

WebSphere MQ Integrator - Publish / Subscribe client utility Version 1.1

15 July 2002

Kareem Yusuf PhD
WebSphere Business Integration
Worldwide Technical Sales

yusufk@us.ibm.com

Property of IBM

Take Note!

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

Second Edition, May 2002

This edition applies to Version 1.1 of *WebSphere MQ Integrator - Publish / Subscribe client utility* and to all subsequent releases and modifications unless otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2002.** All rights reserved. Note to US Government Users -- Documentation related to restricted rights -- Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule contract with IBM Corp.

Table of Contents

WebSphere MQ Integrator - Publish / Subscribe client utility	i
Table of Contents	iii
Notices.....	v
Trademarks and service marks	v
Acknowledgments.....	vi
Summary of Amendments	vii
Preface	viii
Bibliography.....	ix
Chapter 1. Introduction	1
Chapter 2. Installation and Setup.....	2
Packaging	2
Prerequisites	2
General Setup.....	2
Chapter 3. IH06 Quick Tour.....	3
Initial Configuration	3
The JMS Tab.....	4
The SCADA Tab	6
The MQe Tab.....	8
Chapter 4. Hands On Example	10
Scenario description.....	10
Configure JMS/MQ for Publish/Subscribe	10
Configure JMS Administered Objects	11
Publish/Subscribe Flow Configuration: SCADA and JMS/MQ	12
Create Message Flow: SCADAPublish	12
Create Message Flow: JMSPublish	13
Execute Message Flow.....	13
Publish/Subscribe Flow Configuration: MQe	17
Define MQe Queue Managers	17

Create Message Flow: MQePublish	21
Execute Message Flow:.....	23

Notices

The following paragraph does not apply in any country where such provisions are inconsistent with local law.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

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

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe any of the intellectual property rights may be used instead of the IBM product.

Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

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

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

Trademarks 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
- WebSphere MQ
- WebSphere MQ Integrator
- WMQI
- MQSeries Everyplace
- SupportPac

The following terms are trademarks of other companies:

- Windows NT, Windows 2000 Microsoft Corporation
- Java Sun Microsystems

Acknowledgments

I would especially like to thank Jonathan Woodford for developing the original prototype, Andy Stanford-Clark for technical guidance, and for providing the SCADA protocol implementation, which the utility makes use of and Neil Kolban for his assistance in developing the GUI.

For v1.1, I would additionally like to thank Ron Daniels and Dave Pugh for their assistance with implementing the support for MQSeries Everyplace (MQe).

Summary of Amendments

Date	Changes
28 February 2002	Initial release
21 May 2002	Introduced support for MQSeries EveryPlace (MQe) Included detailed hands-on example demonstrating use of the SupportPac with WMQI

Preface

The WebSphere MQ Integrator - Publish/Subscribe client utility encapsulates the APIs for JMS, SCADA, and MQSeries Everyplace (MQe), supporting multiple JMS, SCADA and MQe clients acting as publishers and subscribers from a simple GUI interface. The primary objective of the SupportPac is to assist in the development of WebSphere MQ Integrator (WMQI) solutions, by allowing WMQI functionality to be explored independently of the actual applications.

The SupportPac is a simple generic Publish/Subscribe client that allows potential solutions based on WebSphere MQ (using JMS), SCADA or MQe to be investigated. However, as the SupportPac's JMS client has been written as a pure JMS application, it is well positioned to exploit additional transports such as the high performance IP transport and MQSeries Everyplace (MQe) when their JMS implementations become available.

Although no official support is provided, the author is interested in hearing of any problems or suggestions for improvement for this SupportPac. If a bug is suspected, please send an email with a problem description. The author's email address is on the front cover of this document.

Bibliography

- *MQSeries Using Java*, IBM Corporation. SC34-5456-07.
- *MQSeries Publish/Subscribe Applications*, IBM Corporation Redbook, SG24-6282-00.
- *WebSphere MQ Integrator, Using the Control Center v2.1*, IBM Corporation, SC34-5602-04
- *WebSphere MQ Integrator, Programming Guide v2.1*, IBM Corporation, SC34-5603-03
- *MQSeries Everyplace for Multiplatforms – MQe Explorer User Guide v1.26, ES02*

Chapter 1. Introduction

The Publish/Subscribe messaging paradigm offers an effective means of disseminating information to multiple recipients, specifically when the number of producers (publishers) and consumers (subscribers) of the information can dynamically change, and indeed may not be known. WebSphere MQ Integrator (WMQI) is a functionally rich Publish/Subscribe broker, matching publications to subscribers, across a variety of transmission protocols.

Currently, publishers and subscribers may interact with WMQI over MQSeries utilizing their API of choice (MQI, AMI, and JMS), they can also employ the MQe protocol and API, or the SCADA protocol which is designed for use by remote telemetry devices, such as oil pipeline sensors. This SupportPac provides a utility that allows the Publish/Subscribe functionality of WMQI to be explored and demonstrated without having to invest a lot of time in developing driving applications (that is, publishers and subscribers).

The Publish/Subscribe client utility encapsulates the APIs for JMS, SCADA and MQe, and supports multiple JMS, SCADA and MQe clients acting as publishers and subscribers from a simple GUI interface. The primary objective of the SupportPac is to assist in the development of WMQI solutions, by allowing WMQI functionality to be explored independently of the actual applications. It should be noted that the SupportPac is not intended as a performance test tool, it is a simple generic Publish/Subscribe client, which allows solutions based on MQSeries (via JMS), SCADA or MQe to be investigated. Consequently, it is limited by the limitations of the APIs utilized, and automatically benefits from any improvements to these underlying APIs.

Use of SupportPac assumes a basic understanding of the use of WMQI, MQe and SupportPac MA88 (MQSeries classes for Java and MQSeries classes for JMS). Associated documentation is referenced throughout this guide, and it is assumed that the user is familiar with this material.

Chapter 2. Installation and Setup

Packaging

The SupportPac is composed of the following files:

- `com.ibm.IH06.jar` Contains application files
- `com.ibm.mqisdpc.jar` Java implementation of the SCADA protocol, rendered as an API
- `IH06.dat` Default configuration file
- `IH06.cmd` Command file for launching the utility on the Windows platform
- `ih06.pdf` This User Guide

To install, simply unzip the distribution file, placing these files in an executable directory (for example, `c:\utilities\IH06`).

Prerequisites

The SupportPac is a pure Java application and has no built-in platform dependences. In principle, it should therefore be able run on any platform that provides a suitable JVM, and supports the other prerequisites. The following are required:

- Java 2 SDK, Standard Edition v1.3.1, or higher
- WebSphere MQ Integrator v 2.1, or higher
- SupportPac MA88 v5.2 or higher (v5.2.1 recommended, released April 2002)

Associated product documentation (see **Bibliography**) discusses the installation and configuration of the prerequisites.

General Setup

In order to enable WMQI to act as a Publish/Subscribe broker, it is necessary to create and deploy "Publish" message flows, which at the barest minimum have an input node and a publication node. With IH06 three input nodes are relevant:

- **MQInput node.** This supports the MQ JMS publisher, the queue name attribute should match the value of the publish queue attribute of the *TopicConnectionFactory* (For example, `SYSTEM.BROKER.DEFAULT.STREAM`)
- **ScadaInput node.** This supports the SCADA publisher and subscriber, and holds as key attribute the port on which it will receive messages. The default value is 1883
- **MQe Input node.** This supports the MQe publisher and subscriber, and sets up an MQe Gateway with which other MQe Queue Managers communicate.

Being input nodes, they all have a default tab that allows the message properties of the incoming message to be defined, enabling messages to be parsed and manipulated before transmission to the Publication node. Chapter 4 provides a detailed hands-on example that describes how to configure WMQI message flows for Publish/Subscribe, and use IH06 to exercise them.

Chapter 3. IH06 Quick Tour

Initial Configuration

The *IH06.cmd* holds the commands required to start the utility, and provides a shortcut on the Windows platform. It contains the following:

```
SET PATH=.,;%PATH%
```

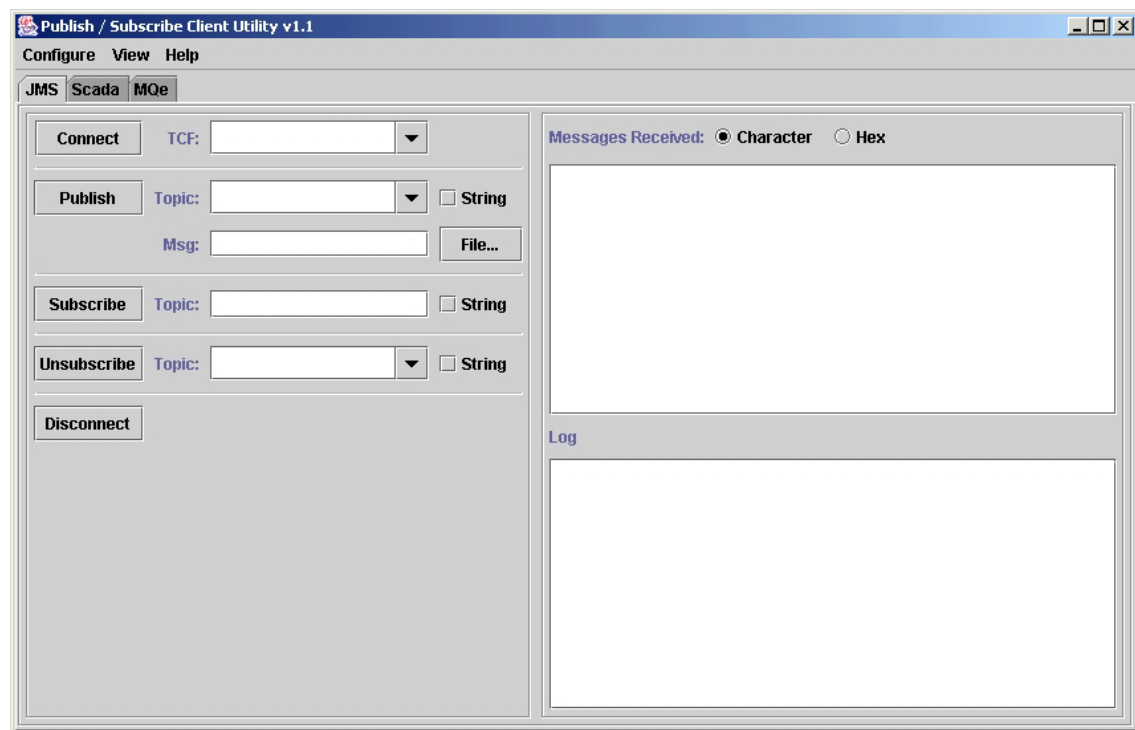
```
SET CLASSPATH=.,com.ibm.IH06.jar;com.ibm.mqisdj.jar;%CLASSPATH%
```

```
java com.ibm.IH06.IH06
```

The *SET PATH* command sets the runtime path to the system path, the key requirement here is that the directory hosting the java runtime (java.exe) be on the path. For example, c:\jdk1.3.1\bin.

The *SET CLASSPATH* command sets the runtime classpath to the system classpath plus the current directory and the application jars. It is assumed here that the system classpath has been appropriately setup for SupportPac MA88. Clearly, if one is not using the IH06.cmd file (for instance, on a UNIX system), then the IH06 jars must be added to the system or runtime environment classpath.

java com.ibm.IH06.IH06 runs the main executable IH06.class resulting in the screen shown below.

