MQ Internet Pass –Through (MQIPT)

- Arthur Rodriguez
- art@txmq.com
- TxMQ Inc.

MQ Internet Pass-Through (IPT)

This session will discuss the installation, administration, configuration and use of MQIPT. We will also cover how MQITP works, possible configurations, features, use cases, security considerations, upgrading to V2.1, and migrating from v2.0 to 2.1.

- Agenda
  - MQIPT Overview
  - Features
  - Security
  - Installation
  - Upgrade and migration
  - Configuration
  - Administration

MQIPT Overview

- What is MQIPT?
  - MQ Channel Protocol Forwarder (Proxy Server)
    - Simplifies the passage of WebSphere MQ channel protocols through a firewall by tunneling the protocols inside HTTP or acting as a proxy
  - A stand alone Java application that runs as a service that receives and forwards MQ channel connections
    - Between an MQ Client to queue manager
    - Between two queue managers
  - An IBM SupportPac - MS81

MQIPT Overview

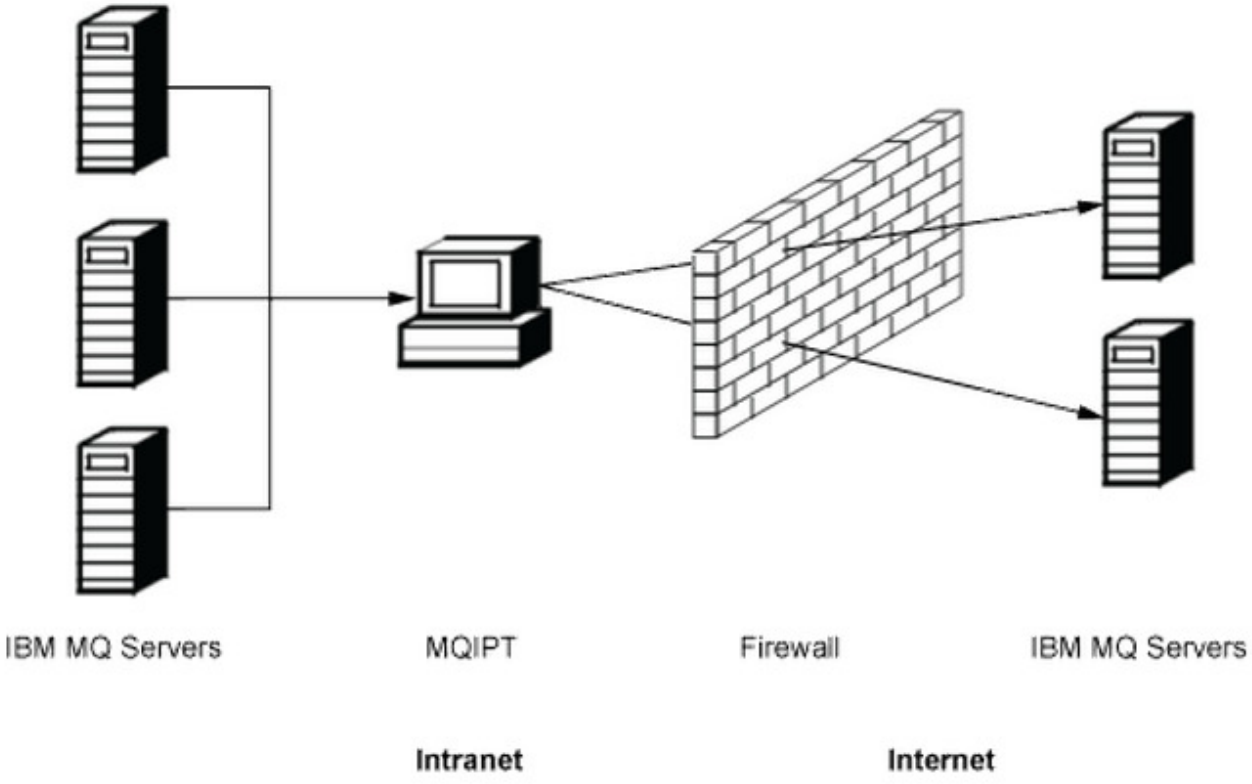
- How does MQIPT Work ?
  - MQIPT listens on a TCP/IP port and accepts connection requests from MQ channels
  - Establishes a TCP/IP connection between itself and the destination I queue manager
  - Relays all protocol packets it receives from the incoming connection to the destination queue manager
  - Returns protocol packets from the destination queue manager back to the original incoming connection
  - To use MQ ITP The connecting channel is configured with the MQIPT hostname and port in the CONNNAME of the channel.
  - MQIPT reads the incoming data and routes it to the destination queue manager based on information in it's configuration file 'mqipt.conf'
  - Other configuration fields, such as the user ID and password in a client/server channel, are also passed to the destination queue manager.

