manager (or subsequent MQIPT) to which this route is to connect. Each route section **must** contain an explicit Destination value. You are allowed to have several route sections pointing to the same Destination. If a change to this property affects a route, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

DestinationPort

The port on the Destination host to which this route is to connect. It is valid for more than one route to point at the same combination of Destination and DestinationPort. Each route section **must** contain an explicit DestinationPort value. If a change to this property affects a route, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

HTTP

Set this to true for routes responsible for making outbound HTTP tunneling requests (that is, communicating with another MQIPT over HTTP). Set to false for routes directed at MQSeries queue managers. If you change this property to false, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated. To use HTTP chunking, set this property to true.

HTTPChunking

Set this to true for routes responsible for making outbound requests using HTTP tunneling with chunking. The HTTP property must also be set to true. Set to false when you are not using HTTP chunking. If you change this property to false, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

HTTPProxy

The host name (or dotted decimal IP address) of the HTTP proxy that all connections for this route use. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

HTTPProxyPort

The port address to use on the HTTP proxy. The default value is 8080. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

IdleTimeout

The time, in minutes, after which an idle connection is closed. Note that queue manager to queue manager channels also have the DISCINT property. If you set the IdleTimeout parameter, take note of DISCINT. A value of 0 indicates no idle timeout. Changes to this property take effect only when the route is restarted.

ListenerPort

The port number on which the route should listen for incoming requests. Each route section **must** contain an explicit ListenerPort value; moreover, the ListenerPort values set in each section must be distinct. Any valid port number can be used, including ports 80 and 443, provided that the ports chosen are not already in use by any other TCP/IP listener running on the same host.

MaxConnectionThreads

The maximum number of connection threads, and thus the maximum number of concurrent connections, that can be handled by this route. If this limit is reached, the MaxConnectionThreads value also indicates the number of connections that will be queued once all the threads are in use. Beyond that number, subsequent connection requests are refused. The minimum allowed value is the greater of 1 or the value of MinConnectionThreads. If a change to this property affects a route, the new value is used when the REFRESH command is issued. All connections pick up the new value immediately. The route is not terminated.

MinConnectionThreads

The minimum number of connection threads (threads to handle incoming connections on this route). This is the number of threads allocated when the route is started, and the total number of threads allocated does not drop below this value during the time the route is active. The minimum allowed value is 0 and the value must be less than that specified for MaxConnectionThreads. Changes to this property take effect only when the route is restarted.

Name
An optional name to help identify the route. It appears in console messages and tracing information. Changes to this property take effect only when the route is restarted.

NDAdvisor
Set this property to `true` for routes managed by the Network Dispatcher to allow the route to respond to requests from the custom advisor. If you change this property to `false`, the route is stopped when a REFRESH command is issued. All connections to this route are terminated. To use the NDAdvisorReplaceMode property, set this property to `true`.

NDAdvisorReplaceMode
Set this property to `true` to use the “replace” mode of the Network Dispatcher custom advisor. You must have started the `mqipt_replace custom advisor` for the ListenerPort address of this route. Set this property to `false` to use “normal” mode. You must set the NDAdvisor property to `true` to use this property.

QMgrAccess
The route allows incoming queue manager channel connections (for example sender channels) only if the value of QMgrAccess is set to the value `true`. If you change this property to `false`, the route is stopped when a REFRESH command is issued. All connections to this route are terminated.

SSLClient
Set this property to `true` to make the route act as an SSL client and make outgoing SSL connections. Setting `true` implies that the destination is another MQIPT acting as an SSL server. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLClientCipherSuites
The name of the SSL cipher suite to use on the SSL client side. This can be one or more of the supported cipher suites. If you leave this blank, the SSL client uses the supported cipher suites from the SSLClientKeyRing. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLClientConnectTimeout
Set this property to the number of seconds an SSL client will wait for an SSL connection to be accepted. If a change to this property affects a route, the new value is used when the REFRESH command is issued. The route is not terminated.

SSLClientDN_C
Use this property to accept certificates received from the SSL server of this company name. The name can be prefixed or suffixed with an asterisk (*) to extend its scope. If you do not specify this property, you imply “all company names”. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLClientDN_CN
Use this property to accept certificates received from the SSL server of this common name. The name can be prefixed or suffixed with an asterisk (*) to extend its scope. If you do not specify this property, you imply “all common

names”. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLClientDN_L

Use this property to accept certificates received from the SSL server of this location. The name can be prefixed or suffixed with an asterisk (*) to extend its scope. If you do not specify this property, you imply “all locations”. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLClientDN_O

Use this property to accept certificates received from the SSL server of this organization. The name can be prefixed or suffixed with an asterisk (*) to extend its scope. If you do not specify this property, you imply “all organizations”. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLClientDN_OU

Use this property to accept certificates received from the SSL server of this organizational unit. The name can be prefixed or suffixed with an asterisk (*) to extend its scope. If you do not specify this property, you imply “all organizational units”. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLClientDN_ST

Use this property to accept certificates received from the SSL server of this state. The name can be prefixed or suffixed with an asterisk (*) to extend its scope. If you do not specify this property, you imply “all states”. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLClientKeyRing

The fully-qualified file name of the key ring file containing the client certificate. On **Windows platforms**, you must use a double back slash (\\) as the file separator. You must specify SSLClientKeyRing if you set SSLClient to `true`. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLClientKeyRingPW

The fully-qualified file name containing the password to open the client key ring. On **Windows platforms**, you must use a double back slash (\\) as the file separator. You must specify SSLClientKeyRingPW if you set SSLClient to `true`. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLServer

