

# FioranoMQ® 9

## C++ (Native) RTL Sample Application Guide

Copyright (c) 1999-2008, Fiorano Software Technologies Pvt. Ltd.,

Copyright (c) 2008-2009, Fiorano Software Pty. Ltd.

All rights reserved.

This software is the confidential and proprietary information of Fiorano Software ("Confidential Information"). You shall not disclose such ("Confidential Information") and shall use it only in accordance with the terms of the license agreement enclosed with this product or entered into with Fiorano.

# Contents

---

<b>Native C++ Runtime Examples.....</b>	<b>1</b>
PTP Samples.....	2
Admin .....	2
Basic.....	2
Browser.....	3
HTTP.....	3
HTTPS.....	4
MsgSel .....	4
Mtptp .....	5
reqrep.....	5
basic .....	5
TimedOut .....	6
SSL.....	6
Transaction.....	7
PubSub Samples .....	7
Admin .....	7
Basic.....	7
Dursub .....	8
HTTP.....	9
HTTPS.....	9
Msgsel.....	10
Mtpubsub .....	10
Reqrep .....	11
Basic .....	11
TimedOut .....	11
SSL.....	12
Transaction.....	12

---

# Native C++ Runtime Examples

---

This chapter describes usage of C++ Runtime for connecting to FioranoMQ server. The various sample programs illustrates the use of simple Publish-Subscribe and Point-To-Point operations.

## PTP

This directory contains samples which demonstrate the following functionality over PTP using C++ Runtime.

- Admin
- Basic Send Receive
- Browser
- HTTP
- HTTPS
- Message Selector
- Multi-thread PTP
- RequestReply
  - Basic
  - TimedRequestReply
- SSL
- Transaction

## PubSub

This directory contains samples which demonstrate the following functionality over PubSub using C++ Runtime.

- Admin
- Basic Pub Sub
- Durable Subscriber
- HTTP
- HTTPS
- Message Selector

- Multi-thread PubSub
- RequestReply
  - Basic
  - TimedRequestReply
- SSL
- Transaction

These samples are available in %FMQ\_DIR%\clients\cpp\native\samples directory. The %FMQ\_DIR%\clients\cpp\native\script directory contains a script called build\_samples.bat which compiles the C++ programs.

## PTP Samples

### Admin

This directory contains one sample program which illustrates basic JMS Administration API functionality using the FioranoMQ C++ Runtime Library.

- **AdminTest.cpp** - Creates an Admin Connection with the MQServer and gets an MQAdminService object to create and delete Queues and QueueConnectionFactory and retrieves information of users connected from the server.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cpplclientbuild.bat which compiles the C++ program.
2. Run the AdminTest by executing the *AdminTest.exe* executable file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRAM~1\Fiorano\FIORAN~2\fmq)

### Basic

This directory contains two sample programs which illustrate JMS Send-Receive mechanism using the FMQ C++ Runtime Library.

- **Sender.cpp** - Reads strings from standard input and sends the text messages on the queue "primaryQueue".
- **Receiver.cpp** - Implements a synchronous blocking receiver, which listens on the queue "primaryQueue", and prints the text of the received text messages on the console.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called `cppclientbuild.bat` which compiles the C++ program.
2. Run the Sender by executing the *Sender.exe* executable file.
3. Run the receiver by executing the *Receiver.exe* file.

**Note:** To run any of the C++ samples, ensure environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## Browser

This directory contains two sample programs which illustrate basic JMS Browser functionality using the FioranoMQ C++ Runtime Library.

- **QSender.cpp** - Reads strings from standard input and sends them on the queue "primaryqueue".
- **Browser.cpp** - Implements a browser, which is used to browse the messages on the queue "primaryqueue", and prints out the received messages.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. For convenience, compiled versions of the sources are included in this directory. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called `cppclientbuild.bat` which compiles the C++ program.
2. Run the Sender by executing the *QSender.exe* executable file. Send some messages before running the browser application
3. Run the browser by executing the *Browser.exe* file.

**Note:** To run any of the C++ samples, ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (This defaults to C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## HTTP

This directory contains two sample programs which illustrate the use of HTTP protocol for basic JMS PTP functionality using the FioranoMQ C++ Runtime Library.

- **QReceiver.cpp** - Receives messages asynchronously on primaryQueue. This program implements a synchronous listener to listen for messages published on the queue "primaryQueue".
- **QSender.cpp** - Implements a client application publishing user specified data on primaryQueue. This program reads strings from standard input and publishes them on the Queue "primaryQueue".

