MQSeries - Put Message Utility
Version 1.0.2

6 Feb 2002

Annette Green
IBM Global Services
6710 Rockledge Drive
Bethesda, Maryland 20817

greenac@us.ibm.com

Take Note!

Before using this report be sure to read the general information under "Notices".

Third Edition, February 2002

This edition applies to Version 1.0.2 of *MQSeries – Put Message Utility* and to all subsequent releases and modifications unless otherwise indicated in new editions.

Table of Contents

MQSeries - Put Message Utility i

Notices iv

Trademarks and service marks iv

Summary of Amendments v

Preface vi

Bibliography vii

Chapter 1. Overview 1

Chapter 2. Installation 2

Prerequisites 2

Software 2

Hardware 2

Setting Up and Running the Application 2

Chapter 3. Using the Put Message Utility Program 5

Chapter 4. Graphical and Functional Description 6

PutMessage Main Screen 6

MQ Environment List Screen 10

PutMessage Help Screen 11

MQMD Header Fields Screen 12

MQ Constant List Screen 16

PutMessage Advance Options Screen 17

PutMessage Standard Output 20

Notices

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

Information contained in this report has not been submitted to any formal IBM test and is distributed “AS-IS”. The use of this information and the implementation of any of the techniques is the responsibility of the reader. Much depends on the ability of the reader to evaluate these data and project the results to their operational environment.

The performance data contained in this report was measured in a controlled environment and results obtained in other environments may vary significantly.

Trademarks and service marks

The following terms, used in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

- IBM
- MQSeries
- AIX
- OS/2 Warp

The following terms are trademarks of other companies:

- Windows NT Microsoft Corporation
- Java Sun Microsystems, Inc.

Summary of Amendments

Date	Changes
13 October, 2000	Initial release
11 May, 2001	Updated documentation to include additional installation instructions for later versions of Java
6 February 2002	Updated installation instructions for MQSeries classes for Java Version 5.2

Preface

The MQSeries - Put Message Utility is a Java application geared towards testers or developers using MQSeries v5.1 or v5.2. The application uses the MQPUT operation to place messages into a MQSeries queue specified by the user. The message body can be created from a file specified by the user or from text manually entered by the user. In addition, the application allows the user to directly update the MQMD message header fields as well as the context information of a message prior to each MQPUT operation. This application can run as a MQSeries client application or locally connected to the MQSeries server.

Bibliography

- *MQSeries Using Java,* IBM Corporation. SC34-5456.
- *MQSeries Application Programming Reference*, IBM Corporation. SC33-1673
- *MQSeries Application Programming Reference Summary*, IBM Corporation. SX33-6095
- *MQSeries Application Programming Guide*, IBM Corporation. SC33-0807

Chapter 1. Overview

The MQSeries - Put Message Utility application uses the MQPUT operation to place messages into a MQSeries queue specified by the user. The message body can be created from a file specified by the user or from text manually entered by the user. In addition, the application allows the user to update the MQMD message header fields using MQSeries constant variables or corresponding literal values. The application also allows the user to control the context information of a message when the message is placed into a queue.

The MQSeries - Put Message Utility application defaults to the following put-message options (MQPMO) when implementing the MQPUT operation:

- MQPMO_SET_IDENTITY_CONTEXT,
- MQPMO_FAIL_IF_QUIESCING, and
- MQPMO_NO_SYNCPOINT.

However, the application also gives the user the option of setting the message context to one of the following values:

- MQPMO_NO_CONTEXT,
- MQPMO_DEFAULT_CONTEXT,
- MQPMO_SET_ALL_CONTEXT, as well as
- MQPMO_SET_IDENTITY_CONTEXT.

When the message context is changed then the user's access to the MQMD identity and origin context fields are modified according to the user selection.

The user can additionally update the put-message options (MQPMO) by enabling segmentation of a message. If the user opts to enable segmentation of a message that is larger than the *MaxMsgLength* for the user-specified Queue or Queue Manager, then

- The MQPMO_NO_SYNCPOINT option is automatically replaced by the MQPMO_SYNCPOINT option and the put operation is executed within a unit of work.
- The MQPMO_LOGICAL_ORDER option is automatically added to the put-message options.

Using the application's graphical user interface (GUI), the user can connect or disconnect from any queue manager accessible by the application and put various messages to any corresponding queue using different MQPUT options. This application can run as a MQSeries client application or locally connected to the MQSeries server. The status of any MQSeries operations executed by the application is redirected to standard output.

Chapter 2. Installation

Prerequisites

This section lists the required hardware and software components needed to run the MQSeries - Put Message Utility application. The application also requires access to at least one MQSeries server.

Software

The following third party software components must be installed in order to properly run these applications:

- IBM MQ base Java Code, V5.1, or

  SupportPac MA88 (MQSeries classes for Java and MQSeries classes for Java Message Service)

- Java Runtime Environment 1.1 or greater

Hardware

The MQSeries - Put Message Utility application *should* be able to run on the following hardware platforms that support the MQSeries Version 5.1 and V5.2 product:

- Windows NT
- Sun Solaris
- AIX
- HP-UX
- OS/2 Warp

Setting Up and Running the Application

The following section provides instructions on installing and running the MQSeries - Put Message Utility application:

1. Install the software components mentioned in the “Prerequisites” section. Refer to the documentation or instructions associated with these components for further details on the installation. Take note of the directory where the MQ base Java code is installed and the directory where the Java application launcher is located. For version 1.1 of Java, the application launcher is called “jre”, however, for later versions of Java the launcher is called “java”.

