

FioranoMQ® 9

C (Native) RTL Sample Applications 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

Native C-Runtime Examples.....	1
PTP Samples.....	2
Admin	2
Browser.....	2
msgsel	3
reqrep.....	3
basic	3
timeout.....	4
basic.....	4
Transactions	4
Http.....	5
Https	5
SSL.....	6
PubSub Samples	7
Admin	7
dursub	7
msgsel	8
basic.....	8
reqrep.....	9
basic	9
timeout.....	9
Transaction.....	10
Http.....	10
Https	11
ssl	11

Native C-Runtime Examples

This guide explains you about the sample programs used for C-RTL for PubSub and PTP operations.

PTP

- Admin
- Browser
- Message Selectors
- RequestReply
- TimedRequestReply
- SendReceive
- Transactions
- Http
- Https
- SSL

PubSub

- Admin
- Durable Subscribers
- Message Selectors
- Pubsub
- RequestReply
- TimedRequestReply
- Transactions
- Http
- Https
- SSL

These samples are available in %FMQ_DIR%\clients\c\native\samples directory.

PTP Samples

Admin

This directory contains one sample programs which illustrate basic JMS Administration API functionality using the FioranoMQ C Runtime Library.

- **AdminTest.c** - Uses C Native RTL Administration API to create and delete Queues, QueueConnectionFactory and retrieves information of users connected from the server.

To run these samples using FioranoMQ, do the following:

1. Compile the source files. For convenience, compiled version of the sources are included in this directory. The %FMQ_DIR%\clients\c\native\scripts directory contains a script called clientbuild.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\FioranoMQ9\fmq)

Browser

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

- **Sender.c** - Reads strings from standard input and sends them on the queue "primaryqueue".
- **Browser.c** - 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 version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called client-build.bat which compiles the C program.
2. Run the Sender by executing the Sender.exe executable file. Send some messages before running the browser application
3. Run the asynchronous receiver 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 \Fiorano\FioranoMQ\fmq)

msgsel

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

- **SelectorSend.c** - Selector sends messages with the string property "name" and an int property "value", set differently for 3 consecutive messages.
- **SelectorRecv.c** - Implements an asynchronous 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, perform the following steps:

1. Compile each of the source files. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called client-build.bat which compiles the C program.
2. Run the Sender by executing the SelectorSend.exe executable file.
3. Run the asynchronous receiver by executing the selectorRecv.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 \Fiorano\FioranoMQ\fmq)

reqrep

basic

This directory contains two sample programs which illustrate the request/reply abstraction supplied by the JMS API using a C requestor application and replying application

- **Requestor.c** - Reads strings from standard input and use it to send a request message on the queue "primaryqueue".
- **Replier.c** - Implements an asynchronous listener, which listens on the queue "primaryqueue", and replies to each request

To run these samples using FioranoMQ, perform the following steps:

1. Compile each of the source files. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclient-build.bat which compiles the C program.
2. Run the Replier by executing the Replier.exe executable file.
3. Run the requestor by executing the Requestor.exe file.

Note: To run any of the C samples, ensure that environment variable FMQ_DIR points to Fiorano installation directory (this defaults to \Fiorano\FioranoMQ\fmq).

timeout

This directory contains two sample programs which illustrate the timed request/ reply abstraction provided by FioranoMQ.

- **TimedRequestor.c** - Reads strings from standard input and use it to send a request on the queue "primaryqueue". It waits for the reply, for a specific time interval.
- **TimedReplier.c** - Implements an asynchronous listener, which listens on the queue "primaryqueue", and replies to each request.

To run these samples using FioranoMQ, perform the following steps:

1. Compile each of the source files. For convenience, compiled version of the %FMQ_DIR%\c\native\scripts directory contains a script called cclient-build.bat which compiles the C program.
2. Run the replier by executing the TimedReplier.exe executable file.
3. Run the timed requestor by executing the TimedRequestor.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 \Fiorano\FioranoMQ\fmq).

basic

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

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

To run these samples using FioranoMQ, perform the following steps:

1. Compile each of the source files. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclient-build.bat which compiles the C program.
2. Run the Sender by executing the *sender.exe* executable file.
3. Run the asynchronous receiver by executing the Receiver.exe file.

Note: To run any of the C samples, ensure that environment variable FMQ_DIR points to Fiorano installation directory (this defaults to \Fiorano\FioranoMQ\fmq).

