JMS Admin for the WMQ guy

Simon Gormley (sgormley@uk.ibm.com)
IBM Hursley Park

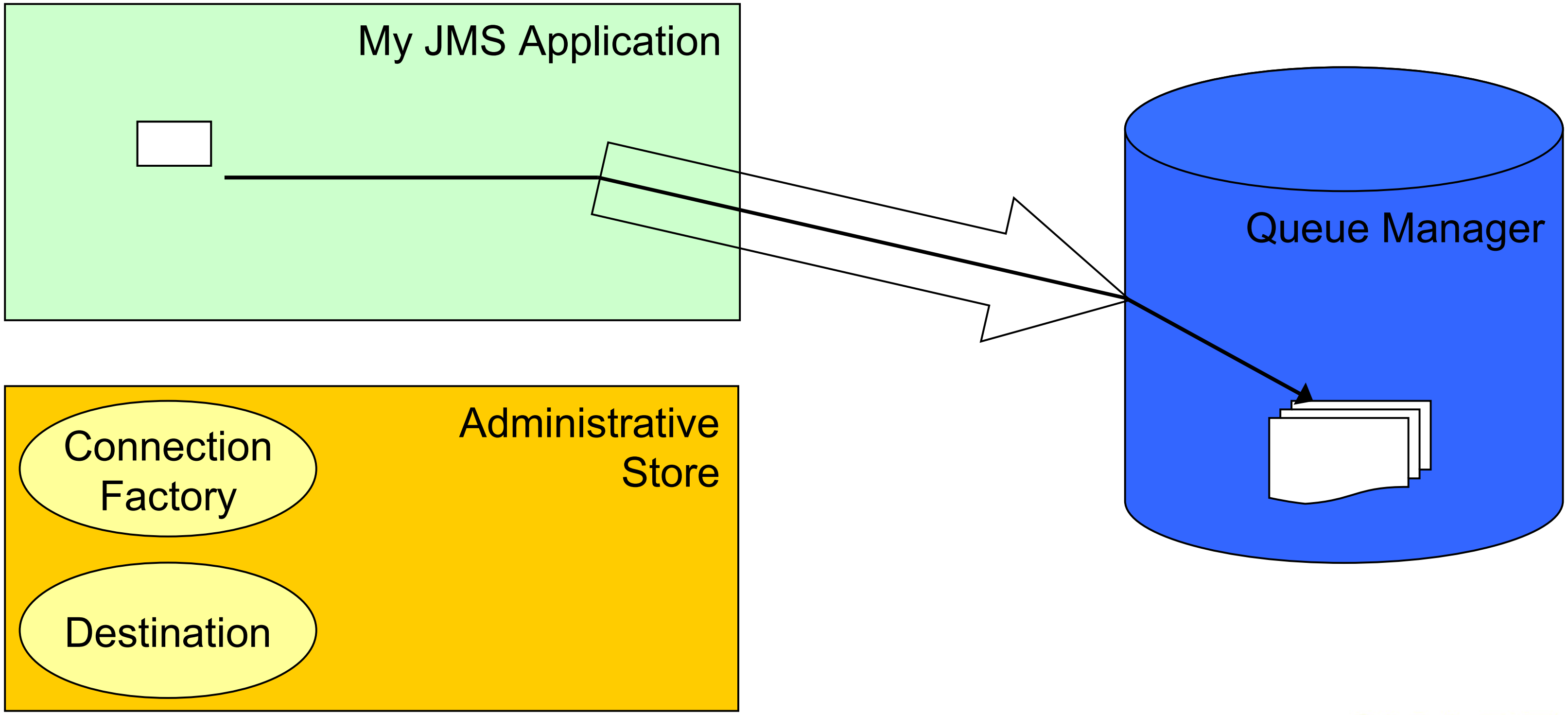
March 2nd, 2011
Session 8715

Agenda

- **JMS Administered Objects**
  - **Connection Factories**
  - Destinations
- WMQ JMS Administration Tools
  - JNDI Repositories
  - JMSAdmin
  - MQ Explorer
- Connection Factory Properties
- Destination Properties