MQIPT Overview

- How can it be used ?
  - MQIPT can be used to implement messaging solutions between remote sites across the internet
  - As a channel concentrator
  - As a single point of access in the DMZ
  - To enable HTTP tunneling

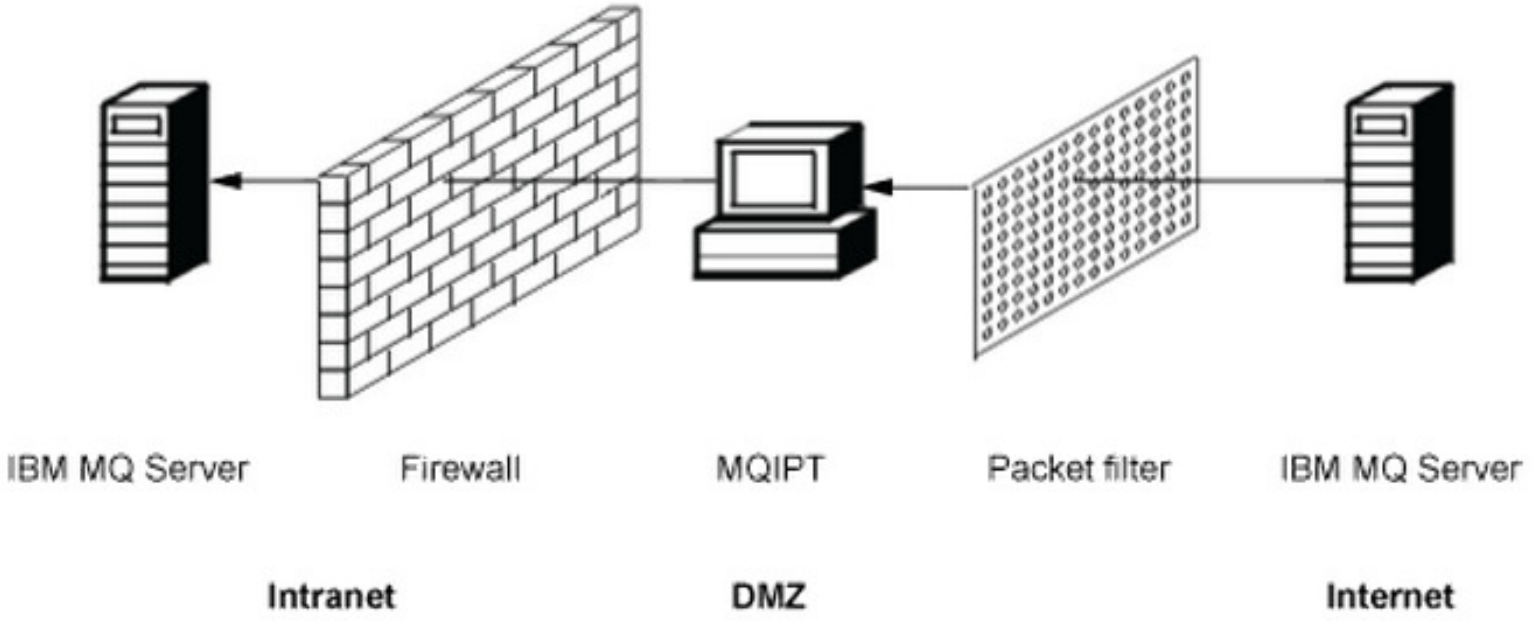
MQIPT Overview

- As a channel concentrator



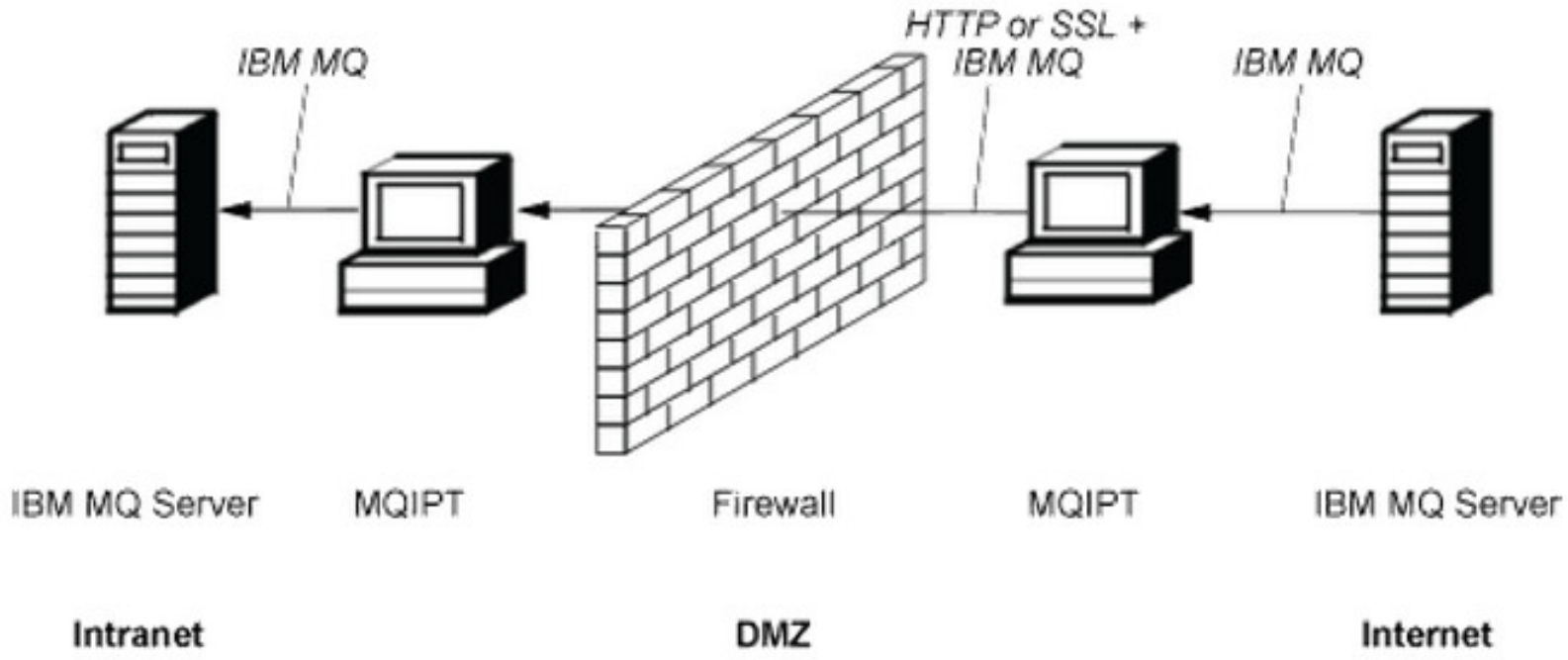
MQIPT Overview

- Single point of access in the DMZ



MQIPT Overview

- To enable HTTP tunneling



Supported MQ Channel Types

N O T E S

- Client/server channels
  - MQIPT listens for incoming client connection requests, and then forwards them by using either HTTP tunneling, SSL/TLS, or as standard IBM MQ protocol packets. If MQIPT is using HTTP tunneling or SSL/TLS 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). When the destination queue manager has accepted the client connection, packets are relayed between client and server.
- Cluster sender/receiver channels
  - If MQIPT receives an incoming request from a cluster-sender channel, it assumes the queue manager has been SOCKS-enabled and the true destination address will be obtained during the SOCKS handshaking process. It forwards the request to the next MQIPT or destination queue manager in exactly the same way as for client connection channels. This also includes auto-defined cluster-sender channels.