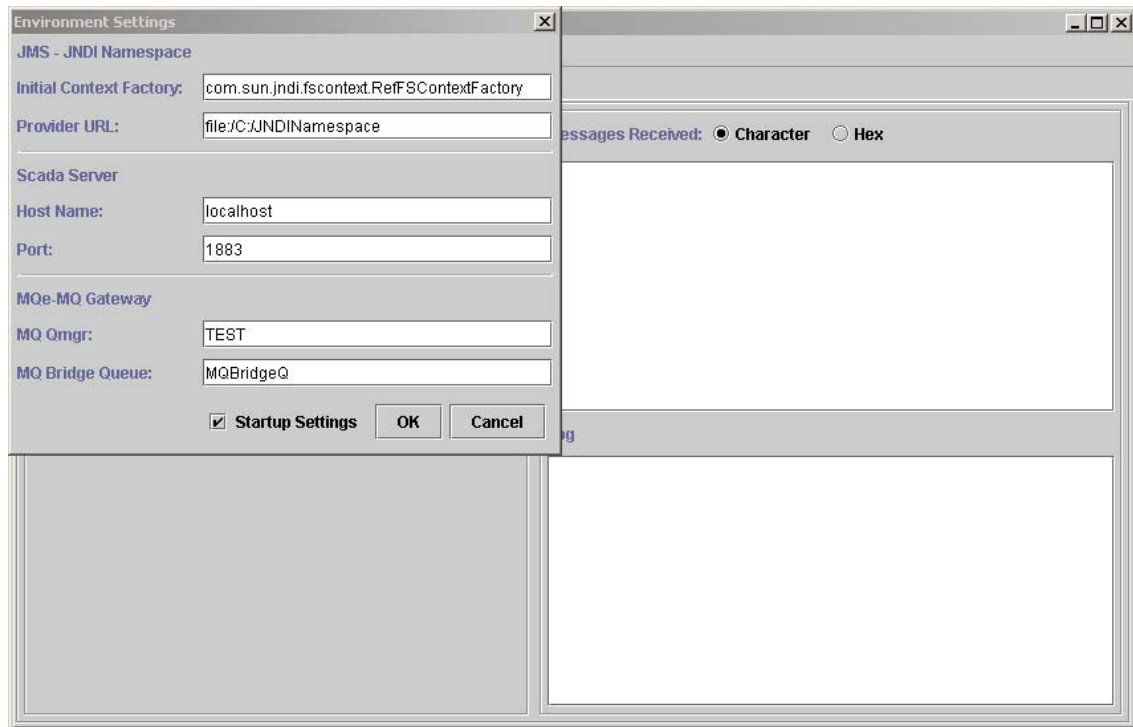


The first step is to configure the utilities environment. This is done by selecting the *Configure->Environment...* menu. The resulting dialog box allows the user to specify configuration options for JMS, SCADA and MQe. The configuration options are encapsulated by the following variables:

- **JNDI Initial Context Factory.** This identifies the type of namespace being used and will typically have the same value as that set in the *JMSAdmin.config* file used by SupportPac MA88.

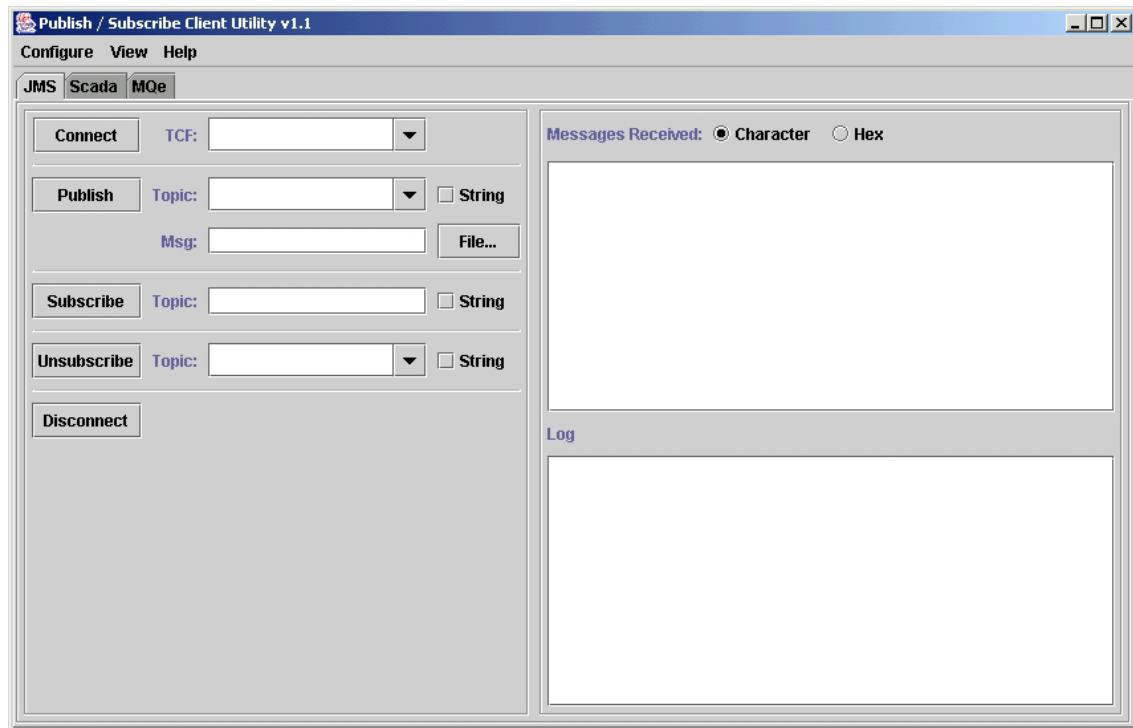
- **JNDI Provider URL.** This defines the location of the namespace and will typically have the same value as that set in the *JMSAdmin.config* file, used by MA88.
- **SCADA Server Hostname.** The machine name where WMQI is running. If this is remote to the utility a network path (TCP/IP) must exist to the broker.
- **SCADA Server Port.** The port specified in the *SCADAInput* node, by default 1883.
- **MQe-MQGateway MQ Qmgr.** Name of the MQ Queue Manager hosting WMQI
- **MQe-MQGateway MQ Bridge Queue.** Name of the MQ Bridge Queue defined in the MQe Input Node

If checked, the “startup settings” checkbox results in the configuration being persisted to a file (IH06.dat) in the install directory. This allows the configuration to be restored when the utility is restarted. It is relevant to note that the JMS-JNDI namespace, SCADA Server and MQe Gateway can be configured independently of each other. Indeed, the default configuration file that ships with the SupportPac only sets up the SCADA Server to localhost and port 1883, leaving all other configuration options blank. Please note that if the JNDI Namespace is not setup or does not exist an “Unable to setup JNDI Namespace” error is logged to the log window. However, this does not prevent the use of the SCADA or MQe Client.



The JMS Tab

The JMS Tab provides access to the JMS Publish/Subscribe implementation. Interaction is based on five commands; *Connect*, *Publish*, *Subscribe*, *Unsubscribe*, and *Disconnect*.



JMS Connect

To *Connect*, type in the name of the *TopicConnectionFactory* (TCF) to be used. The status of the connection (started or exception) is displayed in the *Log* window. If a connection is successfully established, the TCF is added to the combo box. Another TCF name can be entered and connected to. The tool thus supports multiple active connections using different TCFs. Before issuing any other command, ensure that the desired active connection is selected in the combo box.

JMS Subscribe

The *Subscribe* command requires that you either supply the name of a Topic object to be retrieved from the JNDI namespace, or that you type in the actual topic string. This is in keeping with the JMS API that offers both options. If a topic string (as opposed to the name of a topic object) is supplied, the *String* checkbox should be selected, otherwise reference to a topic object is assumed. On successful completion of a subscription request, the Topic is added to the *Topic* combo boxes associated with the Publish and Unsubscribe commands. This reduces the need to type in the same information multiple times. While you can have multiple subscribers to different topics, you cannot subscribe to a topic twice. In keeping with its design point, the utility only creates non-durable subscribers.

JMS Publish

The *Publish* command similarly takes as argument the name of a Topic object or a topic string. The message can be text typed into the *Msg Field* or can be taken from a file (selected using the file button). If data is read in from a file, the file path is displayed in the *Msg Field*. The file path begins with the special characters *::file:/.* The utility uses this to determine that the message should be sent from a populated buffer, as opposed to the contents of the *Msg Field* itself.

If a subscriber is registered for the published topic, the publication arrives in the *Messages Received* window. The format of the display is *topic::message*. By default the message is displayed as a character string. However by selecting the *Hex* radio button the messages received can be displayed

in hex format. It should be noted that switching between hex and character display, affects the next message to be received. Messages already received are not re-displayed.

JMS Unsubscribe

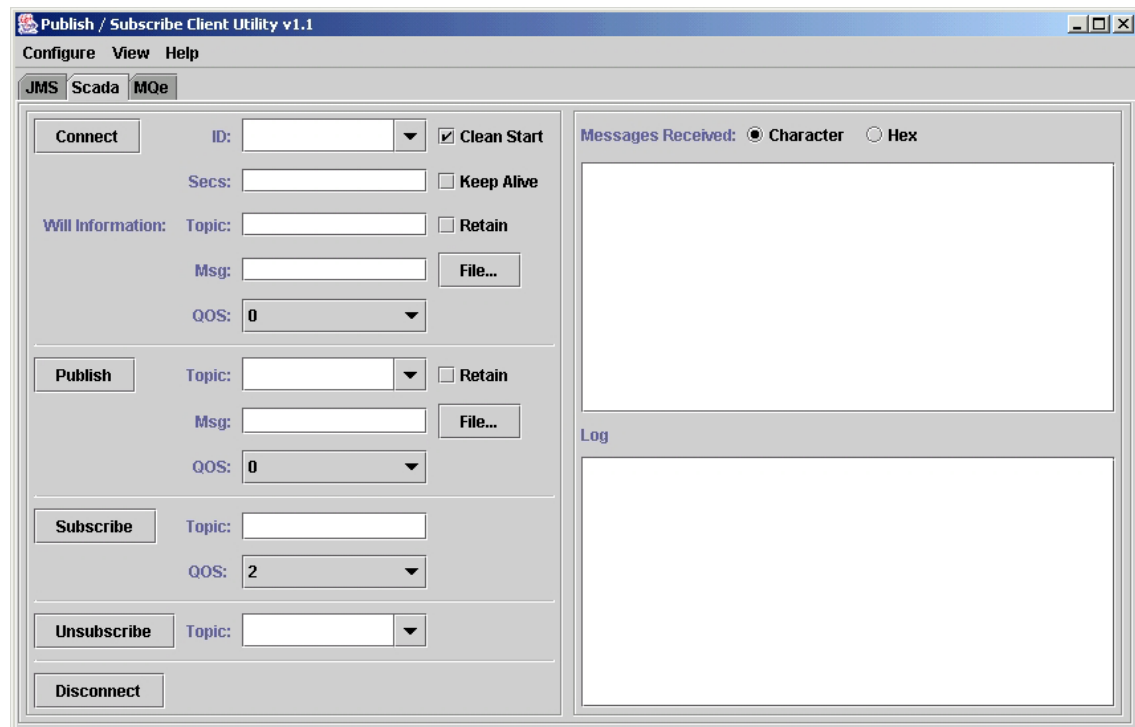
The *Unsubscribe* command, deregisters a subscriber to a specified topic. The topic can be similarly encapsulated within a Topic object, or can be explicitly defined as a string. If successful the topic is removed from the *Unsubscribe* and *Publish* combo boxes.

JMS Disconnect

The *Disconnect* command closes the active TCF connection. On successful completion it removes the TCF entry from the *Connect* combo box.

The SCADA Tab

The *SCADA* tab encapsulates access to the SCADA protocol, enabling the utility to act as a SCADA publisher or subscriber. As with the *JMS* tab, interaction is based around five commands: *Connect*, *Publish*, *Subscribe*, *Unsubscribe* and *Disconnect*.



SCADA Connect

As with *JMS Connect*, the *SCADA Connect* command establishes an active connection to the broker, and similarly supports the creation of multiple connections. A number of parameters may be specified on the *Connect* call. *WebSphere MQ Integrator, Programming Guide v2.1*, SC34-5603-03, discusses in detail the SCADA protocol and associated parameters. A brief overview will be offered here.