JMS Administered Objects

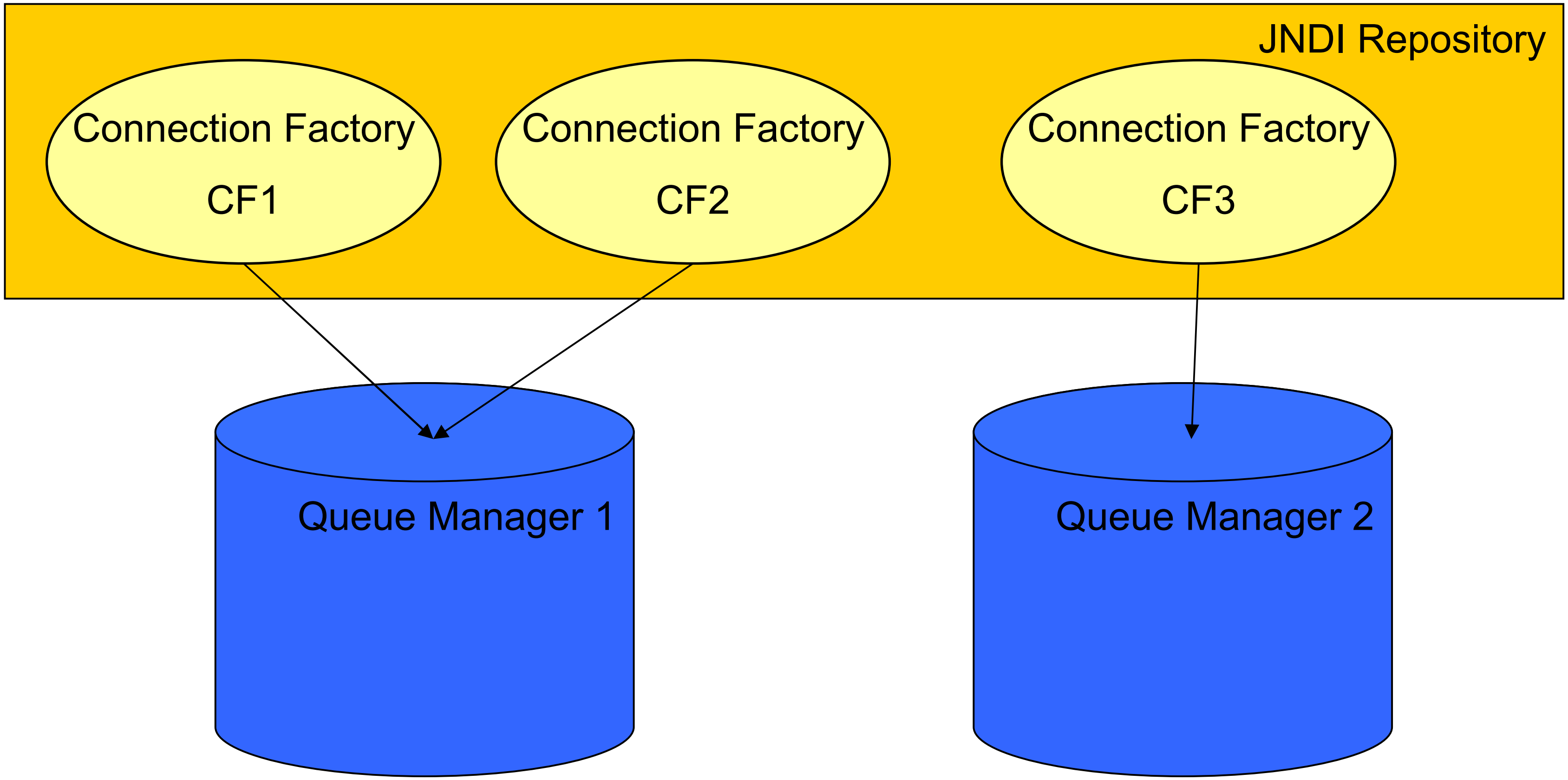
- What do administered objects do?



JMS Administered Objects: Connection Factories

- Contains information about how to connect to a queue manager.
- Used by JMS applications to create connections.
- WMQ supports three types:
  - Queue Connection Factories
    - Used by applications who want to use point-to-point messaging.
  - Topic Connection Factories
    - Used by applications to create connections to queue managers hosting a pub/sub engine.
  - Connection Factories
    - Can be used by either point-to-point or publish/subscribe applications to connect to a queue manager.
- Stored in a JNDI Repository.