Supported MQ Channel Types

N O T E S

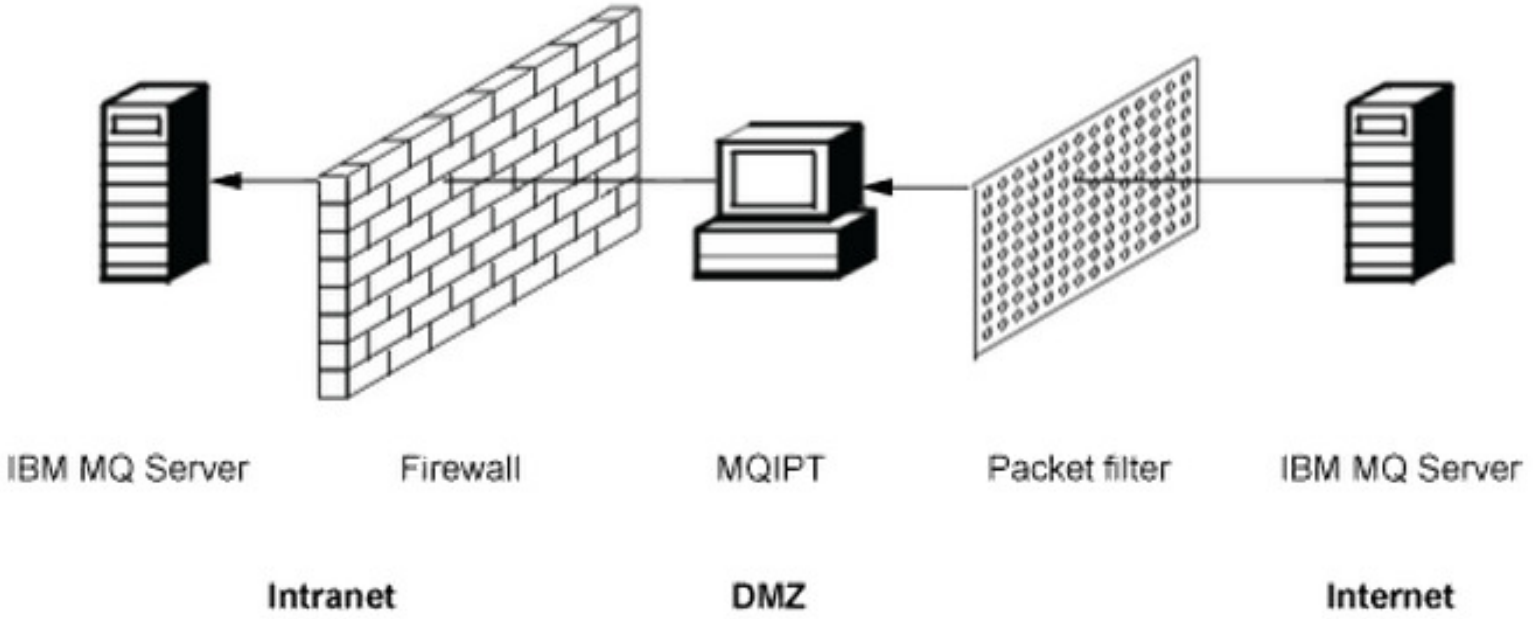
- Sender/receiver
  - If MQIPT receives an incoming request from a sender channel, it forwards it to the next MQIPT or destination queue manager in exactly the same way as for client connection channels. The destination queue manager validates the incoming request and starts the receiver channel if appropriate. All communications between sender and receiver channel (including security flows) are relayed.
- Requester/server
  - This combination is handled in the same manner as the preceding configurations. 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 in the same way that it handles the Sender/Receiver configuration

MQIPT Configuration Scenarios

Single ITP Instance

Source Protocol	MQIPT Route Mode	Destination Protocol
MQ_CHL	MQ_CHL Proxy (Default)	MQ_CHL
	MQ_CHL – SSL Client	SSL/TLS
SSL/TLS	SSL Proxy	SSL/TLS
	SSL/TLS – MQ_CHL	MQ_CHL
	SSL Server – SSL Client	SSL/TLS