To run this sample using FioranoMQ, do the following:

1. Compile the source file. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called `cppclientbuild.bat` which compiles the C++ program.
2. Run the HttpReceiver by executing the *QReceiver.exe* executable file.

3. Run the HttpSender by executing the *QSender.exe* executable file.

**Note:** To run any of the C++ samples, ensure that environment variable FMQ\_DIR points to Fiorano installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq).

## HTTPS

This directory contains four sample programs which illustrate the use of HTTPS protocol for basic JMS PTP functionality using the FioranoMQ C++ Runtime Library.

- **HReceiver.cpp** - Receives messages synchronously on primaryQueue. This program implements a synchronous listener to listen for messages published on the queue "primaryQueue".
- **HSender.cpp** - Implements a client application publishing user specified data on primaryQueue. This program reads strings from standard input and publishes them on the Queue "primaryQueue".

To run this sample using FioranoMQ, do the following:

1. Compile the source file. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cplusplusclientbuild.bat which compiles the C++ program.
2. Run the HttpsReceiver by executing the *HReceiver.exe* executable file.
3. Run the HttpsSender by executing the *HSender.exe* executable file.

**Note:** To run any of the C++ samples, ensure that environment variable FMQ\_DIR points to Fiorano installation directory (this defaults to C:\PROGRA~1\Fiorano\FIORAN~2\fmq).

## MsgSel

This directory contains two sample programs which illustrate the use of message selectors using the FioranoMQ C++ Runtime Library.

- **QSelSender.cpp** - Selector sends messages with the string property "name" and an int property "value", set differently for 3 consecutive messages.
- **SelReceive.cpp** - Implements a synchronous listener, which listens on the queue "primaryqueue" for the messages which match the criteria specified in the message selector, and prints out the received messages.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cplusplusclientbuild.bat which compiles the C++ program.
2. Run the Sender by executing the *QSelSender.exe* executable file.
3. Run the synchronous receiver by executing the *QSelReceive.exe* file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## Mtptp

This directory contains one sample programs which illustrate basic JMS Sender/Receiver functionality using the FioranoMQ C++ Runtime Library multithreading support.

- **mtPtp.cpp** - The multithreaded version of basic PTP. Single Sender is created, sends 10 text messages on 'primaryQueue' and a single receiver blocking receive with timewait of 1 second reads the messages. Each executes on a separate thread. On receipt of 10 messages, the receiver notifies the main thread to end. Sender and Receiver threads are joined to the main thread.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cpplibbuild.bat which compiles the C++ program.
2. Run the mtPtp by executing the *mtPtp.exe* executable file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## reqrep

This directory contains two folders Basic and timeout

### basic

This directory contains two samples which illustrate JMS Request-Reply mechanism over Queues.

- **QueueRequestor.cpp** - Reads strings from standard input and sends the text messages on the queue "primaryQueue".
- **QueueReplier.cpp** - Implements an asynchronous listener, which listens on the queue "primaryQueue", and replies to the received message. The reply is sent on TemporaryQueue.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cpplibbuild.bat which compiles the C++ program.
2. Run the Replier by executing the *QueueReplier.exe* executable file.
3. Run the requestor by executing the *QueueRequestor.exe* file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)



## TimedOut

This directory contains two sample programs which illustrate Timed Request-Reply mechanism over Queues using the FioranoMQ C++ Runtime Library.

- **TimedQueueRequestor.cpp** - Reads strings from standard input and sends the text messages on the queue "primaryQueue". The Requestor waits for a specified time for the reply. If the reply is not received within the stipulated time requestor times out.
- **TimedQueueReplier.cpp** - Implements an asynchronous listener, which listens on the queue "primaryQueue", and replies on a TemporaryQueue.

To run these samples using FioranoMQ, perform the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cpplibbuild.bat which compiles the C++ program.
2. Run the replier by executing the *QueueReplier.exe* executable file.
3. Run the timed requestor by executing the *TimedQueueRequestor.exe* file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## SSL

This directory contains two sample programs which illustrate the basic JMS Send/Receive functionality over Secure Socket Layer using the FioranoMQ C++ Runtime Library.