2. The `PutMessage.jar` file is the program code provided by this SupportPac. Copy the `PutMessage.jar` file into a directory on your machine (i.e. c:\mqprog\)

3. To run the application, change the current directory to the location of the Java application launcher and start the Java application `mqjava.programs.PutMessage` by explicitly specifying its classpath using the –cp option. Enter the following commands into a PutMessage batch file or from the operating system command prompt:

   ```
   cd <Java_Directory>
   ```

```
<jre> -cp <MQ_Directory>com.ibm.mq.jar;<PutMsg_Directory>PutMes
sage.jar  mqjava.programs.PutMessage

For MQ Java Version 5.2, use:

<jre> -cp <MQ_Directory>com.ibm.mq.jar;<MQ_Directory>jta.jar;<P
utMsg_Directory>PutMessage.jar  mqjava.programs.PutMessage
```

where

<Java_Directory> = the directory where the Java interpreter/application launcher is located.

<jre> = the launcher for Java applications. For Java 1.1, it is the **`jre`** tool for deployment environments and the **`java`** tool for development environment. For later versions of Java, it is the **`java`** tool.

<MQ_Directory> = the directory where the MQ Java code was installed (i.e. c:\mqm\java\lib\). The `com.ibm.mq.jar` file should be located in this directory. For MQ Java v5.2, both the MQ base Java and the MQ JMS code should have been installed. The `jta.jar` file is included with the MQ JMS code.

<PutMsg_Directory> = the directory where `PutMessage.jar` is located (i.e. c:\mqprog\). See step 2 above.

From whichever directory the MQSeries - Put Message Utility application is running, the user should make sure that the application has write-access to that directory. Otherwise, the MQSeries - Put Message Utility application will not be able to conveniently save the user's MQ environment settings. When a user successfully connects to a queue manager, these settings are saved to text files in the same directory where the application is executing. The following text files may be created by the application in order to store the user's MQ environment settings:

- MA0JQMgr.txt
- MA0JQueu.txt
- MA0JHost.txt (Note: Created only if connected as a client)
- MA0JChan.txt (Note: Created only if connected as a client)
- MA0JPort.txt

If the user changes the location where the application is running, then the user will probably want to copy the above files (if any were created) to the new directory location.

NOTE:

Attempting to put large messages into a queue may result in a ***java.lang.OutOfMemoryError***. In order to avoid this problem, the user will need to increase the virtual memory of the Java runtime environment (the default value is 16M). To increase memory, the user should use the **–mx** option of the Java application launcher tool, which specifies the maximum size of the Java virtual memory. For versions 1.2 and later of Java, this option has become nonstandard and it is recommended that **–Xmx** is used instead.

For example,

```
jre -mx32m -cp <…> mqjava.programs.PutMessage

or

java -Xmx32m -cp <…> mqjava.programs.PutMessage
```

The above command doubles the default memory from 16 megabytes to 32 megabytes. The Java tool reference documentation can provide more information on this option.

Chapter 3. Using the Put Message Utility Program

After starting up the MQSeries – Put Message Utility application, the user only needs to perform a few basic steps in order to effectively use the application. These steps are:

1. Enter values for the following MQ environment fields:

   - Queue Manager
   - Queue
   - Hostname (optional)
   - Channel (optional)
   - Port

   If the Hostname and Channel are not specified then a local connection will be made to the designated Queue Manager. Otherwise, a client connection is made to the Queue Manager.

2. Connect to the Queue Manager by pressing the “Connect” button.

3. Specify the message body either from a file by selecting the “Get Message from File” option or from typed input by selecting the “Insert Message Text” option.

4. Optionally, specify values for the message descriptor fields by pressing the “View/Edit Header Options” button. Defaults values for the message descriptor will be used if the user decides not to make any changes.