JMS Administered Objects: Connection Factories



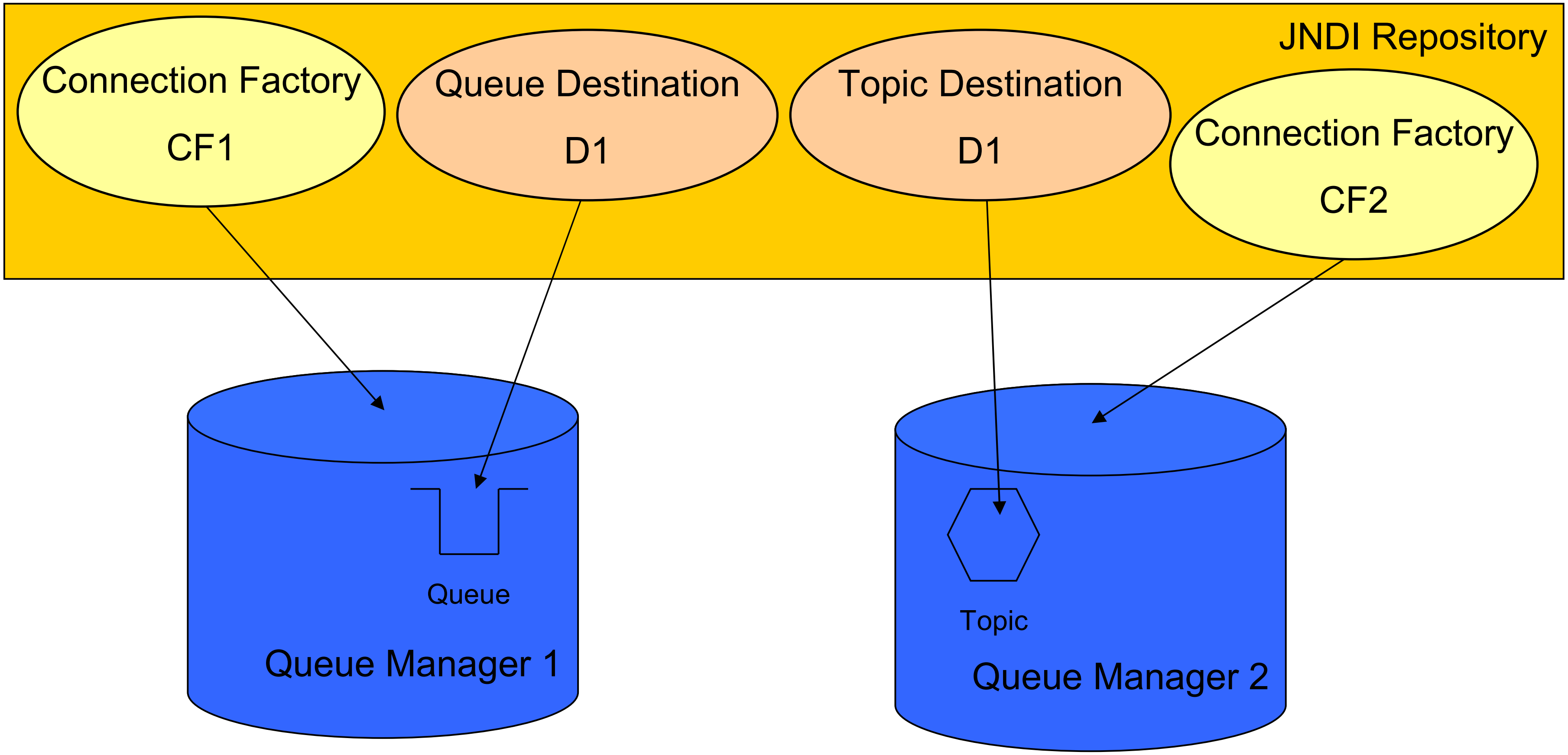
Agenda

- **JMS Administered Objects**
  - Connection Factories
  - **Destinations**
- WMQ JMS Administration Tools
  - JNDI Repositories
  - JMSAdmin
  - MQ Explorer
- Connection Factory Properties
- Queue Destination Properties
- Topic Destination Properties

JMS Administered Objects: Destinations

- The destination of messages sent by producers.
- The source of messages received by consumers.
- WMQ supports two types:
  - Queue Destinations
    - Used by applications to open queues.
  - Topic Destinations
    - Used by applications to publish to, or subscribe on, a topic.
- Destination definitions are stored in a JNDI Repository.