Transactions

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

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

To run this sample using FioranoMQ, do the following:

1. Compile the source files. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclient-build.bat which compiles the C program.
2. Run the sample by executing the *transactions.exe* executable file. For proper results from the sample, ensure that there are no messages in the primary-Queue.

Note: To run any of the C samples, ensure that environment variable FMQ_DIR points to Fiorano's installation directory (this defaults to \Fiorano\FioranoMQ\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.

- **HttpReceiver.c** - Receives messages asynchronously on primaryQueue. This program implements an asynchronous listener to listen for messages published on the queue primaryQueue.
- **HttpSender.c** - 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. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclientbuild.bat which compiles the C program.
2. Run the HttpReceiver by executing the *httpReceiver.exe* executable file.
3. Run the HttpSender by executing the *httpSender.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 \Fiorano\FioranoMQ\fmq). Runtime dependency on Win Platform is on following files [c\native\lib]:

- gnu_regex.dll
- libeay32.dll
- ssleay32.dll
- zlib.dll

These are required to be present in path [either in same directory or in WINDOWS\System32].

Https

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

- **HttpsReceiver.c**- Receives messages asynchronously on primaryQueue. This program implements an asynchronous listener to listen for messages published on the queue primaryQueue.
- **HttpsSender.c**- 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. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclientbuild.bat which compiles the C program.
2. Run the HttpsReceiver by executing the *httpsReceiver.exe* executable file.
3. Run the HttpsSender by executing the *httpsSender.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 \Fiorano\FioranoMQ\fmq). Runtime dependency on Win Platform is on following files [c\native\lib]:

- gnu_regex.dll
- libeay32.dll
- ssleay32.dll
- zlib.dll

These are required to be present in path [either in same directory or in WINDOWS\System32].

SSL

This directory contains two sample programs which illustrate the basic JMS pub-sub and ptp functionality over Secure Socket Layer using the FioranoMQ C Runtime Library.

- **SSLReceiver.c** - Receives messages asynchronously on primaryQueue. This program implements an asynchronous listener to listen for messages published on the queue primaryQueue.
- **SSLSender.c** - 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:

To run this sample using FioranoMQ, do the following:

1. Compile the source file. For convenience, compiled version of the sources, are included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclientbuild.bat which compiles the C program.
2. Run the SSLReceiver by executing the *sslReceiver.exe* executable file.
3. Run the SSLSender by executing the *sslSender.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 \Fiorano\FioranoMQ\fmq). Runtime dependency on Win Platform is on following files [c\native\lib]:

- gnu_regex.dll
- libeay32.dll
- ssleay32.dll
- zlib.dll

These are required to be present in path [either in same directory or in WINDOWS\System32].

PubSub Samples

Admin

This directory contains one sample programs which illustrate basic JMS Administration API functionality using the FioranoMQ C Runtime Library.

- **AdminTest.c**- Uses C Native RTL Administration API to create and delete Topics, TopicConnectionFactory and retrieves information from the server.

To run these samples using FioranoMQ, do the following:

1. Compile the source files. For convenience, compiled version of the sources, are included in this directory. The %FMQ_DIR%\clients\c\native\scripts directory contains a script called cclientbuild.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.

(Defaults: c:\progra~1\Fiorano\FioranoMQ9\fmq)

dursub

This directory contains two sample programs which illustrate basic JMS Durable-Subscriber functionality using the FioranoMQ C Runtime Library.

- **Publisher_d.c** - Reads strings from standard input and publishes PERSISTENT messages on the topic "primarytopic".
- **DurableSubscriber.c** - 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, perform the following steps:

1. Compile each of the source files. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclient-build.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 (a) 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, ensure that environment variable FMQ_DIR points to Fiorano installation directory (this defaults to \Fiorano\FioranoMQ\fmq).

msgsel

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

- **SelectorSend.c** - Selector sends messages with the string property "name" and an int property "value", set differently for 3 consecutive messages.
- **SelectorRecv.c** - Implements an asynchronous 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, perform the following steps:

1. Compile each of the source files. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclient-build.bat which compiles the C program.
2. Run the Sender by executing the *selectorSend.exe* executable file.
3. Run the asynchronous receiver by executing the *selectorRecv.exe* file.

Note: To run any of the C samples, ensure that environment variable FMQ_DIR points to Fiorano's installation directory (default is \Fiorano\FioranoMQ).

basic

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

- **Publisher.c** - Reads strings from standard input and sends them on the topic "primarytopic".

- **Subscriber.c** - Implements an asynchronous listener, which listens on the topic "primarytopic", and prints out the received messages.