Set this property to `true` to make the route act as an SSL server and accept incoming SSL connections. Setting `true` implies that the caller is another MQIPT acting as an SSL client. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLServerAskClientAuth

Use this property to request SSL client authentication by the SSL server. The SSL client must have its own certificate to send to the SSL server. The certificate is retrieved from the key ring file. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLServerCipherSuites
The name of the SSL cipher suite to use on the SSL server side. This can be one or more of the supported cipher suites. If you leave this blank, the SSL server uses the supported cipher suites from the SSLServerKeyRing. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLServerDN_C
Use this property to accept certificates received from the SSL client of this company name. The name can be prefixed or suffixed with an asterisk (*) to extend its scope. If you do not specify this property, you imply “all company names”. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLServerDN_CN
Use this property to accept certificates received from the SSL client of this common name. The name can be prefixed or suffixed with an asterisk (*) to extend its scope. If you do not specify this property, you imply “all common names”. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLServerDN_L
Use this property to accept certificates received from the SSL client of this location. The name can be prefixed or suffixed with an asterisk (*) to extend its scope. If you do not specify this property, you imply “all locations”. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLServerDN_O
Use this property to accept certificates received from the SSL client of this organization. The name can be prefixed or suffixed with an asterisk (*) to extend its scope. If you do not specify this property, you imply “all organizations”. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLServerDN_OU
Use this property to accept certificates received from the SSL client of this organizational unit. The name can be prefixed or suffixed with an asterisk (*) to extend its scope. If you do not specify this property, you imply “all organizational units”. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLServerDN_ST
Use this property to accept certificates received from the SSL client of this state. The name can be prefixed or suffixed with an asterisk (*) to extend its scope. If you do not specify this property, you imply “all states”. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLServerKeyRing
The fully-qualified file name of the key ring file containing the server certificate. On **Windows platforms**, you must use a double back slash (\\) as the file separator. You must specify SSLServerKeyRing if you set SSLServer to `true`.

If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

SSLServerKeyRingPW

The fully-qualified file name containing the password to open the server key ring. On **Windows platforms**, you must use a double back slash (\\) as the file separator. You must specify SSLServerKeyRingPW if you set SSLServer to `true`. If you change this property, the route is stopped and restarted when a REFRESH command is issued. All connections to this route are terminated.

Trace

The level of tracing required can be specified by an integer in the range 0-5. A value of 0 means no tracing; 5 requests full tracing.

If a change to this property affects a route, the new value is used when the REFRESH command is issued. All connections pick up the new value immediately. The route is not terminated.

Chapter 9. Looking after internet pass-thru

This chapter describes how to keep internet pass-thru running, under these headings:

- “Maintenance”
- “Problem determination”
- “Performance tuning” on page 62

Maintenance

You should back up the following files on a regular basis as part of your normal backup procedures:

- The configuration file, `mqipt.conf`
- The SSL key ring files, as defined by the `SSLClientKeyRing` and `SSLServerKeyRing` properties in `mqipt.conf`.
- The SSL key ring password files, as defined by the `SSLClientKeyRingPW` and `SSLServerKeyRingPW` properties in `mqipt.conf`.
- The Administration Client configuration file, `client.conf`, which contains connection information about all the MQIPTs known to the Administration Client.

Problem determination

There are some common pitfalls to check first if you encounter a problem:

- The properties entered are in the wrong case. The properties are case-sensitive.
- The MQIPT system has just been installed and has not been rebooted.
- HTTP has been set to true on a route directly connected to a queue manager.
- The CLASSPATH has not been set up correctly.
- The PATH has not been set up correctly.
- The passwords stored for the key ring files are case-sensitive.

The next step is to follow the flowchart shown in Figure 11 on page 60. The numbers refer to the following notes.