MQIPT

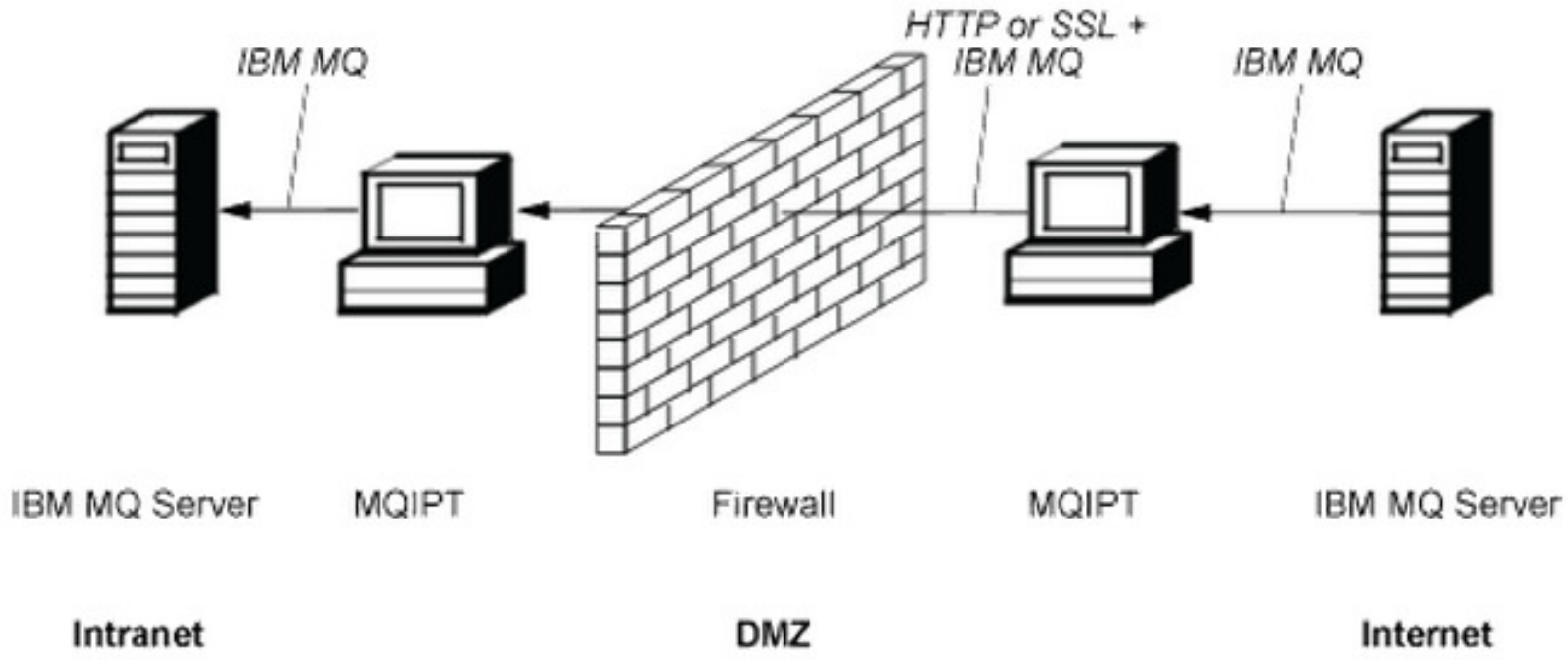

IBM MQ Server
Firewall
MQIPT
Packet filter
IBM MQ Server
Intranet
DMZ
Internet

MQIPT Configuration Scenarios

Multiple ITP Instance

Source Protocol	MQIPT Route Mode (IPT 1)	MQIPT Route Mode (IPT 2)	Destination Protocol
MQ_CHL	MQ_CHL Proxy	MQ_CHL Proxy	MQ_CHL
	FAP-server – SSL-client	SSL Proxy	SSL/TLS
		SSL Server – MQ_CHL	MQ_CHL
		SSL Server – SSL Client	SSL/TLS
	HTTP Client	HTTP Server	MQ_CHL
SSL/TLS	SSL Proxy	SSL Proxy	SSL/TLS
		SSL Server –MQ_CHL	MQ-CHL
		SSL Server – SSL Client	SSL/TLS
	HTTP Client	HTTP Server	SSL/TLS

MQIPT


IBM MQ
HTTP or SSL +
IBM MQ
IBM MQ
IBM MQ Server
MQIPT
Firewall
MQIPT
IBM MQ Server
Intranet
DMZ
Internet