- **Sender.cpp** - Reads strings from standard input and sends them on the queue "primaryqueue".
- **Receiver.cpp** - Implements a synchronous listener, which listens on the queue "primaryqueue", and prints out the received messages.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cpplibbuild.bat which compiles the C++ program.
2. Run the Sender by executing the *Sender.exe* executable file.
3. Run the synchronous receiver by executing the *Receiver.exe* file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## Transaction

This directory contains a sample programs which illustrate JMS Transaction functionality using the FioranoMQ C++ Runtime Library.

- **QTransaction.cpp** - Implements the sender and receiver, and uses the commit/rollback functionality to demonstrate JMS Transactions

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cpplibbuild.bat which compiles the C++ program.
2. Run the sample by executing the *QTransaction.exe* executable file. For proper results from the sample, ensure that there are no messages in the primaryQueue.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## PubSub Samples

### Admin

This directory contains one sample program which illustrates basic JMS Administration API functionality using the FioranoMQ C++ Runtime Library.

- **AdminTest.cpp** - Creates an Admin Connection with the MQServer and gets an MQAdminService object to create and delete Topics and TopicConnectionFactory and retrieves information of users connected from the server.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cpplibbuild.bat which compiles the C++ program.
2. Run the AdminTest by executing the *AdminTest.exe* executable file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

### Basic

This directory contains two sample programs which illustrate basic JMS Publisher/Subscriber functionality using the FioranoMQ C++ Runtime Library.

- **Publisher.cpp** - Reads strings from standard input and sends them on the topic "primarytopic".
- **Subscriber.cpp** - Implements a synchronous listener, which listens on the topic "primarytopic", and prints out the received messages.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called `cppclientbuild.bat` which compiles the C++ program.
2. Run the Publisher by executing the `Publisher.exe` executable file.
3. Run the synchronous subscriber by executing the `Subscriber.exe` file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## Dursub

This directory contains two sample programs which illustrate basic JMS DurableSubscriber functionality using the FioranoMQ C++ Runtime Library.

- **DurPublisher.cpp** - Reads strings from standard input and publishes PERSISTENT messages on the topic "primarytopic".
- **DurSubscriber.cpp** - Implements a durable subscriber using the client ID "DS\_Client\_1" and durable subscriber name "Sample\_Durable\_Subscriber", listening on the topic "primarytopic".

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called `cppclientbuild.bat` which compiles the C++ program.
2. Start the DurableSubscriber program first, so that the subscriber can register with the FioranoMQ Server.
3. Next, start the Publisher\_d program. When the program comes up, type in a few strings, pressing the Enter key after each string. The string is published and is received by the Durable Subscriber started in step (2) above.
4. Now, shut down the Durable Subscriber, but keep typing in messages into the Publisher program. These messages are automatically stored by the FioranoMQ Server, since a Durable Subscriber was previously registered on the topic to which the messages are being published.
5. After a while, restart the DurableSubscriber program. On restart, you would find that all messages that were published during the time that the durable subscriber was down are now made available to the subscriber.
6. Repeat steps (4) and (5) over. Each time, you would find that all messages published during the time that the Subscriber is down are immediately made available to the Subscriber when it restarts.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## HTTP

This directory contains four sample programs which illustrate the use of HTTP protocol for basic JMS PubSub functionality using the FioranoMQ C++ Runtime Library.

- **Subscriber.cpp** - Receives messages synchronously published on "primaryTopic". This program implements an synchronous listener to listen for messages published on "primaryTopic".
- **Publisher.cpp** - Implements a client application publishing user specified data on "primaryTopic". This program reads strings from standard input and publishes them on "primaryTopic".

To run this sample using FioranoMQ, do the following:

1. Compile the source file. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cppclientbuild.bat which compiles the C++ program.
2. Run the HttpSubscriber by executing the *Subscriber.exe* executable file.
3. Run the HttpPublisher by executing the *Publisher.exe* executable file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano installation directory. (This defaults to C:\PROGRA~1\Fiorano\FIORAN~2\fmq).

## HTTPS

This directory contains two sample programs which illustrate basic JMS Publisher/Subscriber functionality using the Fio C++ Runtime Library.

- **HPublisher.cpp** - Reads strings from standard input and sends them on the topic "primarytopic".
- **HSubscriber.cpp** - Implements a synchronous listener, which listens on the topic "primarytopic", and prints out the received messages.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cppclientbuild.bat which compiles the C++ program.
2. Run the Publisher by executing the *Publisher.exe* executable file.
3. Run the synchronous subscriber by executing the *Subscriber.exe* file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## Msgsel

This directory contains two sample programs which illustrate the use of message selectors using the FioranoMQ C++ Runtime Library.