5. Put the message to the designated Queue by pressing the “Put Message to Queue” button.

6. Repeat steps 3-5 for as many messages that the user wants to place on the specified queue.

7. Disconnect from the Queue Manager by pressing the “Disconnect” button.

Chapter 4. Graphical and Functional Description

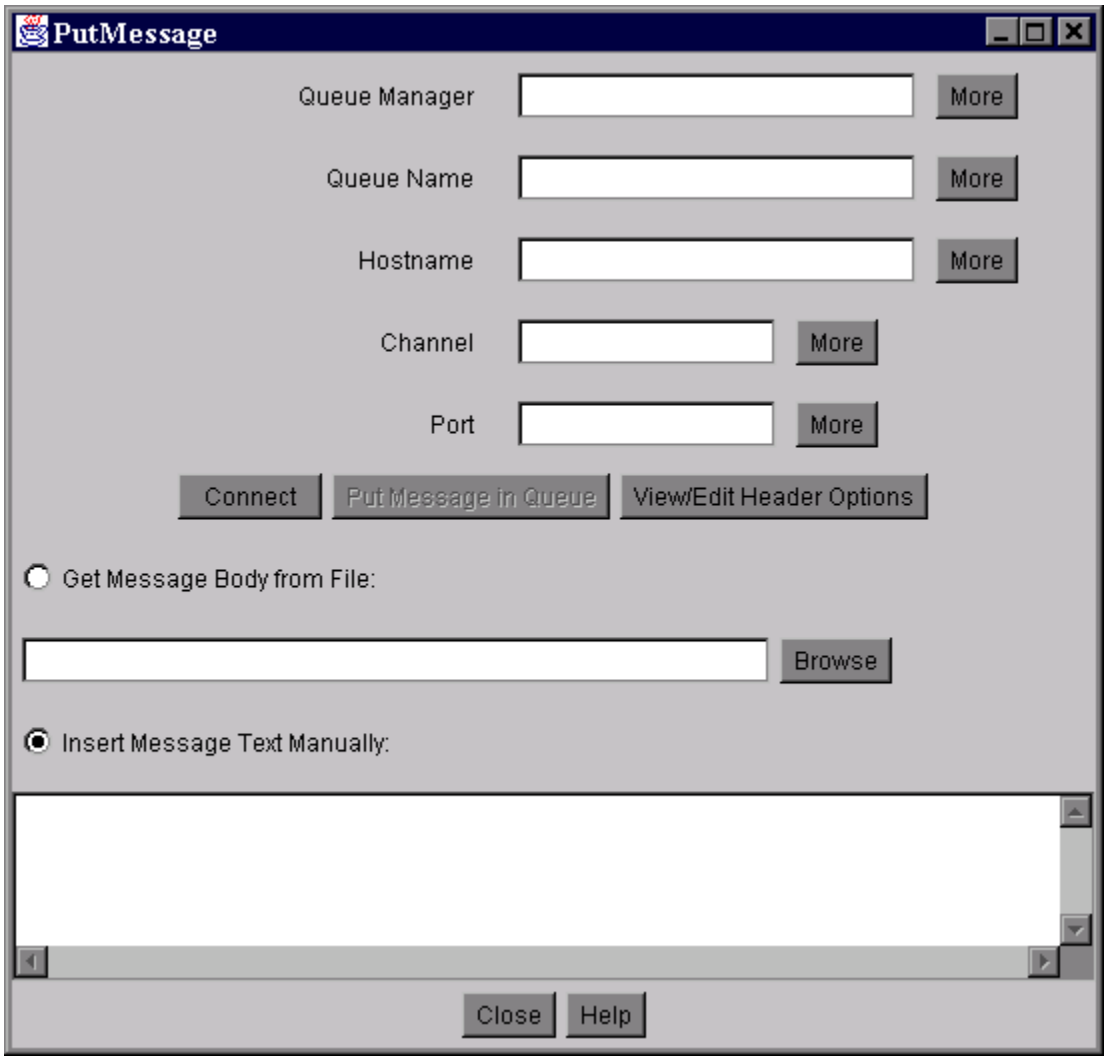
The MQSeries - Put Message Utility application consists of a graphical user interface (GUI) which allows the user to connect to an MQSeries server and put messages to MQSeries queues. The section describes screens and GUI components used by the MQSeries - Put Message Utility application as well as how the application interacts with the user input.

Please note that the screen images may vary slightly depending upon the platform where the application is running.

PutMessage Main Screen

When the user initiates the MQSeries - Put Message Utility application, the following screen will be displayed.

Figure 1 - PutMessage Screen



The table below describes the GUI components that reside on the main PutMessage screen.

Table 1 - Main Screen GUI Description

Name	Type	Description
Queue Manager	Text Field	Enter the name of the Queue Manager to which to connect. This field is required by the application in order to make a connection to the MQSeries server.
Queue Name	Text Field	Enter the name of the queue within the specified Queue Manger to open. This field is required by the application in order to make a connection to the MQSeries server.
Hostname	Text Field	Enter the TCP/IP hostname of the machine on which the MQSeries server resides. If the Hostname is not specified, then a direct connection will be made to the MQSeries server via Bindings Mode. Otherwise a Client connection is made via the network.
Channel	Text Field	Enter the name of the channel to connect to on the target Queue Manager. If the Channel is not set, then the Hostname cannot be set.
Port	Text Field	Enter the port to connect to. This value should be the port on which the MQSeries server is listening for incoming connection requests. This field is required by the application in order to make a connection to the MQSeries server.
More	Button	When pressed, displays the saved list of values for the corresponding field. Values are automatically added to the list when a successful connection to the specified Queue Manager and Queue has been made. Choosing a value from the list can change the value of the corresponding text field. The "More" button is disabled when the list is empty and also when a successful connection to the Queue Manager has been made.
Connect	Button	When pressed, the application will attempt to connect to the specified Queue Manager and open the specified Queue. If a problem occurs, then a corresponding error message will appear from standard output. If the connection is successful then the "Connect" button will change to a "Disconnect" button and the "Put Message in Queue" button will be enabled. In addition, the Queue Manager, Queue Name, Hostname, Channel, and Port fields will be disabled.
Disconnect	Button	This button is enabled when a successful connection has been made to the specified Queue Manager and Queue. When pressed, the application will close the specified Queue and disconnect from the specified Queue Manager. The "Disconnect" button will then change to a "Connect" button. In addition, the Queue Manager, Queue Name, Hostname, Channel, and Port fields will be enabled and the "Put Message in Queue" button will be disabled.