JMS Administered Objects: Destinations



Agenda

- JMS Administered Objects
  - Connection Factories.
  - Destinations.
- **WMQ JMS Administration Tools**
  - **JNDI Repositories**
  - JMSAdmin
  - MQ Explorer
- Connection Factory Properties
- Destination Properties

WMQ JMS Administration Tools: JNDI Repositories

- The WMQ JMS Administration Tools need access to a JNDI Repository to store connection factory and destination definitions.
- Two main types of JNDI Repository are supported.
  - LDAP Directories
    - Heavyweight.
    - Offers high levels of security.
    - Easy to share connection factory and destination objects between multiple JMS clients running on different machines.
    - Can be difficult to set up.
  - File system
    - Very lightweight.
    - Easy to set up and get running.
    - Not very secure.
    - Hard to share repository between multiple JMS clients.

WMQ JMS Administration Tools: JNDI Repositories

- The WMQ JMS Administration Tools can be configured to use other Repositories if required.

- WebSphere Application Server provides it's own Repository and tools for creating the WMQ JMS administered objects.
  - The WMQ JMS Administration Tools should not be used with this application server.

- WebSphere MQ SupportPac ME01 allows the queue manager to be used as a JNDI store.

Agenda

- JMS Administered Objects
  - Connection Factories.
  - Destinations.
- **WMQ JMS Administration Tools**
  - JNDI Repositories
  - **JMSAdmin**
  - MQ Explorer
- Connection Factory Properties
- Destination Properties

WMQ JMS Administration Tools: JMSAdmin

- Command line tool.
- Supported on all platforms.
- Installed into <WMQ_HOME>\java\bin.
- To run it:
  - Edit JMSAdmin.config file to point to the JNDI repository that will be used, and optionally pass in authentication credentials.
  - Then enter JMSAdmin
- Can be used to run scripts
  - For example “jmsadmin < myscript.txt”

WMQ JMS Administration Tools: JMSAdmin

- JMSAdmin.config contains three properties
  - INITIAL_CONTEXT_FACTORY
    - The class used by the JNDI repository to store and retrieve objects.
    - Possible values are:

      com.sun.jndi.ldap.LdapCtxFactory
      Used for LDAP Repositories on distributed platforms.
      com.ibm.jndi.LDAPCtxFactory
      Used to connect to an LDAP repository from JMS applications running on z/OS.
      com.sun.jndi.fscontext.RefFSContextFactory
      Used for file system Repositories

    - JMSAdmin.config contains an entries for com.ibm.ejs.ns.jndi.CNInitialContextFactory and com.ibm.websphere.naming.WsnInitialContextFactory. These should never be used!

WMQ JMS Administration Tools: JMSAdmin

- PROVIDER_URL
  - An address used by the JMSAdmin tool to access the JNDI Repository.
  - Possible values are:

    ldap://<hostname>/<contextname>

    The hostname and port that the LDAP server is listening on, followed by the top level directory context where the objects will be stored.

    file:<drive>/<pathname>

    The path to the directory where the administered object definitions will be saved. The directory must exist before JMSAdmin can be run.

  - JMSAdmin.config contains an entry for iiop://localhost/. It should never be used!

WMQ JMS Administration Tools: JMSAdmin

- SECURITY_AUTHENTICATION
  - Used when connecting to a secure LDAP JNDI Repository.
  - Possible values are:

    none

    The JMSAdmin tool does not pass any security information to LDAP.

    simple

    The JMSAdmin tool sends an LDAP distinguished name and password to the Server for authentication during startup.

    CRAM-MD5

    The JMSAdmin tool flows an LDAP distinguished name and password (encrypted as an MD5 hash) when connecting to the LDAP server.

WMQ JMS Administration Tools: JMSAdmin

- SECURITY_AUTHENTICATION
  - The distinguished name and password can be stored in JMSAdmin.config, using the PROVIDER_USERDN and PROVIDER_PASSWORD properties.
  - If they are not specified in the file, JMSAdmin will request them before connecting to the Respository.

WMQ JMS Administration Tools: JMSAdmin

- JMSAdmin expects commands to be in the format
  - Verb noun(value) noun(value)…..
- Useful verbs are:
  - DEFINE
  - ALTER
  - DISPLAY
- Nouns include:
  - CF - JMS Connection Factory
  - QCF - JMS Queue Connection Factory
  - TCF - JMS Topic Connection Factory
  - Q - JMS Queue
  - T - JMS Topic
  - QMGR - Queue Manager Name
  - QU - Queue Name
  - TO - Topic Name

WMQ JMS Administration Tools: JMSAdmin

- To create a Queue Connection Factory for the queue manager QM1, enter the command:
  - DEFINE QCF(testQCF) QMGR (QM1)
    - The Queue Connection Factory will be stored in the JNDI Repository with the name testQCF.