- **ID:** This is the client identifier and forms the basis for identifying an active connection

- **Clean Start Flag:** On *Connect* if the *Clean Start* flag is checked (default setting), the broker discards any outstanding messages and deletes all subscriptions that may exist for the specified client ID, before establishing a new connection.
- **Keep Alive Flag:** A checked flag activates the *Keep Alive* function that tells the broker the maximum time interval (specified in the *Seconds* field), between receiving messages from this client. Giving the broker the ability to detect dropped clients without waiting for the long TCP/IP time-out.
- **Will Information:** The SCADA protocol provides a mechanism by which a client can define a message to be published on its behalf by the broker, in the event that contact is lost with the client.
- **Will Topic:** The topic against which the will message will be published
- **Will Message:** The message that will be published
- **Will Retain flag:** If checked the message will be retained on publication
- **Will Quality of Service:** The message can be delivered with one of three qualities of service (QOS). QOS 0 suggests *at most once delivery*, as such the message is delivered according to the best efforts of the underlying TCP/IP transport. No response is expected and no retry semantics are defined. QOS 1 maps to *at least once delivery*, which implies that delivery is acknowledged, and retries can occur, however the message can arrive more than once. QOS2 *exactly once delivery*, employs additional flows to ensure that a message is delivered only once.

It is useful to note that the only parameter that is required to effectively connect to the broker is the client ID. On *Connect* status messages are logged to the *Log* window.

SCADA Subscribe

To *Subscribe*, a topic string is entered and a quality of service selected. On successful completion of a subscription request, the Topic is added to the *Topic* combo boxes associated with the *Publish* and *Unsubscribe* commands. This reduces the need to type in the same information multiple times. The utility allows multiple subscribers to be registered against multiple topics.

SCADA Publish

The SCADA *Publish* command takes as arguments a topic string, a message, the quality of service associated with transmission and whether the publication is to be retained by the broker. The message can be text typed into the *Msg Field* or can be taken from a file (selected using the *File* button). If data is read in from a file, the file path is displayed in the *Msg Field*. The file path begins with the special characters *::file:/. The utility uses this to know that the message should be sent from a populated buffer, as opposed to the contents of the Msg Field itself.*

If a subscriber is registered for the published topic, the publication arrives in the *Messages Received* window. The format of the display is *topic::message*. By default the message is displayed as a character string, however by selecting the *Hex* radio button the messages received can be displayed in hex format. It should be noted that switching between hex and character display affects the next message to be received. Messages already received are not re-displayed.

SCADA Unsubscribe

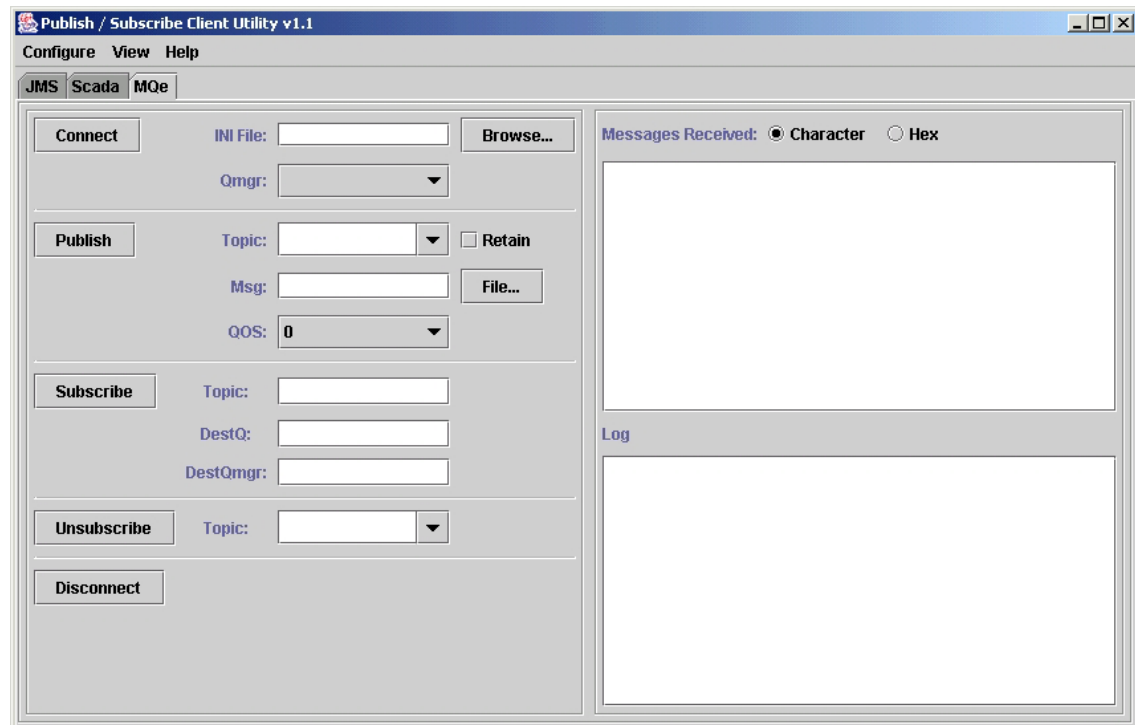
The *Unsubscribe* command, deregisters a subscriber to a specified topic. If successful the topic is removed from the *Unsubscribe* and *Publish* combo boxes.

SCADA Disconnect

The *Disconnect* command closes the active connection. On successful completion it removes the Client ID entry from the *Connect* combo box.

The MQe Tab

The *MQe* tab encapsulates access to MQe, enabling the utility to act as a MQe publisher or subscriber. In keeping with the common interaction design, five commands are implemented : *Connect*, *Publish*, *Subscribe*, *Unsubscribe* and *Disconnect*.



MQe Connect

IH06 uses an MQe Queue Manager *.ini* file to configure and start the queue manager, as discussed in Chapter 4, the *.ini* file can be created using the MQe Explorer (ES02). The *Browse* button can be used to locate the *.ini* file, or a fully qualified path entered into the *INI File Field*. The status of the connection (started or exception) is displayed in the *Log* window. If a connection is successfully established, the name of the MQe Queue Manager is added to the combo box. As MQe only supports a single Queue Manager running in a given JVM, an attempt to connect to another Queue Manager will result in an exception.

MQe Subscribe

The *Subscribe* command takes as argument a topic string, and the name of the MQe Queue and MQe Queue Manager where matching publications are to be delivered. On successful completion of a subscription request, the Topic is added to the *Topic* combo boxes associated with the Publish and Unsubscribe commands. This reduces the need to type in the same information multiple times. The utility allows multiple subscribers to be registered against multiple topics.

MQe Publish

The *Publish* command similarly takes as argument a topic string. The message can be text typed into the *Msg Field* or can be taken from a file (selected using the file button). If data is read in from a file, the file path is displayed in the *Msg Field*. The file path begins with the special characters *::file:/.* The utility uses this to determine that the message should be sent from a populated buffer, as opposed to the contents of the *Msg Field* itself.

If a subscriber is registered for the published topic, the publication arrives in the *Messages Received* window. The format of the display is *topic::message*. By default the message is displayed as a character string. However by selecting the *Hex* radio button the messages received can be displayed in hex format. It should be noted that switching between hex and character display, affects the next message to be received. Messages already received are not re-displayed.

MQe Unsubscribe

The *Unsubscribe* command, deregisters a subscriber to a specified topic. If successful the topic is removed from the *Unsubscribe* and *Publish* combo boxes.

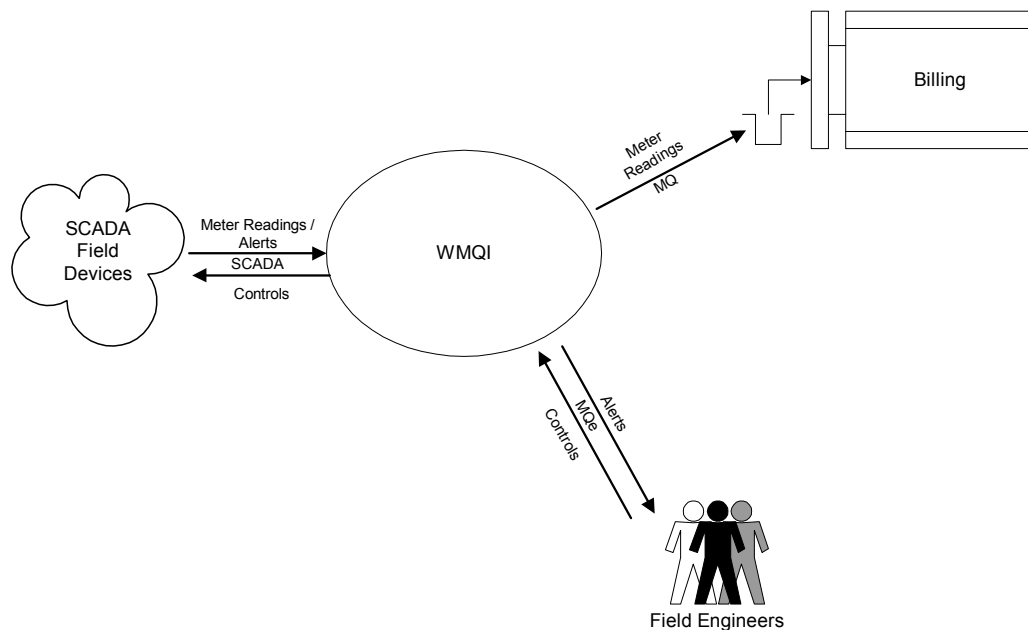
MQe Disconnect

The *Disconnect* command closes the MQe Queue Manager. On successful completion it removes the queue manager entry from the *Connect* combo box.

Chapter 4. Hands On Example

Scenario description

The hands-on example is based on a scenario that considers the middleware infrastructure required to distribute event messages among a number of heterogeneous devices. The scenario focuses on a utilities company, which uses SCADA based devices to monitor the usage of resources. The SCADA devices publish meter readings using MQIsdp to WMQI, where the readings are sent over MQ to a subscribing billing application. The monitoring devices also publish alert messages in the event that things go wrong. Field engineers on duty, receive the alert messages on PDA devices running MQE, and can send control messages to the monitoring devices or physically go to the device location. In this scenario, the Broker handles distribution only and does not provide any transformation services; this is akin to the role WebSphere MQ Event Broker would play in the enterprise. Indeed the exercises done as part of this scenario using WMQI could also be executed using the WebSphere MQ Event Broker.



Realization of the scenario involves configuring WMQI to support the different publishing/subscribing applications, the example details the essential steps required. It will be observed that the required message flows are very simple in design.

Configure JMS/MQ for Publish/Subscribe

Before using MA88 for Publish/Subscribe it is necessary to setup a number of system queues on the queue manager that the JMS application will connect too. In our scenario this is the queue manager that is hosting the Broker.

1. Open a command prompt and navigate to **<MA88 Install dir>\javalbin**, in the directory you will find a runmqsc script file called MQJMS_PSQ.mqsc
2. At a command prompt type:
runmqsc <queue manager name> < MQJMS_PSQ.mqsc

3. The required system queues are now created.

Configure JMS Administered Objects

The JMS subscribing application requires a number of JMS administered objects to be defined. SupportPac MA88 offers a choice of three supported namespaces; LDAP, File System and WebSphere Persistent Name Server (part of the WebSphere Application Server). Setting up the namespace of choice and populating it with objects using the *JMSAdmin* tool supplied in SupportPac MA88 is discussed in Chapter. 5, *MQSeries Using Java*, SC34-5456-07.