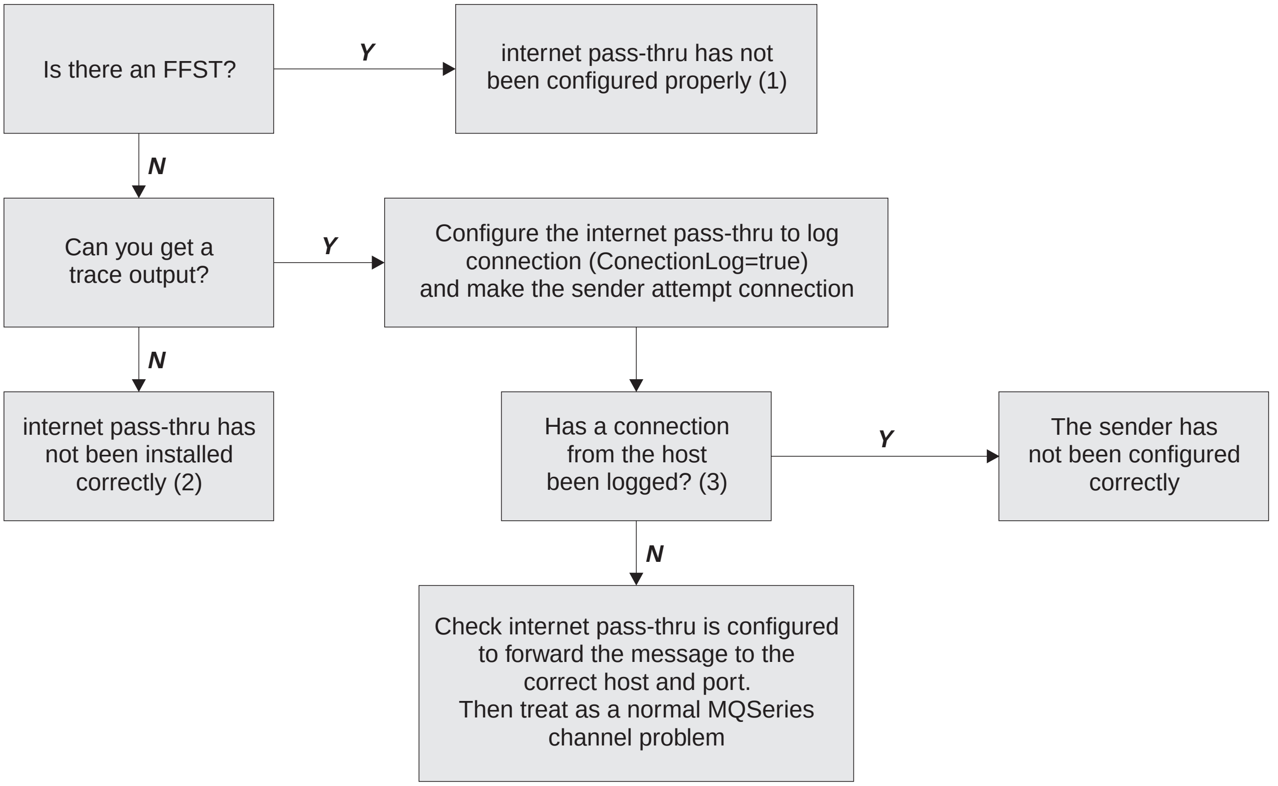

Is there an FFST?
Y
internet pass-thru has not been configured properly (1)
N
Can you get a trace output?
Y
Configure the internet pass-thru to log connection (ConectionLog=true) and make the sender attempt connection
N
internet pass-thru has not been installed correctly (2)
Has a connection from the host been logged? (3)
Y
The sender has not been configured correctly
N
Check internet pass-thru is configured to forward the message to the correct host and port.
Then treat as a normal MQSeries channel problem

Figure 11. Problem determination flowchart

Notes:

1. If you find any FFST reports (in the errors subdirectory), you know that MQIPT was correctly installed. There might have been a problem with the configuration.

   Each FFST reports a problem that causes MQIPT, or a route, to terminate its startup process. Fix the problem that caused each FFST. Then delete the old FFSTs and restart or refresh MQIPT.

2. If MQIPT has not been installed correctly, check that all the files have been put in the correct place and the CLASSPATH has been updated. To check this is correct, try to start MQIPT manually.

3. Manually starting MQIPT.

   Open a command prompt. Go to the `bin` subdirectory and type:

   ```
   mqipt xxx
   ```

   where *xxx* is the MQIPT home directory; in this case, it is “`..`”.

   This will start MQIPT and look for the configuration in the home directory. Look for any error messages and FFSTs in the `errors` subdirectory.

   Look at the text output from MQIPT for any error messages and correct the error(s). Check for FFSTs and correct any errors. MQIPT will not start if there is a problem in the global section of the configuration file. A route will not start if there is a problem in the route section of the configuration file.

Automatically starting internet pass-thru

If you install MQIPT as a Windows NT Service, and have changed its startup to be automatic, it starts when the system is brought up. Always start MQIPT manually once before trying to install MQIPT as a Windows NT Service to confirm correct installation. See “Using a Windows service control program” on page 30 for more details.

If you receive the error message “Unable to locate DLL...”, either you are using the wrong `mqiptService` program or you have not configured the system PATH environment variable correctly. PATH must contain the location of the JNI runtime libraries for JDK 1.2.0 and above. This file (`jvm.dll`) can be found in the `classics` subdirectory of the JDK. For older releases of the JDK, file `javai.dll` can be found in the `bin` subdirectory. After updating the system PATH environment variable, you must reboot the system.

| MQIPT cannot use a SOCKS proxy when started automatically.

Checking for end-to-end connectivity

If MQIPT is correctly installed, the next step is to check that the routes are set up correctly.

In the configuration file, `mqipt.conf`, set the `ConnectionLog` property to true. Start or refresh MQIPT and attempt a connection. The connect log is created in the logs directory below the home directory. If it is not created, you know that MQIPT has not been installed correctly. If no connection attempts are recorded, the sender has not been set up correctly. If attempts are recorded, check that MQIPT is forwarding the messages to the correct address.

Tracing errors

MQIPT provides a detailed execution trace facility, which is controlled by the trace attribute. Each route can be traced independently. Trace files are written to the *xxx*`\errors` directory (where *xxx* is the directory containing `mqipt.conf`. Each trace file produced has a name with the following format:

`mqiptroute`*nnnnn*`.log`

where *nnnnn* is the number of the port on which the route is listening. Trace output from threads not directly associated with any particular route (for example, the thread handling command input) is written to a separate file called `mqiptmain.log`.

Unexpected fatal errors are written as FFST records to an error log file, held in the *xxx*`\errors` directory (where *xxx* is the directory containing `mqipt.conf`). The FFST files have the following format:

`mqipt`*xxx*`.FFST`

where *xxx* is the sequence that the FFST was generated (1 is the oldest). In a long-running system, you might reach the maximum number the system can generate. In this case, any FFSTs that are generated are written to the file `mqipt0.FFST`. If the file `mqipt0.FFST` is created, you should stop and restart MQIPT at the first opportunity and delete the old files.

Performance tuning

Here are some pointers for tuning your system.

Thread pool management

The relative performance of each route can be tuned using a combination of a thread pool and an idle timeout specification.