- **SelSend.cpp** - Selector sends messages with the string property "name" and an int property "value", set differently for 3 consecutive messages.
- **SelRecv.cpp** - Implements a synchronous listener, which listens on the topic "primarytopic" for the messages which match the criteria specified in the message selector, and prints out the received messages.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cpplibbuild.bat which compiles the C++ program.
2. Run the Sender by executing the *SelSend.exe* executable file.
3. Run the synchronous receiver by executing the *SelRecv.exe* file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## Mtpubsub

This directory contains one sample programs which illustrate basic JMS Publish/Subscribe functionality using the FioranoMQ C++ Runtime Library multithreading support.

- **mtPubSub.cpp** - The multithreaded version of basic PubSub. Single Publisher is created, publishes 10 text messages on 'primaryTopic' and a single subscriber blocking receive with timewait of 1 second reads the messages. Each executes on a separate thread. On receipt of 10 messages, the subscriber notifies the main thread to end. Publisher and Subscriber threads are joined to the main thread.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cpplibbuild.bat which compiles the C++ program.
2. Run the mtPubSub by executing the *mtPubSub.exe* executable file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## Reqrep

This directory contains two folders basic and timeout.

### Basic

This directory contains two sample programs which illustrate JMS Request-Reply mechanism over Topics using the FioranoMQ C++ Runtime Library.

- **TopicRequestor.cpp** - Reads strings from standard input and sends the text messages on the topic "primaryTopic".
- **TopicReplier.cpp** - Implements an asynchronous listener, which listens on the topic "primaryTopic", and replies to the received. The reply is sent on TemporaryTopic.

To run these samples using FioranoMQ, perform the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cppclientbuild.bat which compiles the C++ program.
2. Run the Replier by executing the *TopicReplier.exe* executable file.
3. Run the requestor by executing the *TopicRequestor.exe* executable file.

### TimedOut

This directory contains two sample programs which illustrate Timed Request-Reply mechanism over Topics using the FioranoMQ C++ Runtime Library.

- **TimedTopicRequestor.cpp** - Reads strings from standard input and sends the text messages on the topic "primaryTopic". The Requestor waits for a specified time for the reply. If the reply is not received within the stipulated time requestor times out.
- **TopicReplier.cpp** - Implements an asynchronous listener, which listens on the topic "primaryTopic", and replies on a TemporaryTopic.

To run these samples using FioranoMQ, perform the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cppclientbuild.bat which compiles the C++ program.
2. Run the replier by executing the *TopicReplier.exe* executable file.
3. Run the timed requestor by executing the *TimedTopicRequestor.exe* file.

**Note:** To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. (**Default:** C:\PROGRA~1\Fiorano\FIORAN~2\fmq)

## SSL

This directory contains two sample programs which illustrate basic JMS Publisher/Subscriber functionality using the FioranoMQ C++ Runtime Library.

- **Publisher.cpp** - Reads strings from standard input and sends them on the topic "primarytopic".
- **Subscriber.cpp** - Implements a synchronous listener, which listens on the topic "primarytopic", and prints out the received messages.

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cppclientbuild.bat which compiles the C++ program.
2. Run the Publisher by executing the *Publisher.exe* executable file.
3. Run the asynchronous subscriber by executing the *Subscriber.exe* file.

Note: To run any of the C++ samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. **(Default: C:\PROGRA~1\Fiorano\FIORAN~2\fmq)**

## Transaction

This directory contains a sample programs which illustrate JMS Transaction functionality using the FioranoMQ C++ Runtime Library.

- **Transaction.cpp** - Implements the sender and receiver, and uses the commit/rollback functionality to demonstrate JMS Transactions

To run these samples using FioranoMQ, do the following:

1. Compile each of the source files. The %FMQ\_DIR%\clients\cpp\native\scripts directory contains a script called cppclientbuild.bat which compiles the C++ program.
2. Run the sample by executing the *Transaction.exe* executable file. For proper results from the sample, ensure that there are no messages in the primaryTopic

**Note:** To run any of the CPP samples, please ensure that environment variable FMQ\_DIR points to Fiorano's installation directory. **(Default: C:\PROGRA~1\Fiorano\FIORAN~2\fmq)**