To run these samples using FioranoMQ, perform the following steps:

1. Compile each of the source files. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclient-build.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, ensure that environment variable FMQ_DIR points to Fiorano installation directory (default is \Fiorano\FioranoMQ).

reqrep

basic

This directory contains two sample programs which illustrate the request/reply abstraction supplied by the JMS API using a C requestor application and replying application

- **Requestor.c** - Reads strings from standard input and use it to send a request message on the topic "primarytopic".
- **Replier.c** - Implements an asynchronous listener, which listens on the topic "primarytopic", and replies to each request

To run these samples using FioranoMQ, perform the following steps:

1. Compile each of the source files. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclient-build.bat which compiles the C program.
2. Run the Replier by executing the *replier.exe* executable file.
3. Run the requestor by executing the *requestor.exe* file.

Note: To run any of the C samples, ensure that environment variable FMQ_DIR points to Fiorano installation directory (this default to \Fiorano\FioranoMQ).

timeout

This directory contains two sample programs which illustrate the request/reply abstraction supplied by the JMS API using a C requestor application and replying application

- **TimedRequestor.c** - Reads strings from standard input and use it to send a request message on the topic "primarytopic".
- **TimedReplier.c** - Implements an asynchronous listener, which listens on the topic "primarytopic", and replies to each request

To run these samples using FioranoMQ, perform the following steps:

1. Compile each of the source files. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclient-build.bat which compiles the C program.
2. Run the Replier by executing the *timedReplier.exe* executable file.
3. Run the requestor by executing the *timedRequestor.exe* file.

Note: To run any of the C samples, ensure that environment variable FMQ_DIR points to Fiorano installation directory (this defaults to \Fiorano\FioranoMQ\fmq).

Transaction

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

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

To run this sample using FioranoMQ, do the following:

1. Compile the source file. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclient-build.bat which compiles the C program.
2. Run the Replier by executing the Transactions.exe executable file. For proper results from the sample, ensure that there are no messages in the primary-Topic.

Note: To run any of the C samples, ensure that environment variable FMQ_DIR points to Fiorano installation directory (this defaults to \Fiorano\FioranoMQ\fmq).

Http

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

- **HttpSubscriber.c** - Receives messages asynchronously published on "primary-Topic". This program implements an asynchronous listener to listen for messages published on "primaryTopic".
- **HttpPublisher.c** - 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. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclientbuild.bat which compiles the C program.
2. Run the HttpSubscriber by executing the *httpSubscriber.exe* executable file.
3. Run the HttpPublisher by executing the *httpPublisher.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 \Fiorano\FioranoMQ\fmq). Runtime dependency on Win Platform is on following files [c\native\lib]:

- gnu_regex.dll
- libeay32.dll
- ssleay32.dll
- zlib.dll

These are required to be present in path [either in same directory or in WINDOWS\System32].

Https

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

- **HttpsSubscriber.c** - Receives messages asynchronously published on "primaryTopic". This program implements an asynchronous listener to listen for messages published on "primaryTopic".
- **HttpsPublisher.c** - 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. For convenience, compiled version of the sources is included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclientbuild.bat which compiles the C program.
2. Run the HttpsSubscriber by executing the *httpsSubscriber.exe* executable file.
3. Run the HttpsPublisher by executing the *httpsPublisher.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 \Fiorano\FioranoMQ\fmq). Runtime dependency on Win Platform is on following files [c\native\lib]:

- gnu_regex.dll
- libeay32.dll
- ssleay32.dll
- zlib.dll

These are required to be present in path [either in same directory or in WINDOWS\System32].

ssl

This directory contains two sample programs which illustrate the basic JMS pub-sub and ptp functionality over Secure Socket Layer using the FioranoMQ C Runtime Library.

- **SSLSubscriber.c** - Receives messages asynchronously published on "primary-Topic". This program implements an asynchronous listener to listen for messages published on "primaryTopic".

- **SSLPublisher.c** - 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. For convenience, compiled version of the sources, are included in this directory. The %FMQ_DIR%\c\native\scripts directory contains a script called cclientbuild.bat which compiles the C program.
2. Run the SSLSubscriber by executing the *sslSubscriber.exe* executable file.
3. Run the SSLPublisher by executing the *sslPublisher.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 \Fiorano\FioranoMQ\fmq). Runtime dependency on Win Platform is on following files [c\native\lib]:

- gnu_regex.dll
- libeay32.dll
- ssleay32.dll
- zlib.dll

These are required to be present in path [either in same directory or in WINDOWS\System32].