MQIPT Features

- **Multiple Routes**
  - MQIPT can listen on multiple ports
  - MQIPT can connect to multiple destinations
  - MQIPT supports the configuration of multiple routes by mapping incoming ports to destination queue managers
  - Routes are configured in the mqipt.conf file
  - Up to 100 different routes can be configured
  - A single route can manage multiple connections
  - Host name & IP address are not visible to originating channel
  - Each configured route is started when MQIPT is launched

MQIPT Features

- **HTTP Support**
  - Supports sending data between two IPT instances as HTTP requests
  - IPT 1 accepts the channel request, converts the data to HTTP and sends to IPT 2
  - IPT 2 accepts the HTTP request from IPT 1, converts the data back to the original format and forwards it to the destination queue manager on the channel
  - HTTP Replies are processed in the same manner
  - The TCP/IP connection on which the HTTP requests and replies flow is persistent is kept open for the lifetime of the message channel
  - HTTP Proxy server access can be configured
    - Enable HTTP
      - HTTP=true
    - Configure HTTP Proxy
      - HTTPProxyPort=8080
      - HTTPProxy=DNS (FQDN or IP address)

MQIPT Features

- SOCKS Support
  - MQIPT can act as a SOCKS proxy by enabling the *SocksServer* property in *mqitp.conf*
  - Allows SOCKS-enabled MQ applications to connect through MQIPT to a remote IBM MQ queue manager
  - Target destination and destination port address are obtained during the SOCKS handshaking process
  - MQIPT can act as a SOCKS client to MQ applications that are not SOCKS enabled
    - Useful for firewalls that allow outbound connections only via a SOCKS proxy
  - Each MQIPT route can be configured to communicate with a different SOCKS proxy

MQIPT Features

- **SOCKS Support**

- **MQIPT can act as a SOCKS proxy by enabling the *SocksServer* property in *mqitp.conf***
  - **Allows SOCKS-enabled MQ applications to connect through MQIPT to a remote IBM MQ queue manager**
  - **Target destination and destination port address are obtained during the SOCKS handshaking process**
  - **MQIPT can act as a SOCKS client to MQ applications that are not SOCKS enabled**
    - **Useful for firewalls that allow outbound connections only via a SOCKS proxy**
  - **Each MQIPT route can be configured to communicate with a different SOCKS proxy**

MQIPT Features

- SSL/TLS Support
  - MQIPT can act as either an SSL/TLS client or an SSL/TLS server depending on which end initiates the connection
  - SSL/TLS handshaking process occurs during the initial connection request between the SSL/TLS client and server
  - MQIPT can SSL/TLS secure sockets directly or can be configured to operate in SSL/TLS Proxy Mode
  - In Proxy Mode, the route only forwards SSL/TLS data between the two MQ end-points; it does not participate in the SSL/TLS handshake and does not require any digital certificates
  - Each MQIPT route can be independently configured with its own set of SSL/TLS properties

MQIPT Features

- Java Security Manager
  - Java Security Manager is available for enhanced security on any MQIPT feature
  - MQIPT uses the default Java Security Manager
  - MQIPT can be enabled or disabled using the global property SecurityManager
  - JSM uses policy files named java.policy
  - A Policy Tool utility is provided for making changes to the policy files
  - Read the product documentation for details on configuring JSM

MQIPT Features

- **Security Exits**
  - Security exits can be used to control access to target destinations defined by the destination route property
  - The security exit is called when MQIPT receives a connection request from a client, before it makes the connection to the destination
  - The security exit decides whether the connection is allowed to complete based on the initial connection properties
  - Each route can have its own security exit
  - Security exits are enabled by setting properties in *mqipt.connf*
    - SecurityExit
    - SecurityExitName
    - SecurityExitPath
    - SecurityExitTimeout
  - Sample security exits are provided with the product

MQIPT Features

- **Port Number Control**
  - The range of port numbers used when making an outgoing connection can be restricted
  - The *OutgoingPort* property can be used to set the initial port for each route in the mqipt.conf file
  - The *MaxConnectionThreads* can be used to set the number of ports to be used