- To create a JMS Queue Destination that points to the queue “test”, enter:
  - DEFINE Q(testQ) QU(test)
    - The object will be stored in JNDI with the name testQ.

Agenda

- JMS Administered Objects
  - Connection Factories.
  - Destinations.
- WMQ JMS Administration Tools
  - JNDI Repositories
  - JMSAdmin
  - **MQ Explorer**
- Connection Factory Properties
- Destination Properties

WMQ JMS Administration Tools: MQ Explorer

- To create JMS Administered Objects in MQ Explorer:
  - Set up a context
    - Directory-like structure where the objects will be stored.
  - Select the type of JNDI repository that will be used
    - File system
    - LDAP
    - Another JNDI repository
  - Specify the address of the JNDI repository
    - For file system contexts, select the directory where the Administered Objects will be stored
    - For LDAP repositories, enter the URL of the LDAP server
  - Optionally enter the username and password used to connect to the JNDI repository
  - And that's it!

WMQ JMS Administration Tools: MQ Explorer

- MQ Explorer wizards provide step-by-step instructions for creating JMS Administered Objects.

- MQ Explorer also allows Queue or Topic Destination Administered Objects at the same time as creating the actual Queue or Topic.
- MQ Queues and Topics can also be created from Destination Administered Objects.

Agenda

- JMS Administered Objects
  - Connection Factories.
  - Destinations.
- WMQ JMS Administration Tools
  - JNDI Repositories
  - JMSAdmin
  - MQ Explorer
- **Connection Factory Properties**
- Destination Properties

Connection Factory Properties: Queue Manager Name

- The name of the queue manager that this Factory points to.
  - JMS applications will use the factory to create a connection to this queue manager.
- JMSAdmin Property Name: QMANAGER / QMGR

Connection Factory Properties: Transport Type

- The transport mechanism supported by this factory.
  - JMS applications that use this Factory will attempt to connect to the queue manager using the transport mechanism specified here.
- Possible values:
  - BINDINGS
    - JMS applications using this Factory will use inter-process communication (IPC) to connect to the queue manager.
  - CLIENT
    - JMS applications will attempt to connect to the queue manager using TCP/IP.
  - DIRECT
  - DIRECT_HTTP
    - With these options, JMS applications will attempt to create direct network connections to IBM Message Broker.

- JMSAdmin Property Name : TRANSPORT / TRANS

Connection Factory Properties: Provider Version

- The version of WMQ that the queue manager this Factory is pointing to is running on.
  - The client is optimized to connect to Version 7 queue managers, and makes use of Version 7 functionality, such as shared conversations.
  - The WMQ JMS client can be used to connect to Version 5.1 queue managers and above.
    - This uses a slightly modified version of the WMQ Version 6 JMS Client, which is embedded in the Version 7 code.

Connection Factory Properties: Provider Version

- Possible Values
  - Unspecified
    - The JMS client will first try to connect to the queue manager as if it is Version 7
    - If this fails, it will fall back to the Version 6 (and earlier) behaviour.
  - 6
    - The JMS client will use the embedded Version 6 JMS Client to connect to the queue manager.
  - 7
    - JMS applications that use a Factory with Provider Version set to 7 will use the optimized connection code.
- JMSAdmin Property Name: PROVIDERVERSION / PVER

Connection Factory Properties: Connection Name List

- This property specifies a list of hostnames and ports to attempt to connect to.
  - Comma-separated list of “hostname(port)” entries
  - Similar to a CCDT with multiple entries.
  - Can be used with client reconnection options and client reconnection timeout to allow automatic reconnection to a standby queue manager.
- JMSAdmin Property Name: CONNECTIONNAMELIST / CNLIST

Connection Pooling Properties: Shared Conversation Allowed

- This property specifies whether JMS applications that use this Factory can share their connection to a Version 7 queue manager.
  - For example, imagine the Connection Factory is set up to allow shared conversations.
  - If two applications running on the same Java Virtual Machine use this Factory, they will share the same connection to the queue manager.
    - The queue manager would only show one open connection, even if both applications are running.
- Useful to reduce the number of network connections to a queue manager.
- Can have slight performance impact.
  - Multiple JMS applications will be sending data to a queue manager and waiting for a response over the same channel.