Connection threads

Each MQIPT route is assigned a working pool of concurrently running threads that handle incoming communication requests. At initialization, a pool of threads is created (of the size specified in the route's `MinConnectionThreads` attribute), and a thread is nominated to handle the first incoming request. When this request comes in, the thread is set to work on this request immediately, and the next thread assigned as ready for the next incoming request. When all threads are assigned to work, a new thread is created, added to the working pool, and assigned for work. In this way, the pool grows until `MaxConnectionThreads` is reached. When the number of working threads is at `MaxConnectionThreads`, the next incoming request waits until a thread is released back to the working pool. This is the maximum working capacity of the route, after which no additional requests can be accepted. Threads are released back to the pool when a conversation ends, or the specified idle timeout period has elapsed.

Idle timeout

By default, working threads are not terminated because of inactivity. When a thread has been assigned to a conversation, it remains assigned to that conversation until it is closed normally, the route is deactivated, or MQIPT is shut down. Optionally, an idle timeout interval may be specified, so that any thread that has been inactive for the specified period of time (in minutes) is terminated. A monitor thread keeps a regular check on thread idle times, and terminates those that have exceeded the threshold. Threads are recycled for use by placing them back into the working pool.

Chapter 10. Messages

When run from the command line, MQIPT displays a small number of information and error messages on the console, in US English only.

Note that:

- MQCAxxxx messages are Administration Client messages.
- MQCPxxxx messages are MQIPT messages.
- MQCxIxxx messages are information messages.
- MQCxExxx messages are error messages.

MQCAE001 Unknown host: {0}

Explanation: The MQIPT host cannot be found.

User Response: Check you have correctly specified the hostname where the MQIPT is located.

MQCAE002 The following error was reported by the system: {0}

Explanation: An error has occurred. While following a system command, an error was reported.

MQCAE003 Table has not been updated

Explanation: The value that you have just entered in the table was not suitable for passing on to the model in memory.

User Response: Enter a suitable value; values are of three kinds:

- Boolean values must be either true or false
- Integer values must be numeric and, if they refer to port numbers, must be between 1 and 65535
- Any value for a string

MQCAE004 MQIPT data has not been saved

Explanation: The Administration Client was not able to save a new configuration file to the selected MQIPT.

User Response: Check the following, fix, and retry:

- MQIPT is still running
- The MQIPT host can be reached by means of TCP/IP. (You could use ping to test this.)
- The existing remote configuration file is write-protected.

MQCAE005 No valid destination address has been defined

Explanation: This can occur when specifying a route.

User Response: Enter a non-null string.

MQCAE006 No valid destination port has been defined

Explanation: This can occur when specifying a route.

User Response: Enter an integer between 1 and 65535.

MQCAE007 No valid listener port has been defined

Explanation: This can occur when specifying a route.

User Response: Enter an integer between 1 and 65535.

MQCAE008 No valid network address has been defined

Explanation: This can occur when specifying an MQIPT.

User Response: Enter a non-null string.

MQCAE009 No valid command port has been defined

Explanation: This can occur when specifying an MQIPT.

User Response: Enter an integer between 1 and 65535.

MQCAE010 Could not show online help

Explanation: The file for online help was available but could not be displayed.

User Response: Make sure that Acrobat Reader is available in the system PATH.

MQCAE011 Could not parse parameter

Explanation: There has been an internal error that caused an attempt to be made to update a nonexistent parameter in the table.

User Response: If the condition persists contact IBM Technical Support.

MQCAE012 Could not find file for online help

Explanation: File "guiadmin.pdf" could not be found.

User Response: Make sure that this file is available and in the correct path relative to file "guiadmin.jar". That is, it must be in the doc subdirectory.

MQCAE013 Interrupted while trying to show online help

User Response: Try again. Contact IBM Technical Support if the condition persists.

MQCAE014 Receiving invalid data stream from MQIPT {0}

User Response: Check that the connection attributes are correct. Contact IBM Technical Support if the condition persists.

MQCAE015 The password you have just entered has not been recognized

Explanation: The MQIPT expects a valid password, but not the one used in the last command.

MQCAE016 Node mismatch

Explanation: There is an internal inconsistency between the node selected on the tree and the data held in memory.

User Response: Contact IBM Technical Support if the condition persists.

MQCAE017 Could not create NLS text for message {0}

Explanation: No NLS text has been found for the message mentioned in this message.

User Response: Refer to advice given under the message which is being referred to. Contact IBM Technical Support if the condition persists.

MQCAE018 Could not create NLS text for message MQCAE017

Explanation: No NLS text has been found for message MQCAE017. This message appears in English only and is not translated.

User Response: If the condition persists contact IBM Technical Support.

MQCAE019 You have failed to repeat your proposed new password

User Response: Enter the proposed new password again twice.

MQCAE020 Failed to change MQIPT access parameters

Explanation: An internal error has been detected while trying to change MQIPT access parameters.

User Response: If the condition persists contact IBM Technical Support.

MQCAE021 Internal failure to identify MQIPT

Explanation: An internal error has been detected while trying to save a configuration file on an MQIPT.

User Response: If the condition persists contact IBM Technical Support.

MQCAE022 Internal failure to save MQIPT configuration

Explanation: An internal error has been detected while trying to save a configuration file on an MQIPT.

User Response: If the condition persists contact IBM Technical Support.

MQCAE023 MQIPT {0} did not recognize your password.

Explanation: The MQIPT is using an access password and you did not provide the correct one.

User Response: Correct the access password in the connection dialog.

MQCAE024 MQIPT {0} has not recognized the command.

Explanation: An error has been detected while communicating with the MQIPT.

User Response: If the condition persists contact IBM Technical Support.

MQCAE025 MQIPT {0} has failed to send configuration file.

Explanation: The MQIPT attempted to send the configuration file, but failed.

User Response: Check that the configuration file on the remote system still exists, is in the correct path, and that the MQIPT still has authority to read it.

Check that TCP/IP has not run out of resources either at the MQIPT end or the client end.