MQIPT Features

- **Additional Security considerations**
  - Coordinate with your firewall administrator to open ports for MQIPT
  - MQIPT verifies that the messages it receives and transmits are valid, and conform to the MQ protocol.
    - This helps prevent MQIPT being used for security attacks outside the IBM MQ protocol.
  - The MaxConnectionThreads property can be used to restrict the total number of incoming conections
    - This helps protect against denial of service attacks
  - The mqipt.conf must be secured by setting appropriate limited permissions for read and write on the file
  - Work with your firewall administrator to limit access to the MQIPT command port if it is enabled
  - Consider disabling the RemoteShutdown property

MQIPT Features

- **Connection Logs**
  - **MQIPT provides a connection log facility to track all successful and unsuccessful connection attempts**
  - **Use the *ConnectionLog* and *MaxLogFileSize* properties in mqipt.conf to configure Connection logs**
  - ***Each entry in the connection log represents one part of a connection request.***
    - ***The connection to MQ ITP from the source is logged as on entry***
    - ***The connection from MQIPT to the destination is another log entry***
    - ***Likewise two entries will be made when each connection is ended***

MQIPT Installation

- **MQIPT is available for download as a category 3 SupportPac (MS81)**
  - IBM MQ SupportPac website (http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24006386&loc=en_US&cs=utf-8&lang=en)
- **The current version of MQIPT is 2.1**
- **The product can be installed in any location the server**
- **Multiple installations is supported on a single server**

MQIPT Installation

- **Installation packages are a available for Linux, Unix and Windows**
  - *AIX* — *ms81_rios_aix_4.tar*
  - *HP-UX* — *ms81_ia64_hpux_11.tar*
  - *Linux x86 (32 bit)* — *ms81_x86_linux_2.tar*
  - *Linux x86 (64 bit)* — *ms81_amd64_linux_2.tar*
  - *Linux zSeries* — *ms81_s390x_linux_2.tar*
  - *Solaris SPARC* — *ms81_sparc_solaris_2.tar*
  - *Solaris x86 (64 bit)* — *ms81_amd64_solaris_2.tar*
  - *Windows (32 bit)* — *ms81_x86_nt_4.zip*
- **To install MQIPT**
  - *Create a directory for installation (e.g D:ibm/mqipt or /opt/ibm/mqipt)*
  - *Unpack the files into that directory*
  - *Update the files permissions so they are read only*

What's new in MQIPT 2.1

- **A Java Runtime Environment (JRE) is included with MQIPT 2.1**

- **MQIPT 2.1 supports several new SSL/TLS features:**
  - ***TLS 1.1 and TLS 1.2 protocol support***
  - ***SHA-2 hash algorithms (SHA-224, SHA-256, SHA-384 and SHA-512 are all supported)***
  - ***Elliptic Curve encryption***
  - ***Support for many new CipherSuites including those that use Galois/Counter Mode (GCM)***

- **MQIPT 2.1 provides the same iKeyman and iKeycmd tools used to administer digital certificates in IBM MQ.**
  - ***These can be run using the new mqiptKeyman and mqiptKeycmd commands***

What's new in MQIPT 2.1

- MQIPT 2.1 supports multiple installations
- Additional certificate DN attributes are supported
- New route properties for SSL/TLS support
- Route specific trace settings
- Error messages have been added and/or amended

MQIPT Upgrade and Migration

- **To migrate from an existing MQIPT 2.0 installation:**
  - *Backup your configuration files*
  - *Stop MQIPT*
    - *mqiptAdmin –stop*
  - *Run the mqipt 2.0 uninstall program*
    - *See the product manual for platform specific details*
  - *Install MQIPT 2.1*
  - *Copy your configuration files into the new installation*

- *MQIPT 2.1 FixPack upgrades*
  - *Backup your configuration files*
  - *Stop MQIPT*
  - *Delete the current installation files*
  - *Install the new version of the product*
  - *Copy your configuration files into the new installation*