1. Open a command prompt, navigate to **<MA88 Install dir>\java\bin**, and start JMSAdmin. You should see an InitCtx prompt.
2. Define a Topic Connection Factory to represent the Broker Queue Manager

DEFINE TCF(BROKER)

This assumes that your queue manager is specified as the default queue manager). In defining the *TopicConnectionFactory*, it should be remembered that MA88v5.2 utilizes the MQRFH header for Publish/Subscribe and thus still uses stream queues. The default value for the publish queue (stream) is SYSTEM.BROKER.DEFAULT.STREAM. It is advisable to stick with this default when running against a WMQI broker as this avoids having the subscriber topic being further qualified with the stream name in the case when its non-default. A good discussion of stream queues and considerations when using WMQI can be found in Chapter. 5, *MQSeries Publish/Subscribe Applications*, SG24-6282-00.

If you have MA88v5.2.1 (released April, 2002), then you may be aware that it now supports the use of the MQRFHv2 header for Publish/Subscribe. The choice of header used is affected by setting the BROKERVER attribute of the *TopicConnectionFactory* to V2. When using the MQRFHv2 header one can supply any queue name for the publish queue, without fear of the subscriber topic being further qualified.

3. Define a Topic object to represent the order request queue

DEFINE T (READINGS) TOPIC(meter/reading/)

4. Display the defined objects e.g. (DIS CTX) to confirm their creation. You can explore the object definitions by typing DIS TCF (<name>) or DIS T(<name>)
5. Type, "end" to close JMSAdmin.

```

C:\ Command Prompt - jmsadmin
5648-C60 (c) Copyright IBM Corp. 1999. All Rights Reserved.
Starting MQSeries classes for Java(tm) Message Service Administration
InitCtx> def tcf<BROKER>
InitCtx> DEF T<READINGS> TOPIC<meter/reading/>
InitCtx> dis ctx

  Contents of InitCtx

      .bindings
a  EX3.WEBQCF          java.io.File
a  EX3.WEBREQUESTQ   com.ibm.mq.jms.MQQueueConnectionFactory
a  BROKER             com.ibm.mq.jms.MQQueue
a  EX3.WEBREPLYQ     com.ibm.mq.jms.MQTopicConnectionFactory
a  READINGS          com.ibm.mq.jms.MQQueue
a                   com.ibm.mq.jms.MQTopic

6 Object(s)
0 Context(s)
6 Binding(s), 5 Administered

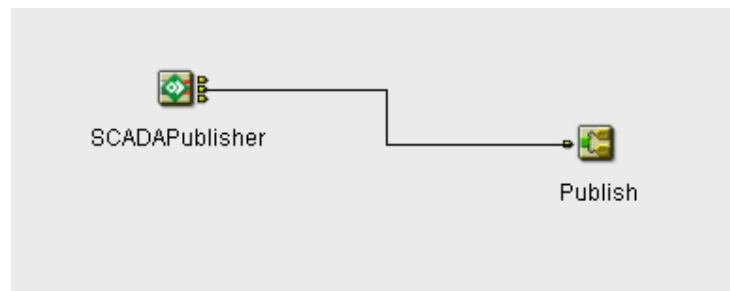
InitCtx>

```

Publish/Subscribe Flow Configuration: SCADA and JMS/MQ

Create Message Flow: SCADAPublish

Open the WMQI Control Center, switch to the Message Flow Tab, and create a new message flow, called “**SCADAPublish**”. Build the message flow as shown below and set the attributes on the nodes as described. If not explicitly mentioned, you can accept the default values for the node properties.



1. SCADAInput Node: SCADAPublisher

- a) View the Basic properties tab, note the port number the Node will be listening too. By default it is port 1883

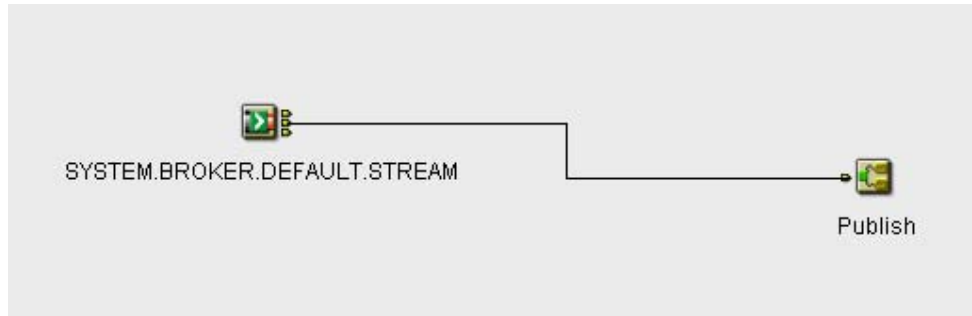
2. Publication Node: Publish

- a) Wire the “out” terminal from SCADAPublisher to the “in” terminal of Publish

3. Check in SCADAPublish and switch to the Assignments tab. Check out the execution group and assign SCADAPublish, check in and deploy. Check Log tab and confirm successful deployment

Create Message Flow: JMSPublish

While the scenario does not call for the MQ application to publish messages, the following steps describe the process required to enable this function. Open the WMQI Control Center, switch to the Message Flow Tab, and create a new message flow, called “**JMSPublish**”. Build the message flow as shown below and set the attributes on the nodes as described. If not explicitly mentioned, you can accept the default values for the node properties.



1. MQInput Node: SYSTEM.BROKER.DEFAULT.STREAM

On the Basic properties tab specify the queue name “SYSTEM.BROKER.DEFAULT.STREAM”.

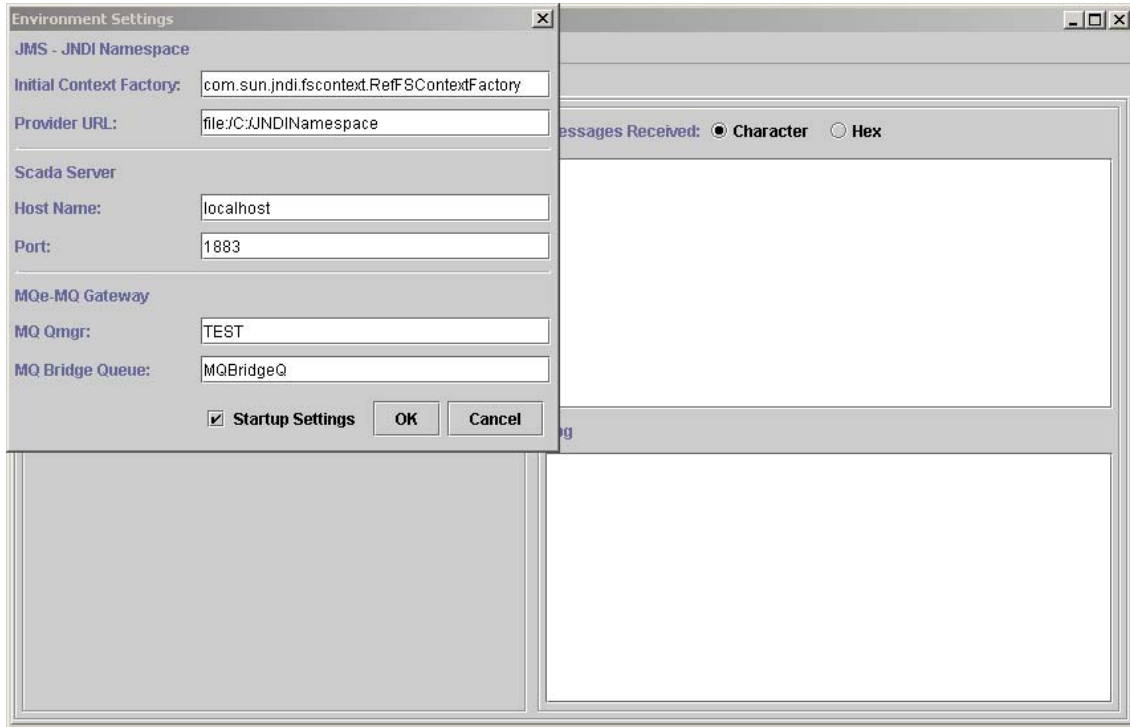
2. Publication Node: Publish

- a) Wire the “out” terminal from SYSTEM.BROKER.DEFAULT.STREAM to the “in” terminal of Publish
3. Check in JMSPublish and switch to the Assignments tab. Check out the execution group and assign JMSPublish, check in and deploy. Check Log tab and confirm successful deployment

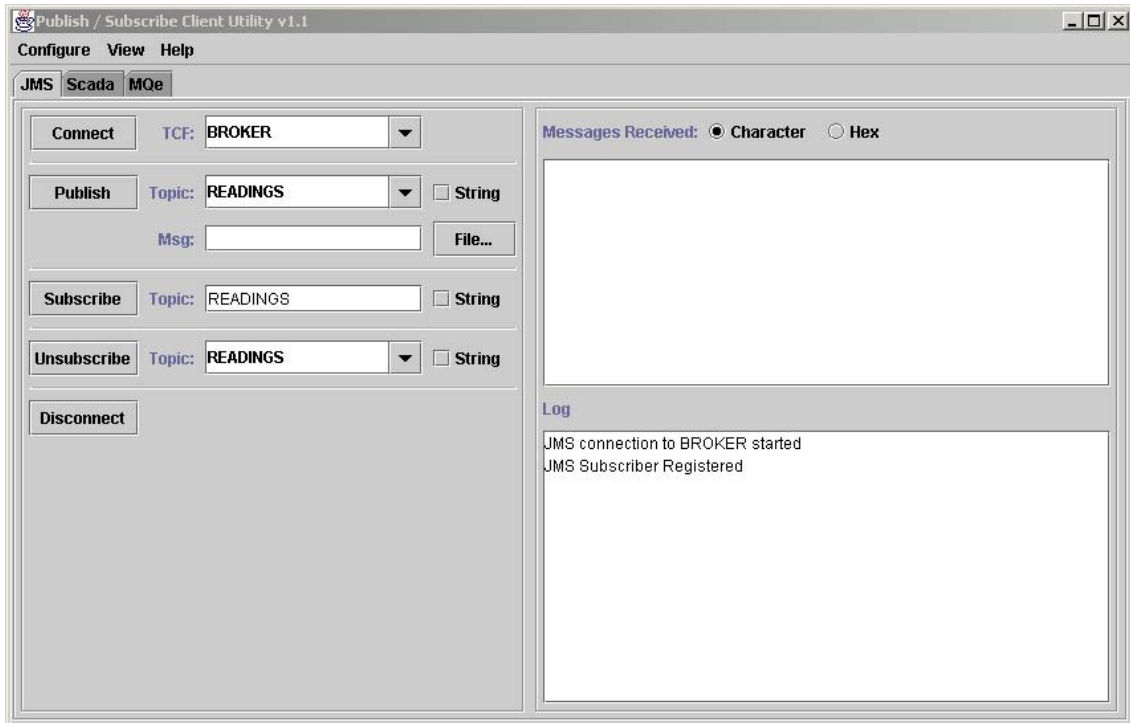
Execute Message Flow

The exercise uses supportpac IH06 to simulate SCADA Publishers and the JMS/MQ subscriber.