Otherwise, contact IBM Technical Support if the condition persists.

MQCAE026 Remote shutdown is disabled on MQIPT {0}.

Explanation: The MQIPT has received a request to shut down but cannot proceed because remote shutdown is not enabled.

User Response: Change the remote shutdown property and try again.

MQCAE027 Look and feel {0} is not supported.

Explanation: The recommended look and feel for the platform you are using is not available.

MQCAE028 Look and feel class {0} cannot be found. Look and feel class {0} cannot be found.

Explanation: The class file that supports the recommended look and feel for the platform you are using cannot be found.

MQCAE029 Minimum Connection Threads must be non-negative and no bigger than Maximum Connection Threads

Explanation: You are attempting to set a value which does not meet the specification.

User Response: Set a value that does meet the specification. First, make Maximum Connection Threads bigger so that the new value of Minimum Connection Threads is acceptable.

MQCAE030 Maximum Connection Threads must be greater than zero and at least as big as Minimum Connection Threads

Explanation: You are attempting to set a value that does not meet the specification.

User Response: Set a value that does meet the specification. First, make Minimum Connection Threads smaller so that the new value of Maximum Connection Threads is acceptable.

MQCAE031 Port numbers must be in the range 0 to 65535

Explanation: You are attempting to set a value that does not meet the specification.

MQCAE032 Trace must be in the range 0 to 5

Explanation: You are attempting to set a value that does not meet the specification.

MQCAE033 Max Log file size must be in the range 5 to 50

Explanation: You are attempting to set a value that does not meet the specification.

MQCAE049 No route has been selected on any MQIPT

Explanation: An attempt has been made to delete a route without first selecting the route to be deleted.

User Response: Select a route on the tree.

MQCAE050 Could not connect to MQIPT {0}

Explanation: The Administration Client could not connect to the specified MQIPT.

User Response: Check the following, fix, and retry:

- The connection properties are correct.
- MQIPT is still running.
- MQIPT is at sufficiently high release level.
- The command port is defined in the MQIPT's configuration file.
- The MQIPT host can be reached by means of TCP/IP. (You could use ping to test this.)
- There is a problem with the network.

MQCAE051 Could not read reply from MQIPT {0}

Explanation: A reply was received from the MQIPT that did not conform to the expected protocol.

User Response: Check that:

- Only one Administration Client is using the MQIPT's command port.
- MQIPT is at sufficiently high release level.

MQCAE052 Configuration has not been saved

Explanation: A valid reply was received from the MQIPT but it subsequently failed to save the configuration file.

User Response: Check that the existing remote configuration file is not write-protected.

MQCAE053 MQIPT has not confirmed saving of configuration

Explanation: The configuration file has been sent to the MQIPT but the MQIPT failed to acknowledge it.

User Response: Check the following, fix, and retry:

- Only one Administration Client is using the MQIPT's command port
- MQIPT is still running
- The command port is defined in the MQIPT's configuration file
- There is a problem with the network

MQCAE054 MQIPT data has not been refreshed

Explanation: Contact has been made with the MQIPT but the Administration Client was unable to read the configuration file.

User Response: Check the following, fix, and retry:

- MQIPT is still running.
- The configuration file still exists in the correct subdirectory.
- The MQIPT host can be reached by means of TCP/IP. (You could use ping to test this.)
- There is a problem with the network.

MQCAE055 No MQIPT or route on an MQIPT has been selected

Explanation: Your chosen menu option cannot be performed because no MQIPT or route has been selected.

User Response: Select an appropriate MQIPT or route and try again.

MQCAE056 Duplicate listener port has been rejected

Explanation: The specified listener port has been rejected because it is already being used by another route.

User Response: Choose a different listener port and try again.

MQCAI001 Usage option: (-stop |-refresh) {'hostname'{'port'}}

Explanation: This message is issued to provide guidance in use of the administration function.

MQCAI002 The MQIPT has been removed from display

Explanation: The MQIPT whose node you selected on the tree has been removed from the client’s memory.

MQCAI003 New route added to the display

Explanation: The new route that you have just specified has been added to the current MQIPT.

MQCAI004 Route has been removed from the display

Explanation: The route that you selected on the tree has been removed from the client's memory.

MQCAI005 Selected MQIPT is being displayed

Explanation: The global parameters of the MQIPT that you selected on the tree are being shown in the table.

MQCAI006 Selected route is being displayed

Explanation: The parameters of the route that you selected on the tree are being shown in the table.

MQCAI007 Client configuration has been saved

Explanation: The access parameters for all the MQIPTs on the tree have been saved.

MQCAI008 Display of online help succeeded

Explanation: The online help has been displayed as requested.

MQCAI009 Table has been updated

Explanation: The value you have just entered on the table has been used to update the model in memory.

MQCAI010 No MQIPT or route has been selected.

Explanation: No action has been taken because there is insufficient information on which to act.

MQCAI011 User Action has been cancelled

Explanation: You have cancelled out of an action, involving a pop-up window, that you had previously initiated.

MQCAI012 Selected MQIPT has been stopped

Explanation: The MQIPT currently selected on the tree has been stopped - you will not be able to communicate with it further until it has been restarted.

MQCAI013 MQIPT has been refreshed

Explanation: The information held in client memory about the MQIPT that is currently selected on the tree has been replaced with information that has been read from its configuration file.

MQCAI014 Configuration has been saved on MQIPT

Explanation: A new configuration file has been saved on the MQIPT that is currently selected on the tree, and it has been used to restart the MQIPT.

MQCAI015 Online help has terminated

Explanation: The online help has been displayed as requested and subsequently terminated.