Configuration & Administration

- MQIPT 2.1 can be administered from the command line, or by using the MQIPT Administrative Client
  - *It is recommended to use the administrative client for most tasks*
- Starting MQIPT
  - *On Windows systems:*
    - *MQIPT_INSTALLATION_PATH\bin\mqipt MQIPT_HOME_DIR*
  - *On UNIX and Linux systems:*
    - *MQIPT_INSTALLATION_PATH/bin/mqipt MQIPT_HOME_DIR*
- Stopping MQIPT
  - *MQIPT can be stopped using the Administrative client or using the command line*
- MQIPT is usually run as a service that stops and starts with the OS
  - *Detailed instructions are provided in the product documentation for Linux, Unix, and Windows*

Configuration & Administration

- *Using the Administrative Client*
  - *The MQIPT Administrative Client can be started by running the mqiptGui script provided in the bin directory of the product installation*
  - *You must provide connection information on first launch*
    - *MQIPT Name*
    - *Network Address*
    - *Command Port*
    - *Access Password*
  - *The tool can manage any instance of MQIPT, and saves list of mqipt instances in a file named client.conf*
  - *The tool allows you to select an instance to manage*
  - *A new route can be added to any instance*
  - *The administrative client directly edits mqipt.conf for the selected instance*
  - *MQIPT properties can be set at the global level or at the route level*

MQIPT Admin Client Menu Options

N O T E S

- File menu
  - You can manage the list of MQIPT instances by using the following options that are available on the File menu:
- Add MQIPT
  - Adds a new instance of MQIPT to the list in Administration Client. See Starting the Administration Client for details of the information that you must enter.
- Remove MQIPT
  - Removes the currently highlighted instance of MQIPT from the list in Administration Client. This option does not stop or affect the running of this instance of MQIPT.
- Save Configuration
  - Saves the list of MQIPT instances to the local Administration Client configuration file so that they can be restored the next time that Administration Client starts. Only this MQIPT is saved locally; [global] and [route] properties are always retrieved from each instance of MQIPT.

MQIPT Admin Client Menu Options

N O T E S

- Quit
  - Stops Administration Client running. You are given the option to save outstanding changes before Administration Client closes.
- MQIPT menu
  - You can manage the selected instance of MQIPT by using the following options that are available on the MQIPT menu:
- Connection
  - Changes the access properties of an instance MQIPT. See Starting the Administration Client for details of the information that you can update.
- Password
  - Changes the password required to access an instance of MQIPT. Leave the Current Password field blank if there is no password currently set. Do not enter a new password if you want to stop using a password. Select the Save Password check box if you want to save the password locally. If you do not save the password, you must enter it evey time you want to access this instance of MQIPT.

MQIPT Admin Client Menu Options

NOTES

- Add Route
  - Adds a route to a selected instance of MQIPT. Each route must have a unique listener port for an instance of MQIPT.
- Delete Route
  - Deletes the selected route from the instance of MQIPT. The deletion does not take effect until it is applied, by clicking MQIPT > Apply.
- Apply
  - Updates the configuration file of an instance of MQIPT. The new settings are made effective immediately.
- Refresh
  - Reads the current configuration file from the selected instance of MQIPT and refreshes the display.

MQIPT Admin Client Menu Options

NOTES

- Stop
  - Stops an instance of MQIPT from running. After this command, you lose contact with the MQIPT. This command is ignored unless the global property RemoteShutdown is turned on.
- Route properties can be updated in the same way as MQIPT global properties. When you change any properties of a route, you must apply the changes to make them take effect. You can do this either by selecting the MQIPT > Apply menu option or replying Yes when you are prompted to save the configuration.

Configuration & Administration

- *Using the Command Line*
  - *While it is not recommended, it is possible to edit mqitp.conf directly with a text editor. A backup of the files should be made prior to editing the file.*
  - *Use the mqiptAdmin script is in the bin subdirectory*
    - *mqiptAdmin -refresh {hostname {port} } sends the refresh command*
    - *mqiptAdmin -stop {hostname {port} } sends the stop command*

Configuration & Administration

- *Tuning*
  - *The performance of each route can be tuned using a combination of a thread pool and an idle timeout specification*
  - *Connection thread pool*
    - *Each route is assigned a working pool of concurrently running threads that handle incoming communication requests*
      - *MinConnectionThreads*
      - *MaxConnectionThreads*
  - *Idle timeout*
    - *Seting the IdleTimeout property will ensure that threads that have been inactive for the specified period of time are recycled.*
    - *Recycled threads are placed back into the working thread pool.*

Questions & Answers