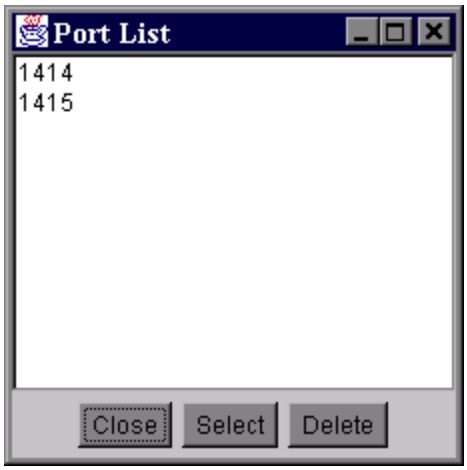
Name	Type	Description
Put Message in Queue	Button	This button is enabled when a successful connection has been made to the specified Queue Manager and Queue. When pressed, the application will create a MQSeries message and put the message to the specified Queue. The message body is either extracted from a file or entered by the user based on whether the "Get Message Body from File" radio button or the "Insert Message Text Manually" radio button is selected, respectively. The application uses the current values from the *MQMD Header Fields* screen and the *PutMessage Advance Options* screen as input to the message descriptor and message context. In addition, the "Disconnect" button will become disabled until the Put operation is complete. To determine the status of a Put operation, a success or failure message is displayed on the title bar of the main screen and additionally more descriptive status messages are written out to standard output for the user to view. **Note:** Multiple Put operations can occur simultaneously (based on the amount of time the application takes to put messages of various sizes to the specified Queue and based on the number of times the user presses the “Put Message in Queue” button during that time interval). In these cases, the user can determine the status of each Put operation based on the pre-fixed index number attached to the status messages.
View/Edit Header Options	Button	When pressed, the application will display the *MQMD Header Fields* screen, which lists configurable fields from the MQMD data structure. The user can update these fields using the MQSeries constant variable name or their corresponding literal values.
Get Message Body from File	Radio Button and Text Field	When selected and the "Put Message in Queue" button is pressed, then the application retrieves the contents of the file specified in the text field as the message body. The user can either manually type the filename into the text field or use the “Browse” button to populate the field. If an absolute path is not specified with the filename, then the application looks for the file in the same directory where the application is running. If no file is specified or the file cannot be accessed by the application, then a blank message is created instead. Text as well as binary files may be specified within the text field. Depending upon the file type, the user may want to modify the MQMD Format field in order to later allow proper retrieval of the message from the queue.
Browse	Button	When pressed, a file dialog box will appear which will allow the user to search for and select a file from the local machine as input for the MQSeries message body. The selected filename will appear on the corresponding text field once the user closes the dialog box.
Insert Message Text Manually	Radio Button and Text Area	When selected and the "Put Message in Queue" button is pressed, then the application uses the text entered into the writing area as input for the MQSeries message body. The writing area is cleared once the message is successfully put to the queue.

Name	Type	Description
Help	Button	When pressed, the application will display the *PutMessage Help* screen, which contains a description of all the GUI components as well as the version of the MQSeries - Put Message Utility application being used.
Close	Button	When pressed, the application will disconnect from the specified Queue Manager (if connected) and then exit the application.

MQ Environment List Screen

When the user presses one of the “More” buttons from the main PutMessage screen, then a screen similar to the one below will be displayed. A screen exists for each of the MQ environment fields: Queue Manager, Queue Name, Hostname, Channel, and Port.

Figure 2 – MQ Environment List Screen



The corresponding table describes the GUI components that reside on this screen.

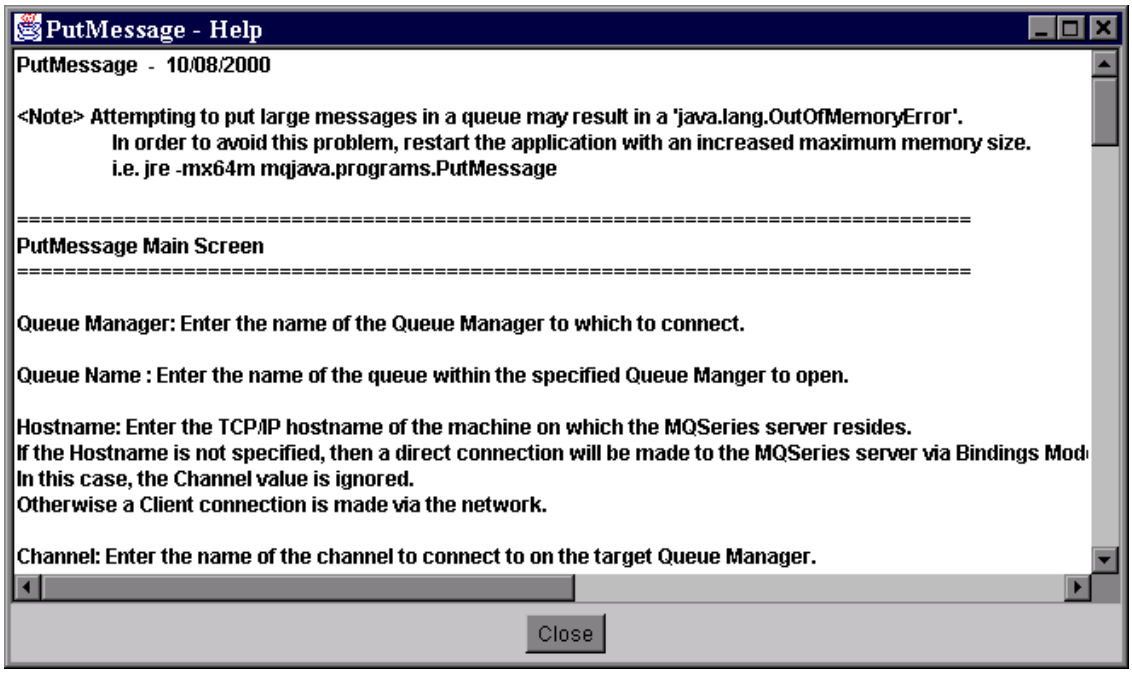
Table 2 - MQ Environment List Screen Description