MQCAI016 No client configuration data has been read

Explanation: This client has no knowledge of the access parameters of any MQIPTs because no client configuration was found when starting.

MQCAI017 Select File/Add MQIPT to add an MQIPT to the tree

Explanation: This message appears when there are no MQIPTs on the tree; it tells you how to add one.

MQCAI018 New MQIPT added to display

Explanation: A new MQIPT has been added to the tree as instructed.

MQCAI019 MQIPT access parameters have been changed

Explanation: The access parameters of the MQIPT that is currently selected on the tree have been changed.

MQCAI020 Dialog has been cancelled

Explanation: You have cancelled out of a dialog that you previously initiated.

MQCAI021 Select an MQIPT or route on the tree to display its contents

Explanation: This message appears when no information is being shown on the table; it tells you how to see some.

MQCAI022 The command port has changed

Explanation: The MQIPT whose command port was instructed to change has now changed.

MQCAI023 The password has changed

Explanation: Any future communication with the MQIPT which you have just changed will use the new password.

MQCAI025 MQIPT {0} has been refreshed.

Explanation: The information you hold on the MQIPT has been updated by reading its configuration file.

MQCAI026 MQIPT {0} has received shutdown request.

Explanation: The MQIPT should now shut down.

MQCPE000 Could not locate message data when handling message {0}

Explanation: Message number {0} cannot be found in the system property list.

MQCPE001 Directory does not exist or is not a directory

Explanation: At initialization, a required directory could not be found. This message refers to a directory specified either in the MQIPT configuration file mqipt.conf or in the MQIPT command line startup options on the default directory.

MQCPE002 An error occurred while trying to access the file system. The following exception was thrown: {0}

Explanation: This error occurs because a file is locked out or missing.

User Response: Read the error text and take appropriate corrective action.

MQCPE003 Unable to listen on port {0}

Explanation: MQIPT is unable to listen for requests on the specified port.

User Response: See further messages and the error log directory for full details.

MQCPE004 Route startup failed on port {0}

Explanation: It was not possible to start the route with the specified ListenerPort number. Further error messages and log records will provide further explanation of the problem.

MQCPE005 The configuration file {0} could not be found

Explanation: The MQIPT configuration file mqipt.conf could not be found in the specified directory

MQCPE006 The number of routes has exceeded {0}. MQIPT will start but this configuration is unsupported.

Explanation: Your configuration has exceeded the maximum supported number of routes for one instance of MQIPT. Operation will not be halted but the system might become unstable or overloaded as a result. Configurations that exceed the stated maximum number of routes will not be supported.

User Response: Consider starting additional instances of MQIPT with fewer routes per instance.

MQCPE007 Route not restarted on listener port {0}

Explanation: On a REFRESH operation, the route that was operating on the specified ListenerPort was not restarted on the new configuration.

MQCPE008 Duplicate route defined for listener port {0}

Explanation: More than one route has been defined with the same ListenerPort value.

MQCPE009 LogPath parameter {0} is not valid.

Explanation: The log path shown in the text either does not exist or is not accessible at the time.

MQCPE010 Listener or command port number {0} is not valid

Explanation: The port number supplied for the command port or listener port parameter is invalid.

User Response: Use a valid port number. For guidance on use of port numbers in your network, consult your network administrator.

MQCPE011 The trace level {0} is outside the valid range 0 - 5

Explanation: The specified trace option was requested, but it is not in the valid range 0-5.

MQCPE012 The value {0} is not valid for the attribute {1}

Explanation: An invalid property value has been specified.

User Response: Refer to this User Guide for full details of the valid values for each control parameter.

MQCPE013 ListenerPort property was not found in route {0}

Explanation: The ListenerPort property is the primary and unique identifier for each route, and is therefore mandatory. MQIPT has detected a route in the configuration file that does not contain a ListenerPort property.

MQCPE014 ListenerPort property value {0} is not valid

Explanation: An invalid port number has been specified for the ListenerPort property of a route. The ListenerPort property represents the TCP/IP port number on this host, on which MQIPT will accept incoming connections from an MQSeries Client, MQSeries Queue Manager or an MQIPT.

MQCPE015 No text was found for message number {0}

Explanation: An internal error has been encountered for which no description is available.

User Response: This condition should be reported to IBM Technical Support.

MQCPE016 The maximum number of connection threads is {0} but this is less than the minimum number of connection threads, which is {1}

Explanation: Your configuration has specified the minimum number of connection threads with a value exceeding the maximum number of connection threads. This could be an error in a single route, a conflict between a global property and a route property, or a route property overriding system default values.

User Response: Refer to the earlier chapters of this User Guide for full details of the valid values and applicable defaults.

MQCPE017 The exception {0} was thrown, causing MQIPT to shut down

Explanation: MQIPT has abnormally terminated and has been shut down. This may have occurred because of system environmental conditions or constraints, such as memory overflow.

User Response: If the condition persists, contact IBM Technical Support.

MQCPE018 The ListenerPort property is blank - the route will not start

Explanation: The ListenerPort number has been omitted in a route.

User Response: A value is required.

MQCPE019 The stanza {0} was not found before the following: {1}

Explanation: A sequence error has occurred in the configuration file.

User Response: See “Using internet pass-thru line mode commands” on page 48 for details of the required configuration file content and format.

MQCPE020 The new value for MaxConnectionThreads is {0}. This must be greater than the current value {1}

Explanation: After the route has started, the MaxConnectionThread property can only be increased.

MQCPE021 The property Destination was not supplied for route {0}

Explanation: The property Destination is mandatory within a route, but was omitted in the route specified.

MQCPE022 The CommandPort value {0} is outside the valid range 1 - 65535.

Explanation: The CommandPort property was outside the range 1-65535.