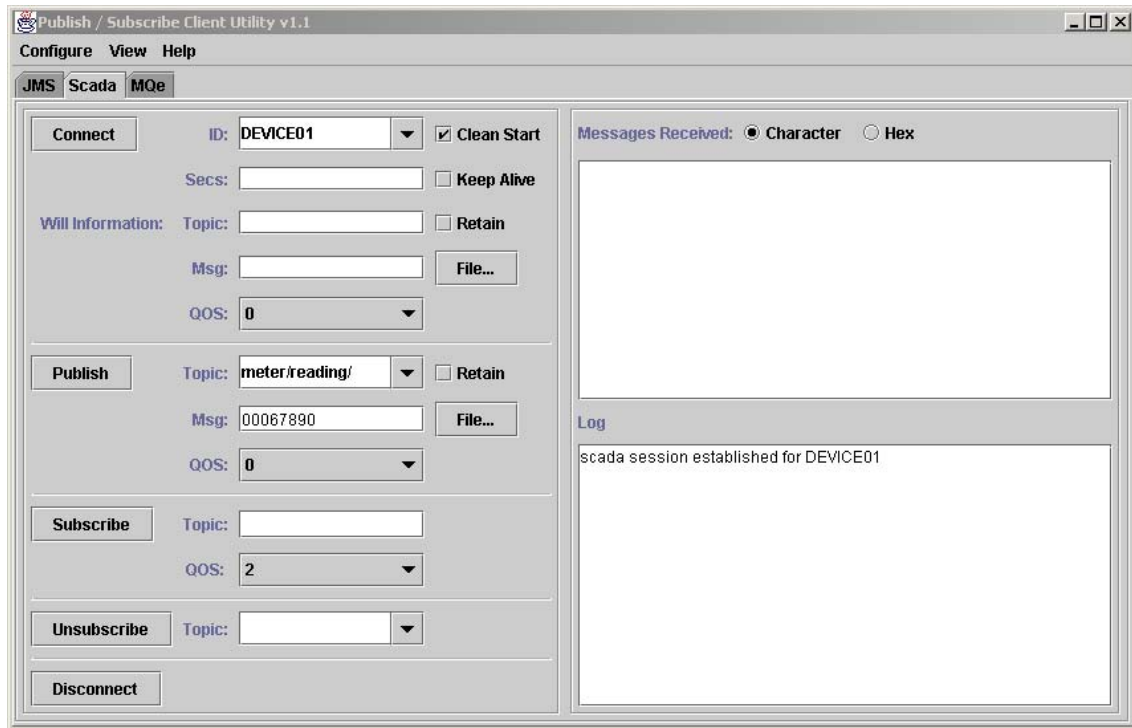
1. Open a command prompt and navigate to your IH06 install directory, run IH06.cmd
2. Check that the environment configuration matches your environment (**Configure-> Environment...**). The settings identify the JNDI Namespace and the machine/port on which the SCADAInput Node is listening. You can ignore the settings for MQe



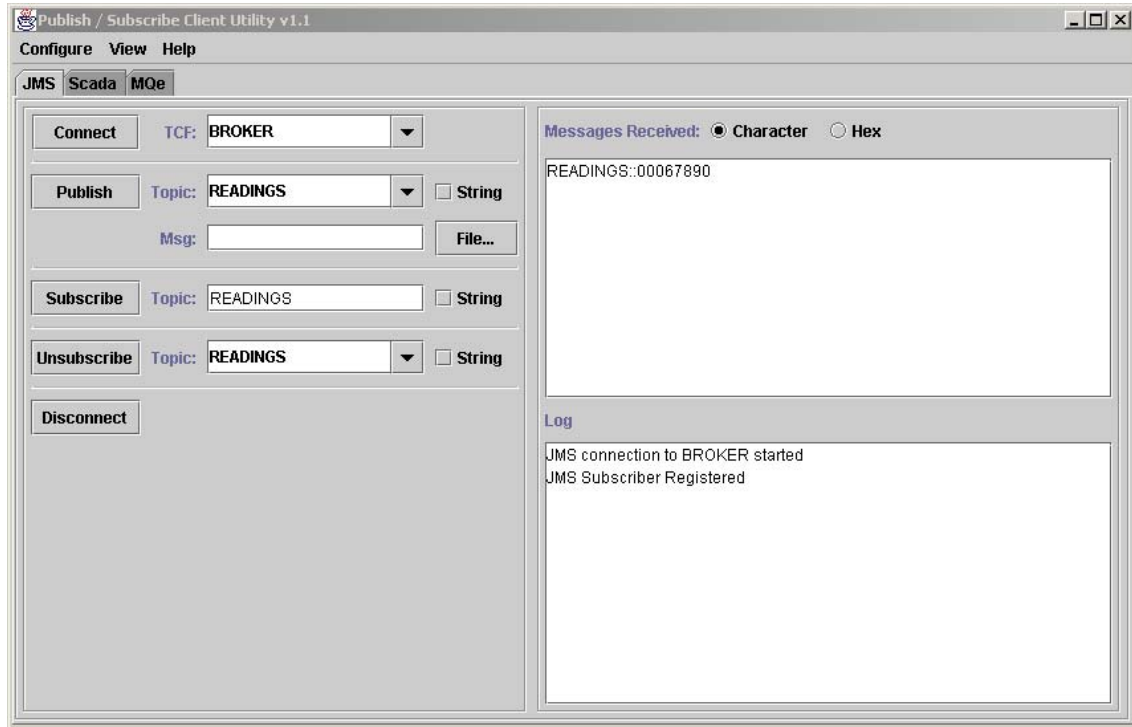
3. Switch to the JMS Tab to configure the JMS Subscriber; In the Connect panel supply the name of the TCF that represents the broker (**BROKER**). Click "Connect".
4. In the subscriber panel, supply the name of the previously defined TOPIC object (**READINGS**). Click "Subscribe"



5. Switch to the SCADA Tab to configure the SCADA Publisher; in the connect panel supply a client ID (**DEVICE01**), you can accept all the other defaults. Click “Connect”
6. In the Publisher panel, supply the topic “meter/reading/”, recall that this corresponds to the topic string defined in the JMS Topic object READINGS
7. Type in a message e.g. “**DEVICE01;00067890**” (you may make up your own message if you wish. Click “Publish”



8. Switch to the JMS Tab; Observe, that the subscriber has received the publication



9. You have successfully configured WMQI to support publish/subscribe using SCADA and MQ transports. You can further explore WMQI functionality reversing the role of publisher and subscriber over single or multiple transports.
10. Inspection of the Subscriptions tab in the WMQI Control Center shows the registered JMS Subscription.

Topic	User	Broker	Subscription P...	Registration Date	Client	Content Filter
\$\$SYS/Broker+/Subscri...	yusufk	STEW		Apr 8, 2002	mqrh2:TEST:SYSTEM.BROKER.ADMIN.REPLY:4366674d6772...	
\$\$SYS/Broker+/Status	yusufk	STEW		Apr 8, 2002	mqrh2:TEST:SYSTEM.BROKER.ADMIN.REPLY:4366674d6772...	
\$\$SYS/Broker+/warning...	yusufk	STEW		Apr 8, 2002	mqrh2:TEST:SYSTEM.BROKER.ADMIN.REPLY:4366674d6772...	
\$\$SYS/Broker+/Status	yusufk	TEST		Jan 4, 2002	mqrh2:TEST:SYSTEM.BROKER.ADMIN.REPLY:4366674d6772...	
meter/reading/	yusufk	STEW		Apr 9, 2002	mqrh2:TEST:SYSTEM.JMS.ND.SUBSCRIBER.QUEUE:414d5120...	
\$\$SYS/Broker+/Status/E...	yusufk	STEW		Apr 8, 2002	mqrh2:TEST:SYSTEM.BROKER.ADMIN.REPLY:4366674d6772...	
\$\$SYS/Broker+/Status/E...	yusufk	TEST		Jan 4, 2002	mqrh2:TEST:SYSTEM.BROKER.ADMIN.REPLY:4366674d6772...	
\$\$SYS/Broker+/Subscri...	yusufk	TEST		Jan 4, 2002	mqrh2:TEST:SYSTEM.BROKER.ADMIN.REPLY:4366674d6772...	
\$\$SYS/Broker+/warning...	yusufk	TEST		Jan 4, 2002	mqrh2:TEST:SYSTEM.BROKER.ADMIN.REPLY:4366674d6772...	
\$\$SYS/SCADA#	yusufk	STEW		Apr 9, 2002	mqrh2:#SCADAoutputNode#:0.0679757375666b000000000000...	

11. On the SCADA tab click "Disconnect" to disconnect the application, similarly do the same on the JMS tab, then exit IH06

Publish/Subscribe Flow Configuration: MQe

Define MQe Queue Managers

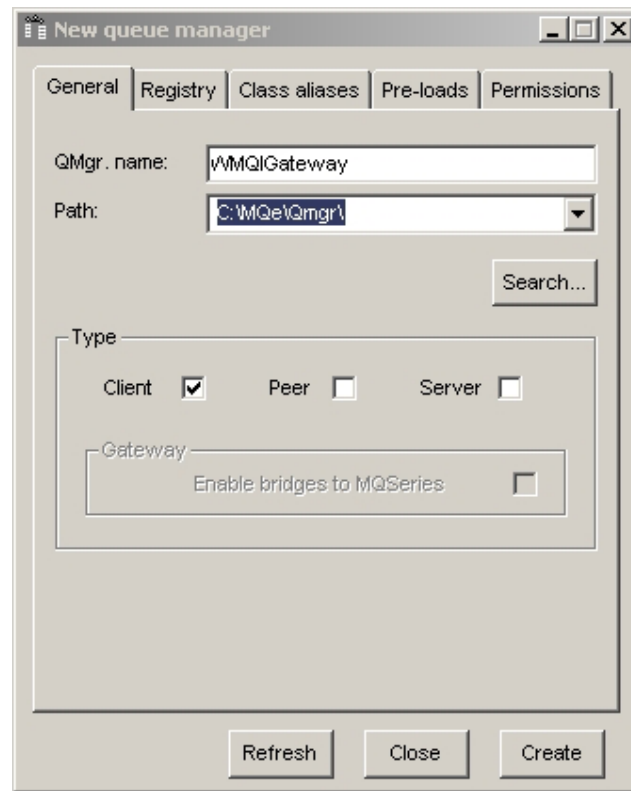
We will use the MQe Explorer to create the configuration shown below.

1. Before using MQe Explorer, the system classpath must be updated to reflect the MQe classes installed under WMQI.
2. Access your system classpath (My Computer ->Properties ->Advanced->Environment Variables)
3. Add the following jar files to your classpath, they can be found in <WMQI Install dir>\classes\
 - a) MQeDevice.jar
 - b) MQeExamples.jar
 - c) MQeGateway.jar
 - d) MQeMQBridge.jar
 - e) mqimqe.jar

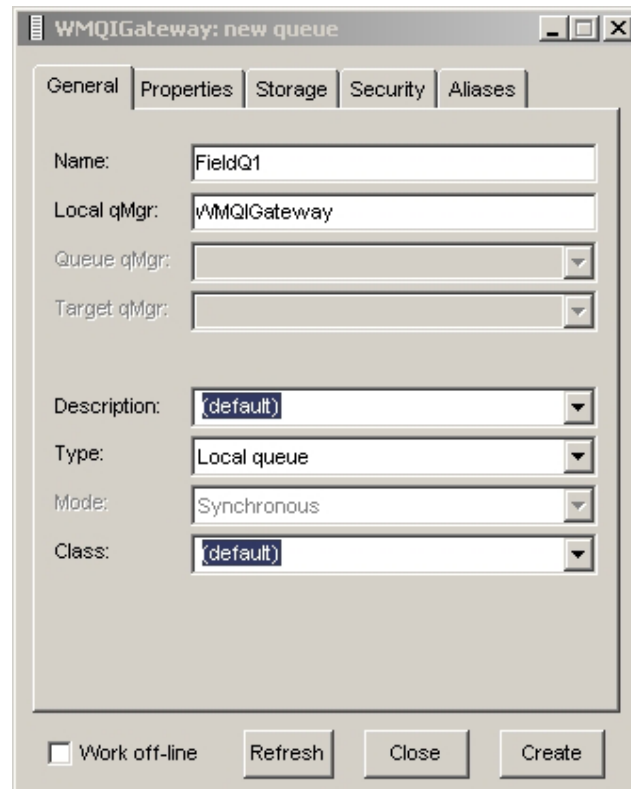
4. Create a directory where MQe Queue Managers will be created and stored e.g. C:\MQe\Qmgr
5. Open a command prompt and navigate to <class directory>\utilities\ES02\, this contains the MQe Explorer; run MQe_Explorer.exe