Name	Type	Description
Saved Value List	List	Displays the saved list of values for the following MQSeries environment fields: Queue Manager, Queue Name, Hostname, Channel, or Port. The values for the list are obtained from user input into the corresponding text fields from the main *PutMessage* screen. Values are automatically added to the list when a successful connection to the Queue Manager and Queue has been made. A separate list is created for each of the MQ Environment fields.
Close	Button	When pressed, the application will close the screen.
Select	Button	When pressed and a value within the list has been highlighted, then the value is copied to the corresponding MQ Environment text field on the main *PutMessage* screen and the screen closes. The user can also select a value by double-clicking on the value, or by pressing the Enter key when a value within the list has been highlighted.
Delete	Button	When pressed and a value within the list has been highlighted, the value is deleted from the list. If the deleted value was also listed on the corresponding text field from the main *PutMessage* screen, then the value is removed from the text field as well. If all values within the list are deleted, then the “More” button for accessing the screen will be disabled.

PutMessage Help Screen

When the user presses the “Help” button from the main PutMessage screen, then the following screen is displayed.

Figure 3 - PutMessage Help Screen



The corresponding table describes the GUI components that reside on this screen.

Table 3 - PutMessage Help Screen Description

Name	Type	Description
Help Description	Text Area	Displays a description of the PutMessage GUI components. In addition, displays the version of the PutMessage application being used. The user can move the horizontal and vertical scroll bars to view the entire description of all components.
Close	Button	When pressed, the application will close the *PutMessage Help* screen.

MQMD Header Fields Screen

When the user presses the “View/Edit Header Options” button from the main PutMessage screen, then the following screen is displayed. The field values on this screen will be used to populate the MQSeries message descriptor whenever a message is put to a specified queue. The user can enter a MQSeries-defined constant within a text field or a literal string or integer value depending upon the expected data type. For numeric data types, the user can enter multiple MQSeries constants into one field using the pipe ("|") character. Otherwise the user can enter only one literal value into a text field.

The header field values are not validated until the user presses the “Put Message in Queue” button from the main screen. The application performs some field validation and correction in order to successfully put a message to the queue. In those cases where the application detects invalid field values entered by the user, the application will replace the invalid values with default values and then write an informational message to standard output. In all other cases, the application allows MQSeries to process the header values. If MQSeries detects a problem with the header values then the put operation will fail and an error message is written to standard output.

To give an example of the processing performed by the application, the following situation is detected as an error by the MQSeries - Put Message Utility application:

```
Message Type:     MQMT_REQUESTMSG
```

The application detects this Message Type value as an error because the value is not a known MQSeries constant. In this case, the default value MQMT_REQUEST will be used by the application when a put operation is initiated. The next situation is an example of an error detected by MQSeries but not by the application:

```
Message Type:     MQMT_REQUEST | MQMT_DATAGRAM
```

The application does not detect a problem because these Message Type values are valid MQSeries constants. However, MQSeries will reject these values because they cannot be used together and therefore the put operation will fail.

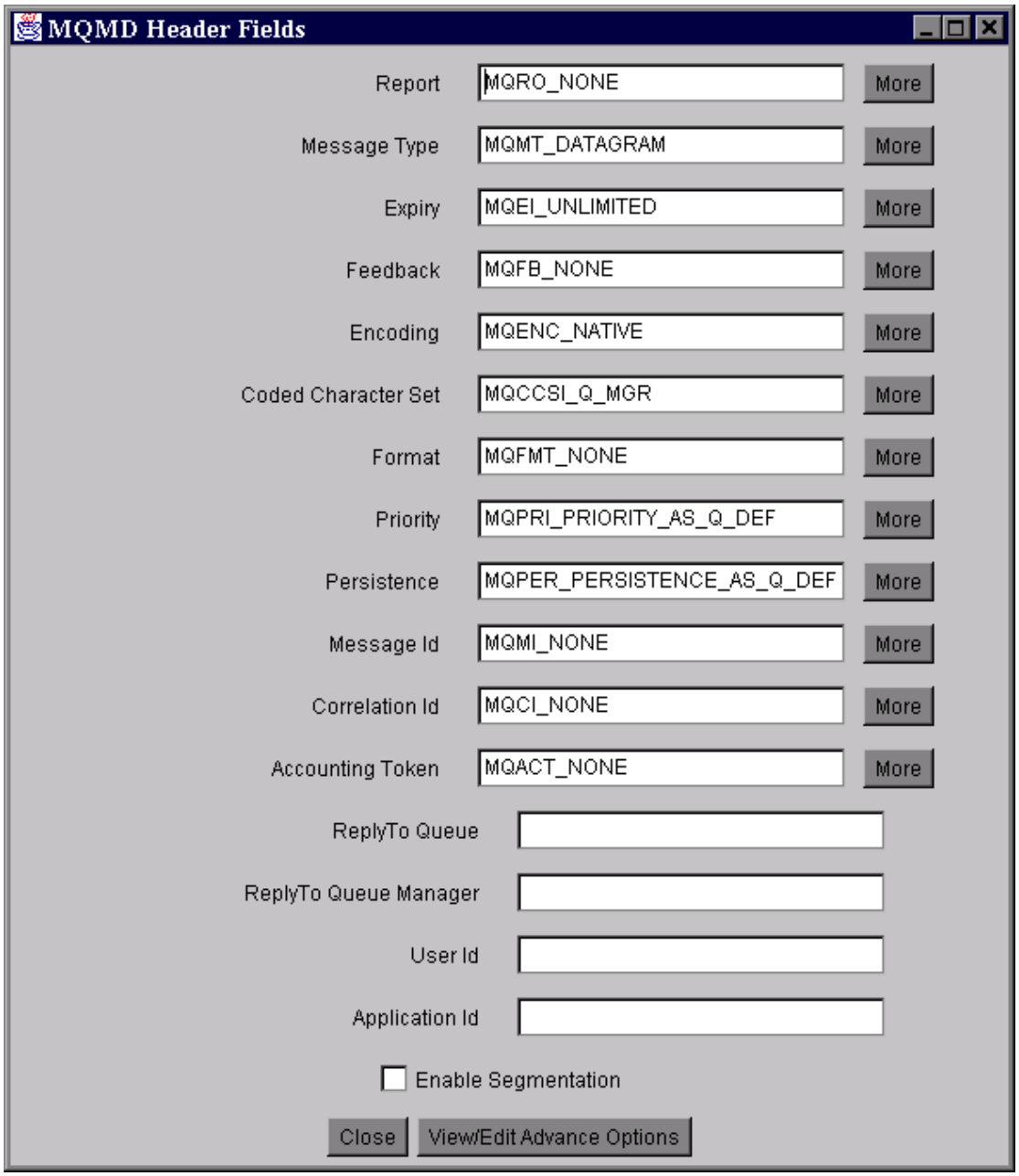
The *MQMD Header Fields* screen displays most but not all of the fields within the MQSeries MQMD data structure. The following message descriptor header fields are not displayed and therefore cannot be modified by the user:

- Structure Identifier
- Version
- Backout Count
- Group Identifier
- Message Sequence Number
- Offset
- Message Flag
- Original Length

The application uses the default value of MQMD_STRUC_ID for the Structure Identifier field and MQMD_VERSION_2 for the Version field. The Backout Count field was omitted because it is ignored on a MQPUT operation. The Group Identifier, Message Sequence Number, Offset, Message Flag, and Original Length fields are not displayed because the application

does not allow the user to manually put message groups onto a specified queue but alternatively allows the user to specify automatic segmentation of a message using the queue manager. Note that the origin context fields are displayed on the *PutMessage Advance Options* screen.

Figure 4 - MQMD Header Fields Screen



The table below describes the header field components that reside on this screen and lists the default values that are used by the application for each of the header fields. Refer to the *MQSeries Application Programming Reference Summary* to determine the allowable values for each field and to get a more detailed description of each field.

Table 4 - MQMD Header Screen Field Description

Name	Type	Default Constant	Constant Value	Data Type
Report	Text Field	MQRO_NONE	0	MQLONG
Message Type	Text Field	MQMT_DATAGRAM	8	MQLONG
Expiry	Text Field	MQEI_UNLIMITED	-1	MQLONG
Feedback	Text Field	MQFB_NONE	0	MQLONG
Encoding	Text Field	MQENC_NATIVE	Varies by platform	MQLONG
Coded Char Set Id	Text Field	MQCCSI_Q_MGR	0	MQLONG
Format	Text Field	MQFMT_NONE	Blanks	MQCHAR8
Priority	Text Field	MQPRI_PRIORITY_AS_Q_DEF	-1	MQLONG
Persistence	Text Field	MQPER_PERSISTENCE_AS_Q_DEF	2	MQLONG
Message Id	Text Field	MQMI_NONE	Nulls	MQBYTE24
Correlation Id	Text Field	MQCI_NONE	Nulls	MQBYTE24
Accounting Token	Text Field	MQACT_NONE	Nulls	MQBYTE32
ReplyTo Queue	Text Field	None	Blanks	MQCHAR48
ReplyTo Queue Manager	Text Field	None	Blanks	MQCHAR48
User Identifer	Text Field	None	Blanks	MQCHAR12
Application Id	Text Field	None	Blanks	MQCHAR32

The following table describes the remaining GUI components that reside on the *MQMD Header Fields* screen.

Table 5 - MQMD Header Screen GUI Description

Name	Type	Description
More	Button	When pressed, displays the list of MQ constants for the corresponding field. Choosing constants from the list can change the value of the corresponding text field.
Enable Segmentation	Checkbox	When checked and the "Put Message in Queue" button is pressed, the application will allow MQSeries to automatically segment messages that are larger than the maximum message length for the specified Queue or Queue Manager. If this field is unchecked and the application attempts to put a message into the specified Queue that is greater than the maximum message length, the application will return an error message to standard output. The default value is to disable segmentation.
View/Edit Advance Options	Button	When pressed, displays the *PutMessage Advance Options* screen which provides additional flexibility on controlling the context information of a message.
Close	Button	When pressed, the application will close the *MQMD Header Fields* screen.

MQ Constant List Screen

When the user presses one of the “More” buttons from the *MQMD Header Fields* screen, then a screen similar to the one below will be displayed. A screen exists for each of the following message descriptor fields: Report, Message Type, Expiry, Feedback, Encoding, Coded Char Set Id, Format, Priority, Persistence, Message Id, Correlation Id, and Accounting Token.