MQCPE023 Request for shutdown from Administration Client {0} is ignored because it is disabled.

Explanation: An attempt to shut down the MQIPT remotely has failed because remote shutdown was not enabled in the configuration file.

MQCPE024 The command received by the MQIPT controller has not been recognized.

Explanation: The MQIPT has received a command which it does not recognize through its command port.

MQCPE025 Failed to connect to server on host {0}, port {1}.

Explanation: The line mode (non-GUI) Administration Client has failed to communicate with the MQIPT.

MQCPE026 No reply received from server on host {0}, port {1}.

Explanation: The line mode (non-GUI) Administration Client has connected with the MQIPT but has not received a reply.

MQCPE027 Reply from MQIPT not recognized.

Explanation: The line mode (non-GUI) Administration Client has received a reply, which it does not recognize, from the MQIPT.

MQCPE028 Invalid stanza detected: {0}.

Explanation: The stated unrecognized stanza has been found in the configuration file.

MQCPE029 Was not able to flush log output.

Explanation: Some messages might not have been written to the log because the communication buffer could not be flushed.

MQCPE030 {0} not found in CLASSPATH.

Explanation: The specified jar file was not found in the system environment CLASSPATH variable.

User Response: MQIPT needs this jar file for execution and therefore the file must be added to the system CLASSPATH.

MQCPE031 {0} class not found.

Explanation: This message is generated when displaying the version number of MQIPT. The specified class could not be found in the MQIPT jar file or the system environment CLASSPATH variable has been corrupted.

MQCPE033 Failed to send configuration file to Administration Client at {0}

Explanation: An error occurred sending the configuration file (mqipt.conf) to the Administration Client.

MQCPE034 Administration Client at {0} did not supply the correct password.

Explanation: An access password is in use for the command port. The remote client issued a protected command without supplying the correct password.

MQCPE035 Failed to start command listener on port {0}

Explanation: An I/O error occurred starting the command listener on the specified port address.

User Response: The most likely reason for failing to start the command listener is that the designated port is already in use by another process.

MQCPE036 Invalid parameter {0}

Explanation: An unknown parameter has been given to the mqipt fork process.

User Response: Check mqiptFork script. Make sure a valid number is passed to IPTFork.

MQCPE037 Number of seconds {0} must be greater than 0

Explanation: An invalid number has been given to the mqipt fork process.

User Response: Check mqiptFork script. Make sure a positive number is passed to IPTFork.

MQCPE038 MQIPT has not started as expected

Explanation: This message is generated by the mqipt fork process, which starts MQIPT as a system service.

User Response: Check the error logs for more information. You can try increasing the sleep time IPTFork uses before it checks if MQIPT is running. Edit mqiptFork script and increase the parameter passed to IPTFork.

MQCPE039 I/O error occurred running mqipt script

Explanation: An error has occurred launching MQIPT from the fork process

User Response: Check the system PATH environment variable contains the location of the JDK and the mqipt script has execute authority.

MQCPE040 Interruption occurred running mqipt script

Explanation: An error has occurred after launching MQIPT from the fork process.

User Response: Check the error logs for more information. If the condition persists contact IBM Technical Support.

MQCPI001 {0} starting

Explanation: This MQIPT instance is beginning execution. Further initialization messages will follow.

MQCPI002 {0} shutting down

Explanation: MQIPT is going to shut down. This can result from a STOP command, or automatically if a configuration error prevents a successful startup or REFRESH action.

MQCPI003 {0} shutdown complete

Explanation: The shutdown process has completed. All MQIPT processes are now ended.

MQCPI004 Reading configuration information from {0}

Explanation: The MQIPT configuration file mqipt.conf is being read from the directory described in this message.

MQCPI005 Listener port specified as not active - {0} -> {1}({0})

Explanation: The route referred to in the message has been marked as inactive. No communication requests will be accepted on this route.

MQCPI006 Route {0} has started and will forward messages to:

Explanation: A route has been started on the listener port shown in this message. This message is followed by other messages listing any properties associated with this route.

MQCPI007 Route stopped on port {0}

Explanation: The route that was operating on the specified ListenerPort is being shut down. This action normally occurs when a REFRESH command is issued to MQIPT and the route configuration has been changed.

MQCPI008 Listening for control commands on port {0}

Explanation: This MQIPT instance is listening for control commands on the specified port.

MQCPI009 Control command received: {0}

Explanation: This message indicates that a control command has been received at the command port. Where applicable, details are included in the message.

MQCPI010 Stopping command port on {0}

Explanation: On a REFRESH operation, the command port is no longer in use in the new configuration. Commands will no longer be accepted at the specified port.

MQCPI011 The path {0} will be used to store the log files

Explanation: Logging output will be directed to the location described in this message, under the current configuration.

User Response: This may change if the configuration is amended and a REFRESH operation is requested.

MQCPI012 Changing the value of MinConnectionThreads has no effect after the route is started

Explanation: The minimum number of connection threads is assigned at route startup and cannot be changed until MQIPT is restarted.

MQCPI013 Connection from {0} to host {1} closed

Explanation: This message is issued in the connection log to record connection activity.

MQCPI014 Connection from {0} to host {1} closed - the protocol was not recognized

Explanation: This message is issued in the connection log to record connection activity.

MQCPI015 Connection from a client on {0} to host {1} was rejected because client access has been disabled on this route

Explanation: This message is issued in the connection log to record connection activity.

MQCPI016 Connection from a queue manager on {0} to host {1} was rejected because queue manager access has been disabled on this route

Explanation: This message is issued in the connection log to record connection activity.

MQCPI017 A queue manager on {0} was connected to host {1}

Explanation: This message is issued in the connection log to record connection activity.