6. Create the Gateway Queue Manager: WMQIGateway

- a) Use File->New->Queue Manager to launch the “New Queue Manager” Wizard
- b) On the General tab, specify the name of the queue manager “WMQIGateway”
- c) On the General tab, specify the path where the queue manager will be created; “C:\MQe\Qmgr”
- d) On the General tab, set the queue manager type to “Client”. We use “Client” because the Server/Gateway configuration will be done in the MQe Input Node, overriding anything we do in MQe Explorer. Click “Create”. An initialization file will be created and the queue manager started



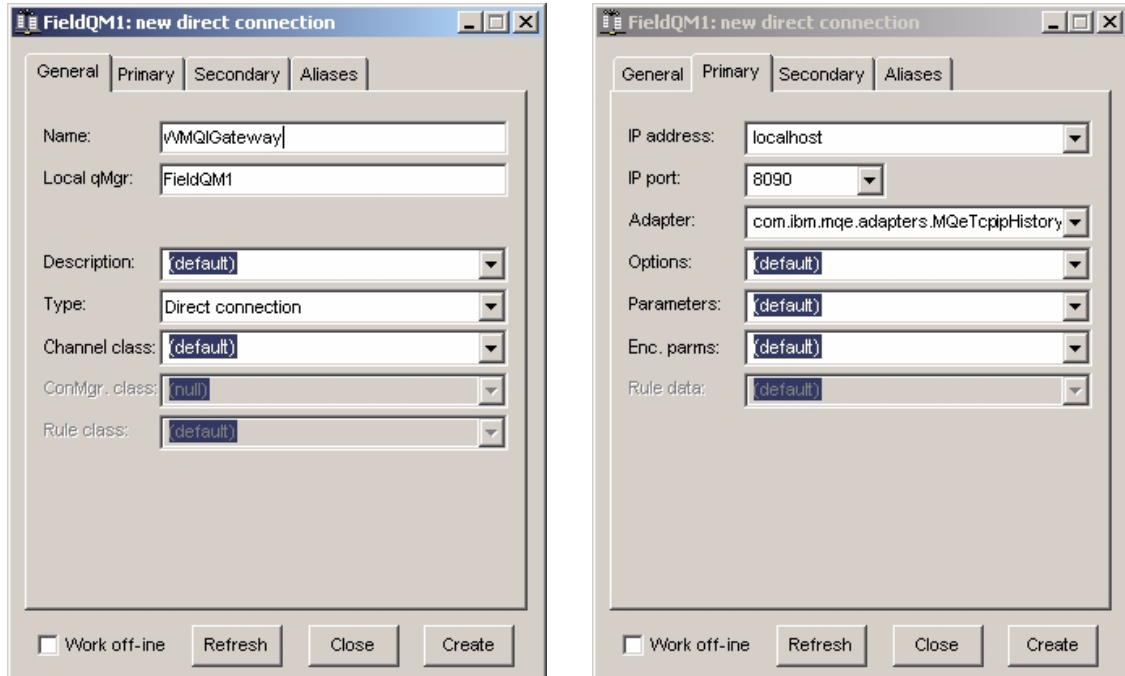
- e) Expand the MQe Root tree to view the local queues defined for WMQIGateway. In order to define the queue that will be used by the Client queue manager to retrieve subscriptions select File->New->Queue
- f) On the general tab specify the queue name “FieldQ1”, accept all other defaults and click “Create”



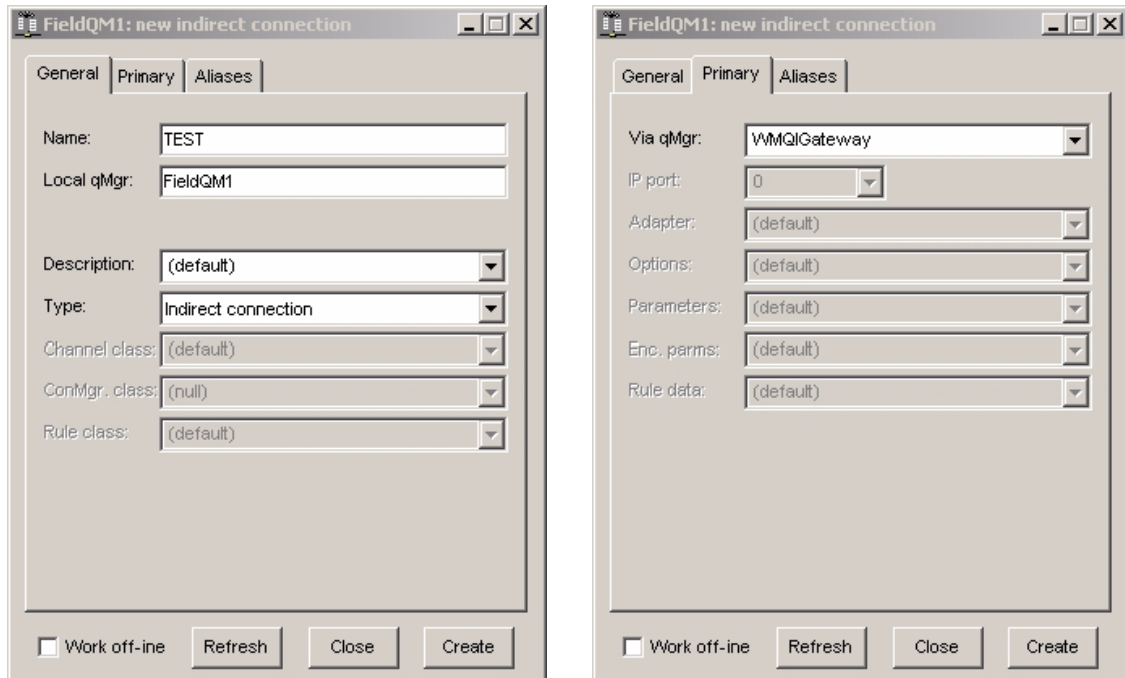
g) Close the "New Queue" Wizard, and close WMQIGateway using File->Close

7. Create the Client Queue Manager: FieldQM1

- As done previously, use File->New->Queue Manager to launch the "New Queue Manager" Wizard
- On the General tab, specify the name of the queue manager "FieldQM1"
- On the General tab, specify the path where the queue manager will be created; "C:\MQe\Qmgr"
- On the General tab, set the queue manager type to "Client". Click "Create". An initialization file will be created and the queue manager started
- Expand the MQe Root tree to view the connections defined for FieldQM1. Use File->New->Connection to launch the "New Connection" wizard
- The first connection will be the direct connection to the Gateway Queue Manager. On the General tab specify the name of the connection "WMQIGateway"
- On the General tab, ensure that the connection type is set to "Direct Connection".
- On the Primary tab, specify an IP Address "localhost" and set the port to 8090
- On the Primary tab, from the Adapter list, select the MqeTcpiHistoryAdapter, accept all other defaults and click "Create"



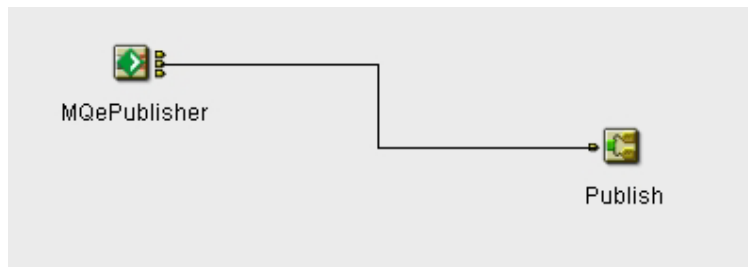
- j) You will receive a warning message “*WMQIGateway – connection refused: couldn’t connect to localhost:8090, Suppress future messages?*” Select **Yes**. The warning arises because WMQIGateway is not yet running.
- k) Create an indirect connection which routes messages via the Gateway Queue Manager, to the MQSeries Queue Manager that hosts the WMQI broker
- l) On the General tab, specify the name of your MQSeries Queue Manager
- m) On the General tab, select a connection type of “Indirect”
- n) On the Primary tab, select “WMQIGateway” as the “via qMgr”, click “Create”



- o) Close the “New Connection” Wizard, and close FieldQM1 using File->Close
- p) Close MQExplorer

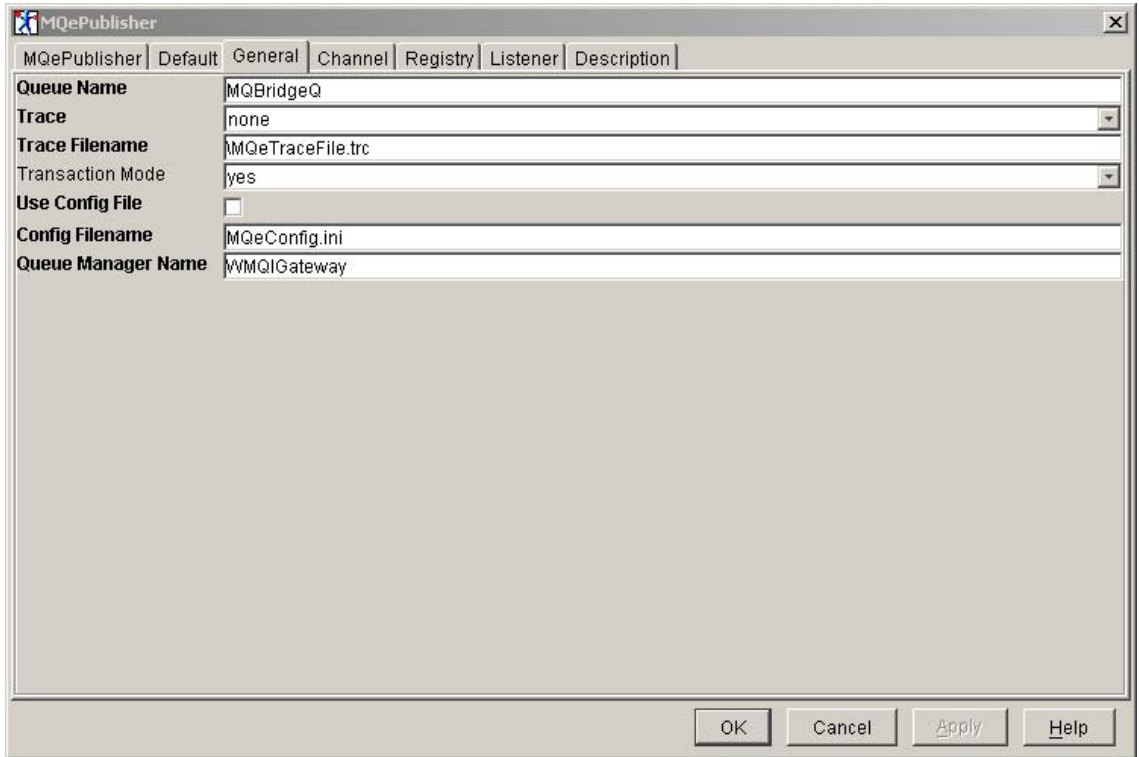
Create Message Flow: MQePublish

While the scenario does not call for the MQe application to publish messages, an MQe message flow is required to support MQe subscribers as well. Open the WMQI Control Center, switch to the Message Flow Tab, and create a new message flow, called “**MQePublish**”. Build the message flow as shown below and set the attributes on the nodes as described. If not explicitly mentioned, you can accept the default values for the node properties.