Figure 5 – MQ Constant List Screen

The corresponding table describes the GUI components that reside on this screen.

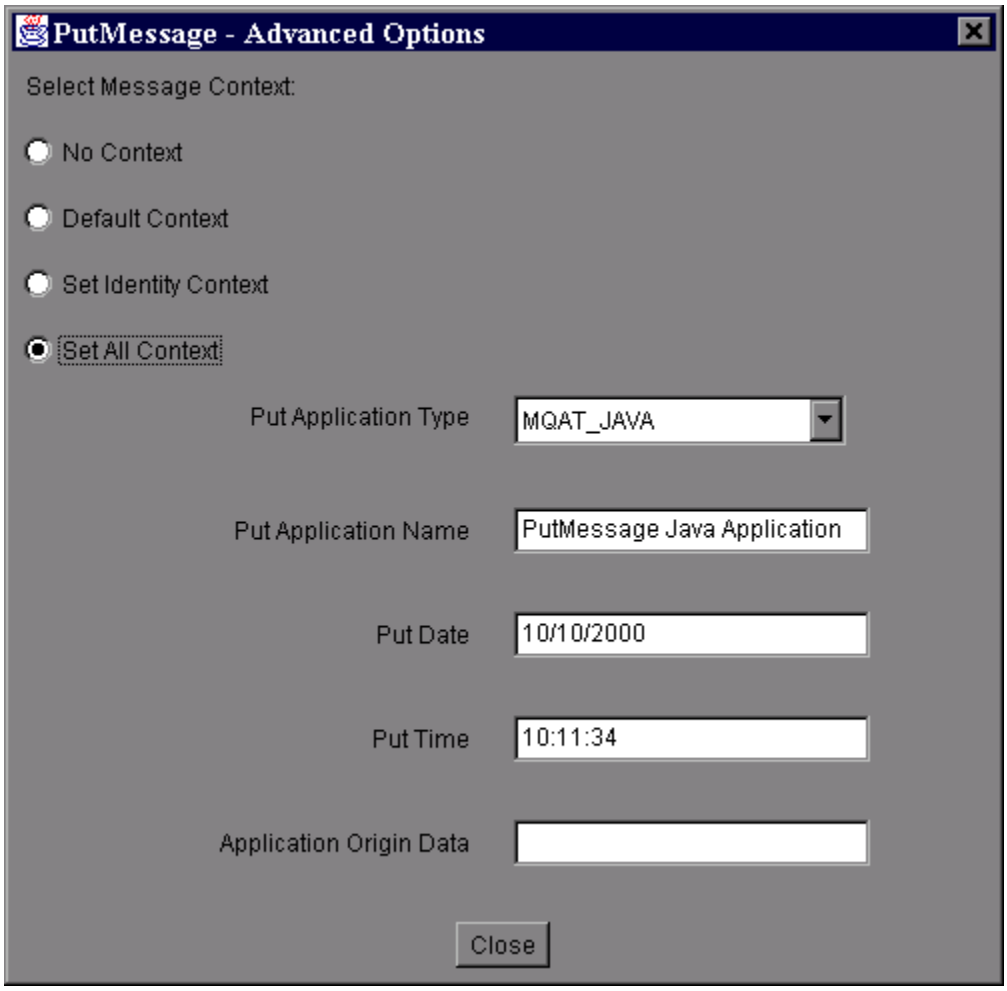
Table 6 - MQ Constant List Screen Description

Name	Type	Description
Constants List	List (Multi-select)	Displays the list of valid MQ constants for the following MQMD header fields: Report, Message Type, Expiry, Feedback, Encoding, Coded Char Set Id, Format, Priority, Persistence, Message Id, Correlation Id, and Accounting Token. A separate list exists for each of these MQ message descriptor fields. Please note that the MQ constants in the list are the only ones recognized by the application. The user may alternatively enter the literal constant value into the corresponding text field.
Close	Button	When pressed, the application will close the screen.
Select	Button	When pressed and at least one MQ constant within the list has been highlighted, then the value(s) are copied to the corresponding MQMD text field on the *MQMD Header Fields* screen and the screen closes. The user can also select a value by double-clicking on the value, or by pressing the Enter key when a value within the list has been highlighted. If the user selects multiple values from the list, then each value will appear on the corresponding text field separated by the pipe character (“|”).

PutMessage Advance Options Screen

The following screen is displayed when the user presses the “View/Edit Advance Options” button from the *MQMD Header Fields* screen. This screen allows the user to control the context information when a message is put to a queue. In addition, when the user opts to set all context fields, the user can specify values for the MQMD origin context fields. Unlike all the other PutMessage screens, the *PutMessage Advance Options* screen is modal and will not allow the user to access any other PutMessage screens until it is closed. For more information on controlling the message context, please refer to the *MQSeries Programming Guide*.

Figure 6 - PutMessage Advance Options Screen



The following table describes the remaining GUI components that reside on the *PutMessage Advance Options* screen.