MQCPI018 A client on {0} was connected to host {1}

Explanation: This message is issued in the connection log to record connection activity.

MQCPI019 {0} routes have been created - this exceeds the maximum number of supported routes, which is {1}

Explanation: The maximum number of supported routes has been exceeded.

User Response: MQIPT will continue to operate, but it is recommended that a second MQIPT instance is created and the routes split between the two.

| **MQCPI020 The configuration file has been sent to the Administration Client.**

| **Explanation:** As a result of a request from the Administration Client, the
| configuration file has been sent.

| **MQCPI021 Password checking has been enabled on the command port.**

| **Explanation:** This message shows that a password is required to access the
| command port.

MQCPI022 Password checking has been disabled on the command port.

Explanation: This message shows that a password is not required to access the command port.

MQCPI024 ....using HTTP proxy {0}({1})

Explanation: This message indicates that the outgoing connection for this route will be made using this HTTP proxy.

MQCPI025 The refresh requested by Administration Client {0} has finished.

Explanation: As a result of receiving a REFRESH command, the MQIPT has reread its configuration file and restarted.

MQCPI026 Administration Client {0} has requested shutdown.

Explanation: As a result of receiving a STOP command, the MQIPT is shutting down.

MQCPI027 {0} sent to {1} on port {2}

Explanation: This displays on the system console the command sent by the line mode (non-GUI) Administration Client to the designated MQIPT.

MQCPI028 Administration Controller has completed.

Explanation: This message is issued just before the MQIPT has finished its work and is closing down.

MQCPI031 ......cipher suites {0}

Explanation: This message lists the cipher suites in use for this route.

MQCPI032 ......key ring file {0}

Explanation: This message gives the file name of the key ring for this route.

MQCPI033 ......client authentication set to {0}

Explanation: This message defines whether an SSL server is requesting client authentication for this route.

MQCPI034 ....{0}({1})

Explanation: This message shows the destination and destination port address for this route.

MQCPI035 ....using {0}

Explanation: This message shows the protocol being used to the destination. It will either be MQSeries protocol, HTTP tunneling or HTTP chunking.

MQCPI036 ....SSL Client side enabled with properties :

Explanation: This message shows that the route will be using SSL to send data to the destination host.

MQCPI037 ....SSL Server side enabled with properties :

Explanation: This message shows that the route will be using SSL to receive data from the sending host.

MQCPI038 ......distinguished name(s) {0}

Explanation: This message lists the distinguished names used to control authentication of certificates.

MQCPI039 ....via Socks proxy {0}({1})

Explanation: This message shows that the outgoing connection for this route will be made using this Socks proxy, which is defined when MQIPT is started from the command line.

MQCPI040 Command port has been accessed by Administration Client {0}

Explanation: This message is written to the system console and the MQIPT log file (if logging is enabled). The MQIPT has received a connection from the Administration Client.

MQCPI041 ....will reply to Network Dispatcher advisor requests in {0} mode

Explanation: This message is written to the system console when a route is started. Used to show which mode MQIPT will use to reply to the Network Dispatcher advisor. Valid options are "Normal" and "Replace" mode.

MQCPI042 Maximum connections reached on route {0} - further requests will be blocked

Explanation: This message is written to the system console when the maximum number of connections has been reached for the given route. Further requests will be blocked until a connection becomes free or the MaxConnectionThreads value is increased.

MQCPI043 Connections on route {0} now unblocked

Explanation: This message is written to the system console when the given route is unblocked for connection requests.

MQCPI044 MQIPT has been launched from system startup

Explanation: MQIPT has been started as a system service.

MQCPI045 Launching MQIPT from system startup

Explanation: MQIPT is going to be started as a system service.

MQCPI046 Sleeping for {0} seconds while MQIPT is launched from system startup

Explanation: The fork process will sleep for this amount of time before it checks if MQIPT has started successfully as a system service.

Index

Sending your comments to IBM

MS81: MQSeries® internet pass-thru
Version 1.1

MS81 SCRIPT

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 request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should 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, use the Readers' Comment Form.
- 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:
  - IBMLink™: HURSLEY(IDRCF)
  - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication number and title
- The page number or topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.

Readers’ Comments

MS81: MQSeries® internet pass-thru
Version 1.1

MS81 SCRIPT

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

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.

Be sure to print your name and address below if you would like a reply.

Name

Address

Company or Organization

Telephone

Email

MQSeries internet pass-thru
MS81 SCRIPT

1 Cut along this line

You can send your comments POST FREE on this form from any one of these countries:

Australia
Belgium
Bermuda
Cyprus
Denmark
Finland
France
Germany
Greece
Hong Kong
Iceland
Israel
Italy
Luxembourg
Monaco
Netherlands
New Zealand
Norway
Portugal
Republic of Ireland
Singapore
Spain
Sweden
Switzerland
United Arab Emirates
United States of America

If your country is not listed here, your local IBM representative will be pleased to forward your comments to us. Or you can pay the postage and send the form direct to IBM (this includes mailing in the U.K.).

2 Fold along this line

By air mail
Par avion

IBRS/CCRI NUMBER: PHQ - D/1348/SO

NE PAS AFFRANCHIR
NO STAMP REQUIRED

REPONSE PAYEE
GRANDE-BRETAGNE

IBM United Kingdom Laboratories Limited
Information Development Department (MP 095)
Hursley Park
WINCHESTER, Hants
SO21 2ZZ United Kingdom

3 Fold along this line

From: Name ____________________
Company or Organization ____________________
Address ____________________
____
EMAIL ____________________
Telephone ____________________

1 Cut along this line

4 Fasten here with adhesive tape