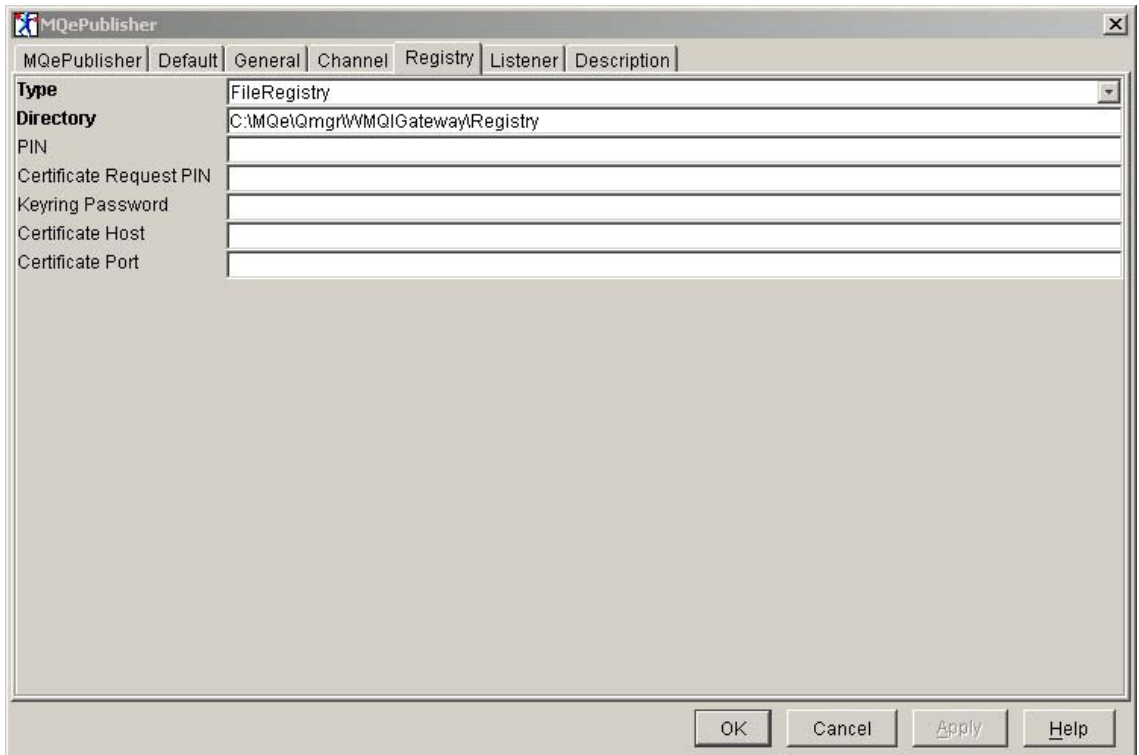


1. MQeInput Node: MQePublisher

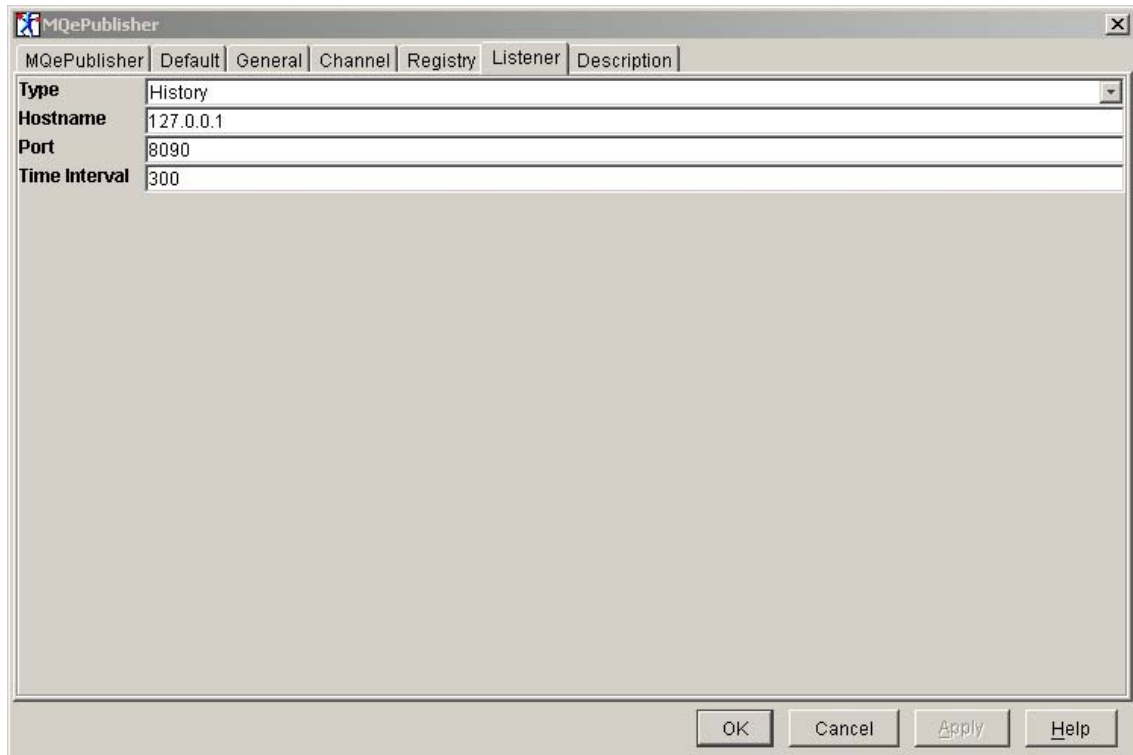
- a) On the General properties tab, specify the Queue Name for the MQ bridge queue; “MQBridgeQ”. The queue will be created automatically by the MQeInput node, when it starts the Gateway Queue Manager
- b) On the General properties tab, specify the Queue Manager Name for the Gateway Queue Manager; “WMQIGateway”.



- c) On the Registry tab, accept the default registry type of "FileRegistry", and specify the registry directory for WMQIGateway; "C:\MQe\Qmgr\WMQIGateway\Registry". This references the directory where MQe Explorer created the queue manager



- d) On the Listener tab, setup the listener to match the direction connection defined for FieldQM1. Specifically select "History" as the adapter type and specify the port "8090"

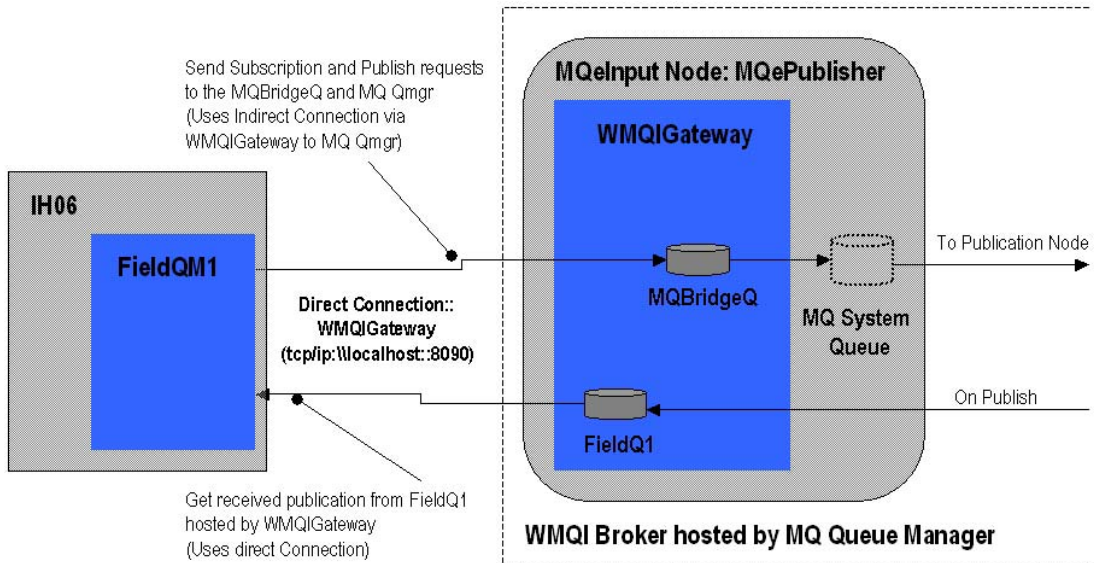


2. Publication Node: Publish

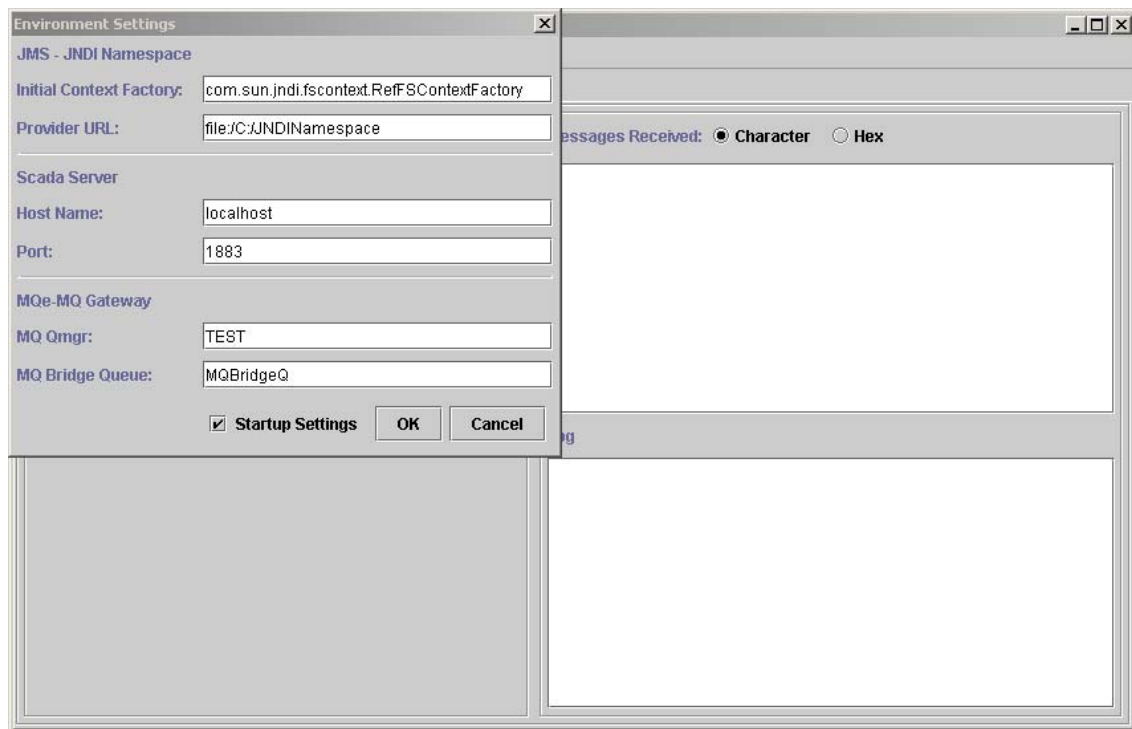
- a) Wire the "out" terminal from MQePublisher to the "in" terminal of Publish
3. Check in MQePublish and switch to the Assignments tab. Check out the execution group and assign MQePublish, check in and deploy. Check Log tab and confirm successful deployment

Execute Message Flow:

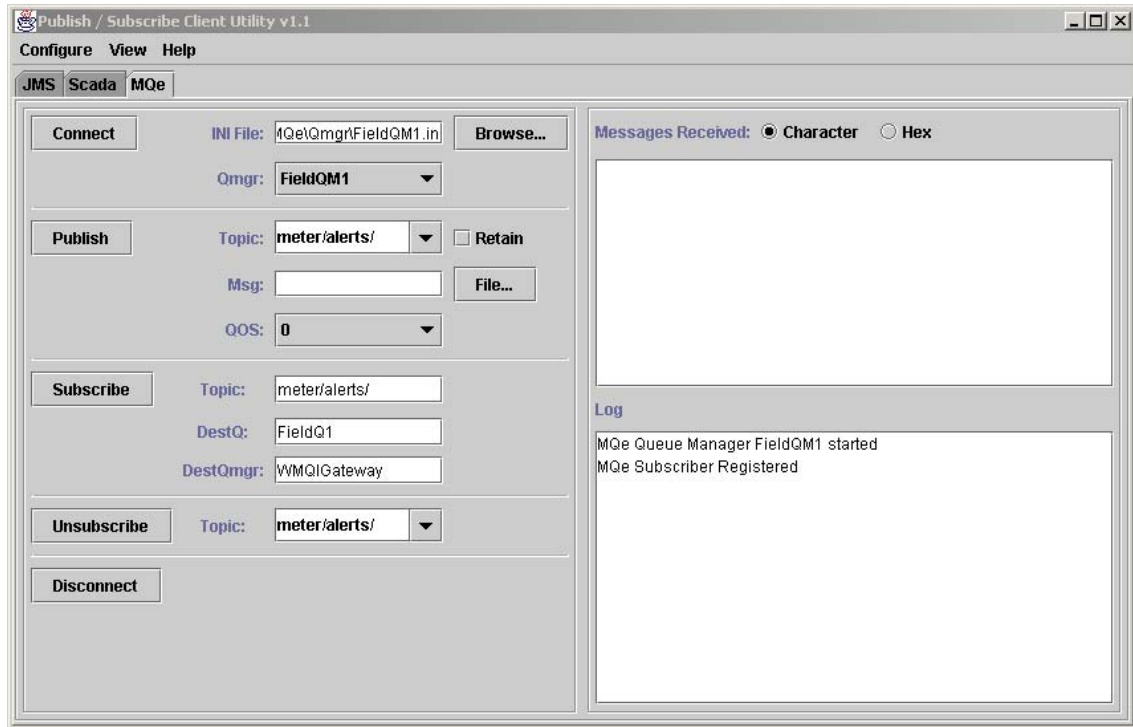
The exercise uses supportpac IH06 to simulate SCADA Publishers and MQe subscribers. Based on the definitions defined above, the MQe Queue Managers interact as shown below:



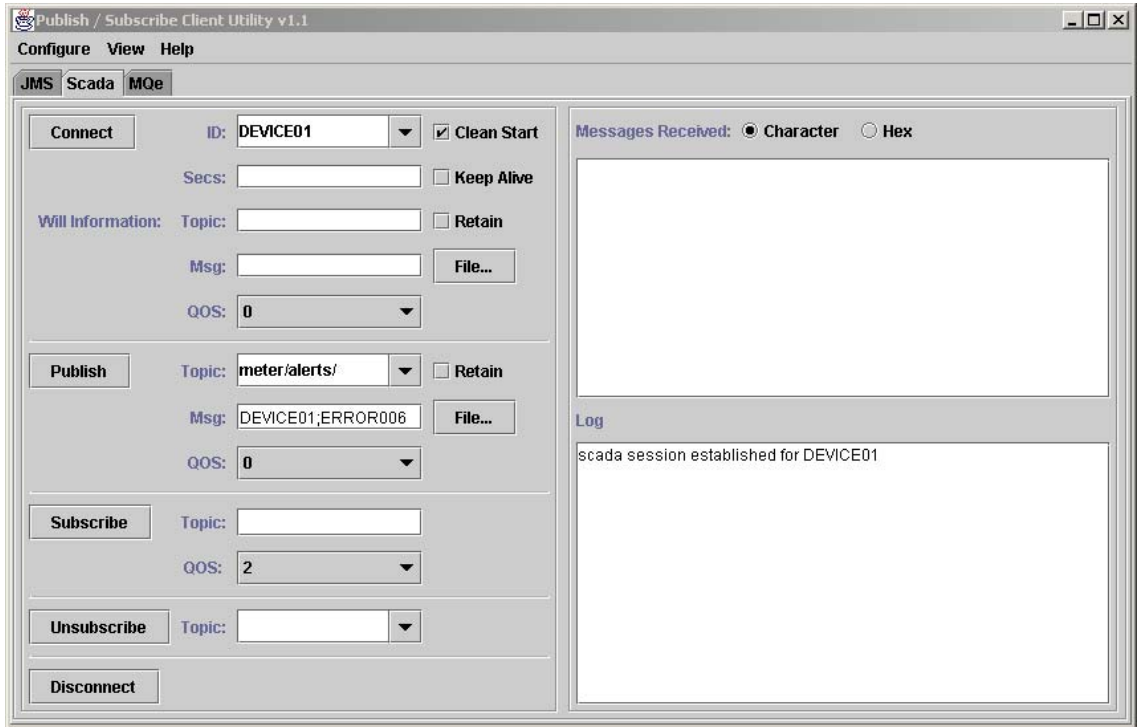
1. Run IH06.cmd
2. Update the supplied MQe environment configuration to match your environment (**Configure-> Environment...**). The "MQ Qmgr" attribute should match the name of the queue manager hosting WMQI, and should correspond to the name of the indirect connection specified for FieldQM1. The "MQ Bridge Queue" attribute should match the queue name defined in the MQeInput Node defined in MQePublish; "MQBridgeQ"



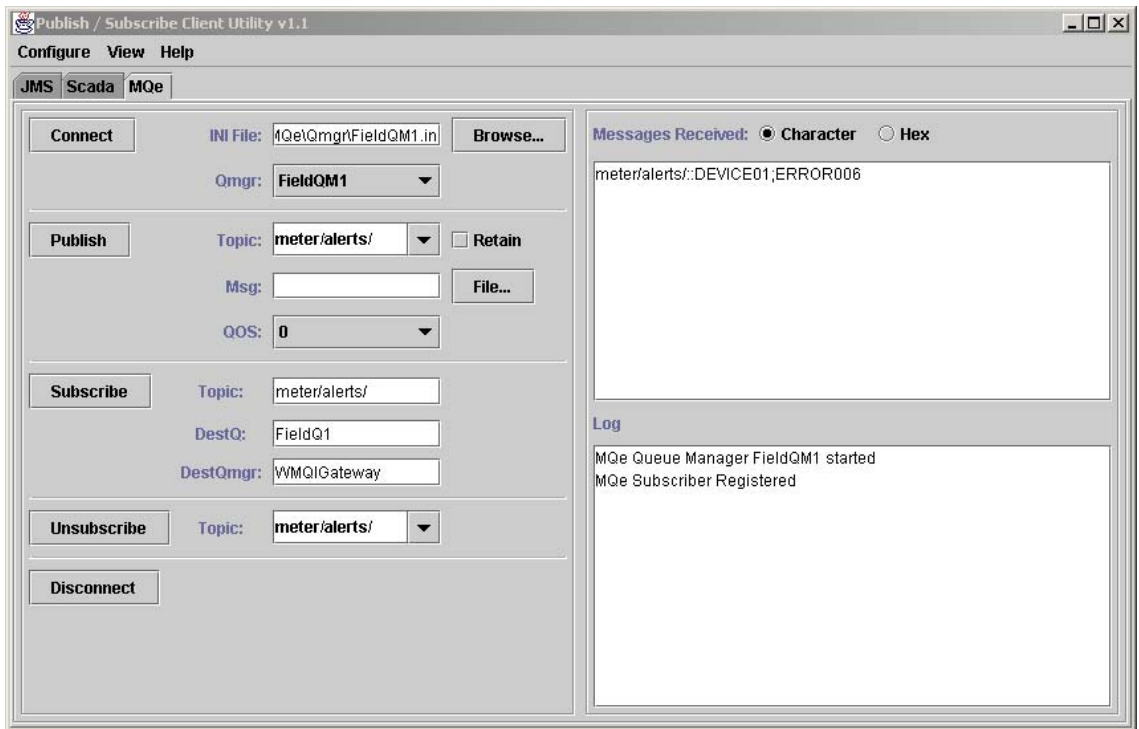
3. Switch to the MQe Tab to configure the MQe Subscriber; In the Connect panel browse for the .ini file defined for FieldQM1 by MQe Explorer, the file should be located in C:\MQe\Qmgr\. Click "Connect"
4. In the Subscriber panel, supply the topic string "meter/alerts/", the DestQ, and DestQmgr attributes specify where subscriptions are to be received. Specify the values **FieldQ1** and **WMQIGateway** respectively. Click "Subscribe".



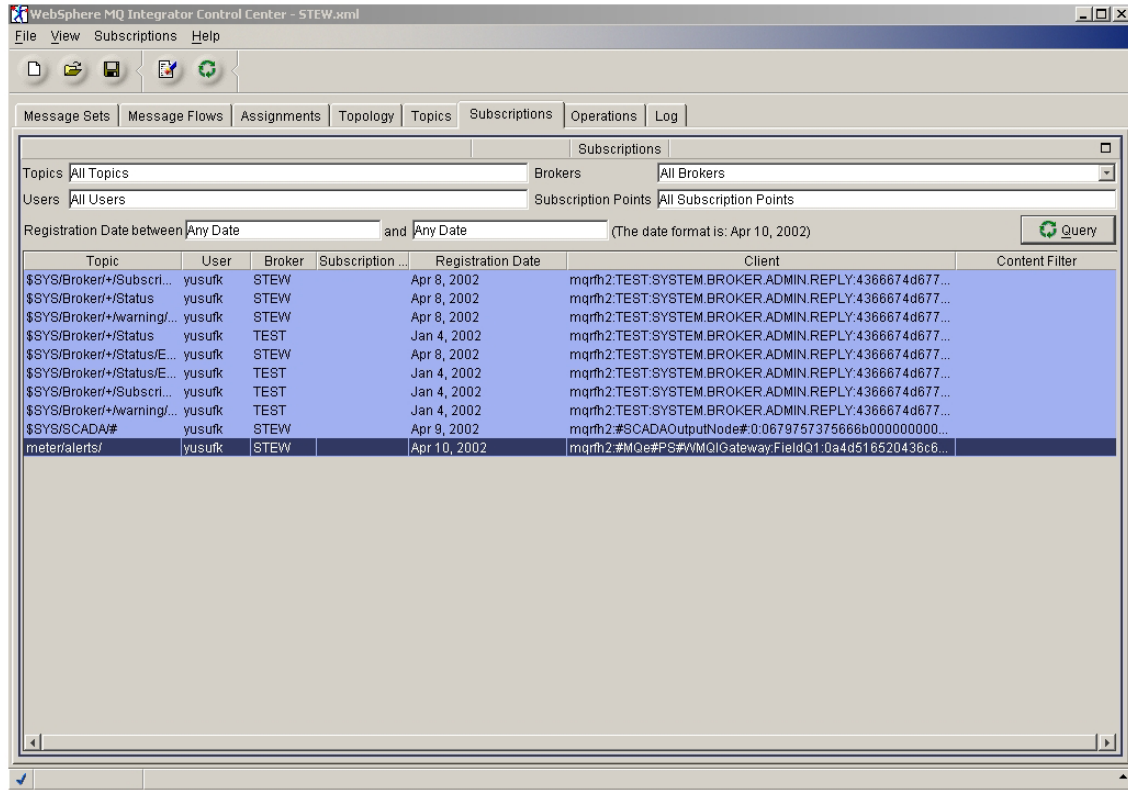
5. Switch to the SCADA Tab to configure the SCADA Publisher; in the connect panel supply a client ID (**DEVICE01**), you can accept all the other defaults. Click "Connect"
6. In the Publisher panel, supply the topic "meter/alerts/", matching the topic to which the MQe subscriber has subscribed
7. Type in a message e.g. "**DEVICE01;ERROR006;**" (you may make up your own message if you wish. Click "Publish"



8. Switch to the MQe Tab; Observe, that the subscriber has received the publication



9. You have successfully configured WMQI to support publish/subscribe using SCADA and MQe transports. You can further explore WMQI functionality reversing the role of publisher and subscriber over single or multiple transports.
10. Inspection of the Subscriptions tab in the WMQI Control Center shows the registered MQe Subscription.



11. On the MQe tab click “Unsubscribe”, then “Disconnect” to disconnect the application, similarly do the same on the SCADA tab, then exit IH06

End of Document