Table 7 - PutMessage Advance Options Description

Name	Type	Description
No Context	Radio Button	When selected and the user presses the “Put Message in Queue” button, then the MQPMO_NO_CONTEXT option is specified in the put-message options. In addition, when selected, the Identity Context fields (User Identifier, Accounting Token, Application Id) and the Origin Context fields (Put Application Type, Put Application Name, Put Date, Put Time, Application Origin Data) are disabled.
Default Context	Radio Button	When selected and the user presses the “Put Message in Queue” button, then the MQPMO_DEFAULT_CONTEXT option is specified in the put-message options. In addition, when selected, the Identity Context fields (User Identifier, Accounting Token, Application Id) and the Origin Context fields (Put Application Type, Put Application Name, Put Date, Put Time, Application Origin Data) are disabled.
Set Identity Context	Radio Button	When selected and the user presses the “Put Message in Queue” button, then the MQPMO_SET_IDENTITY_CONTEXT option is specified in the put-message options. In addition, when selected, the Identity Context fields (User Identifier, Accounting Token, Application Id) are enabled and the Origin Context fields (Put Application Type, Put Application Name, Put Date, Put Time, Application Origin Data) are disabled. The user can update the Identity Context fields and the values will be used in the message descriptor when a message is put to a queue. This option is the default setting for the message context.
Set All Context	Radio Button	When selected and the user presses the “Put Message in Queue” button, then the MQPMO_SET_ALL_CONTEXT option is specified in the put-message options. In addition, when selected both the Identity Context fields (User Identifier, Accounting Token, Application Id) and the Origin Context fields (Put Application Type, Put Application Name, Put Date, Put Time, Application Origin Data) are enabled. The user can update the Identity and Origin Context fields and the values will be used in the message descriptor when a message is put to a queue.
Close	Button	When pressed, the application will close the *PutMessage Advance Options* screen.

The table below describes the header field components that reside on this screen and lists the default values that are used by the application for each of the header fields when the message context is set to “Set All Context”. In all other cases, the Queue Manager determines the values used in these fields. Please note that the default values used by the application are *not* necessarily the same as the default values used by MQSeries. Refer to the *MQSeries Application Programming Reference Summary* to determine the allowable values for each field.

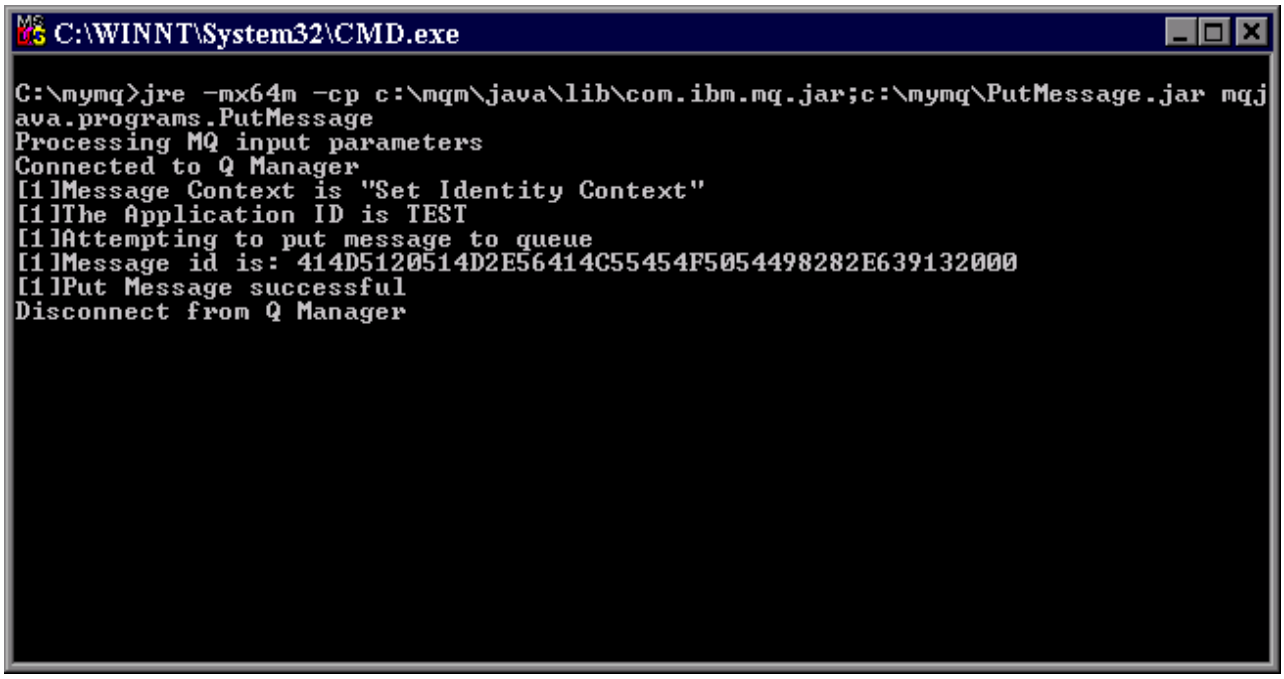
Table 8 - MQMD Header Screen Field Description

Name	Type	Default Constant	Constant Value	Data Type
Put Application Type	Dropdown List	MQAT_JAVA	28	MQLONG
Put Application Name	Text Field	PutMessage Java Application	N/A	MQCHAR28
Put Date	Text Field	<Current Date in MM/DD/YYYY format>	N/A	MQCHAR8
Put Time	Text Field	<Current Time in HH:MM:SS format>	N/A	MQCHAR8
Application Origin Data	Text Field	<Blank>	N/A	MQCHAR4

PutMessage Standard Output

When the user interacts with the MQSeries - Put Message Utility application to connect to a queue manager, open a queue, and put messages to the queue, then the status of these operations are displayed at standard output. An example of the status messages displayed to the user is provided below.

Figure 7 - Standard Output Example



End of Document