Connection Pooling Properties: Shared Conversation Allowed

- Possible Values
  - Yes
    - JMS applications running in the same Java Virtual Machine can use this Factory and share their connections to the queue manager. The amount of sharing is controlled by the Server Connection Channel
  - No
    - Every JMS application that uses this Factory will create a new connection to the queue manager.
- JMSAdmin Property Name: SHARECONVALLOWED / SCALD

Agenda

- JMS Administered Objects
  - Connection Factories.
  - Destinations.
- WMQ JMS Administration Tools
  - JNDI Repositories
  - JMSAdmin
  - MQ Explorer
- Connection Factory Properties
- **Destination Properties**

Destination Properties: Queue Name / Topic Name

- The name of the Queue or Topic that this Destination maps to.
  - JMS applications that send messages using this Destination will either put messages to the specified queue, or publish messages on this topic.
  - JMS applications that receive incoming messages using this Destination will either get messages from the queue, or subscribe to the topic.

- JMSAdmin Property Name: QUEUE / Q
  TOPIC / T

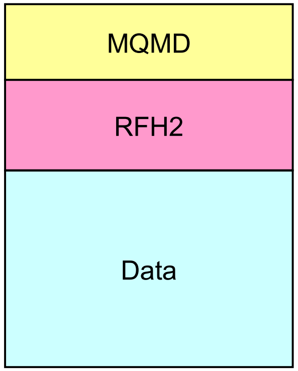
Destination Properties: MQMD Read/Write enabled

- Determines if MQMD properties can be set via JMS get/set message property methods.
  - Allows full access to the MQMD header values
  - Useful for sending or receiving messages from MQ applications that use specific header properties.
  - JMS message property names begin with "JMS_IBM_MQMD…"

- JMSAdmin Property Name: MDREAD / MDR & MDWRITE / MDW

Destination Properties: Target Client

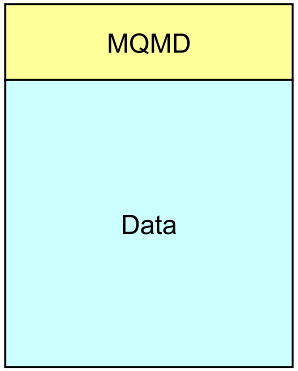
- This property tells the JMS application whether messages sent to this destination are for other JMS applications, or for non-JMS WMQ applications.
  - WMQ JMS messages, by default, have an RFH2 header containing JMS specific information.
  - If a message is for a non-JMS application, this header may cause problems, so can be turned off.


MQMD

RFH2

Data

JMS Messages


MQMD

Data

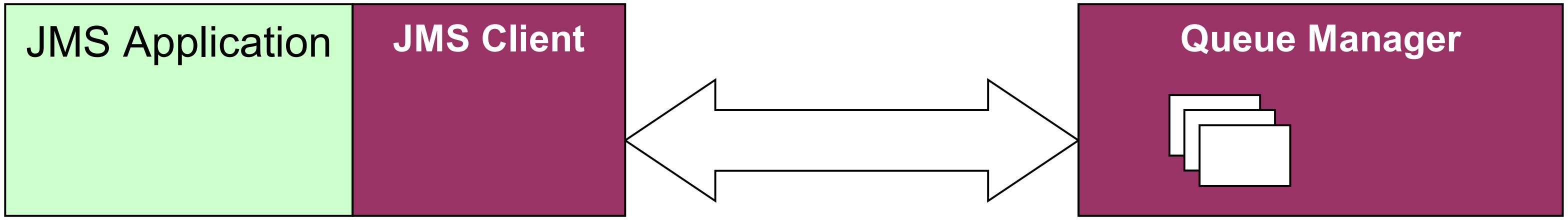
Non-JMS Messages

Destination Properties: Target Client

- Possible Values:
  - JMS
    - JMS messages sent by an application to this Destination will be consumed by another JMS application.
    - The RFH2 header will be included on these messages.
  - MQ
    - The JMS messages sent to this destination are for a non-JMS application.
    - The message will include an MQMD, and the message body.
    - The RFH2 header will not be included.
    - Any information in the RFH2 will be lost.
- JMSAdmin Property Name: TARGCLIENT / TC

Destination Properties: Read Ahead Allowed

- In general, messages are sent to JMS applications one at a time.
- The Read Ahead Allowed property tells the queue manager whether non-persistent messages can be streamed to the client application in preparation for them being consumed.
  - Messages are stored in a buffer on the client.
  - If the client application terminates unexpectedly, all unconsumed non-persistent messages are discarded.



- JMSAdmin Property Name: READAHEADALLOWED / RAALD

Agenda

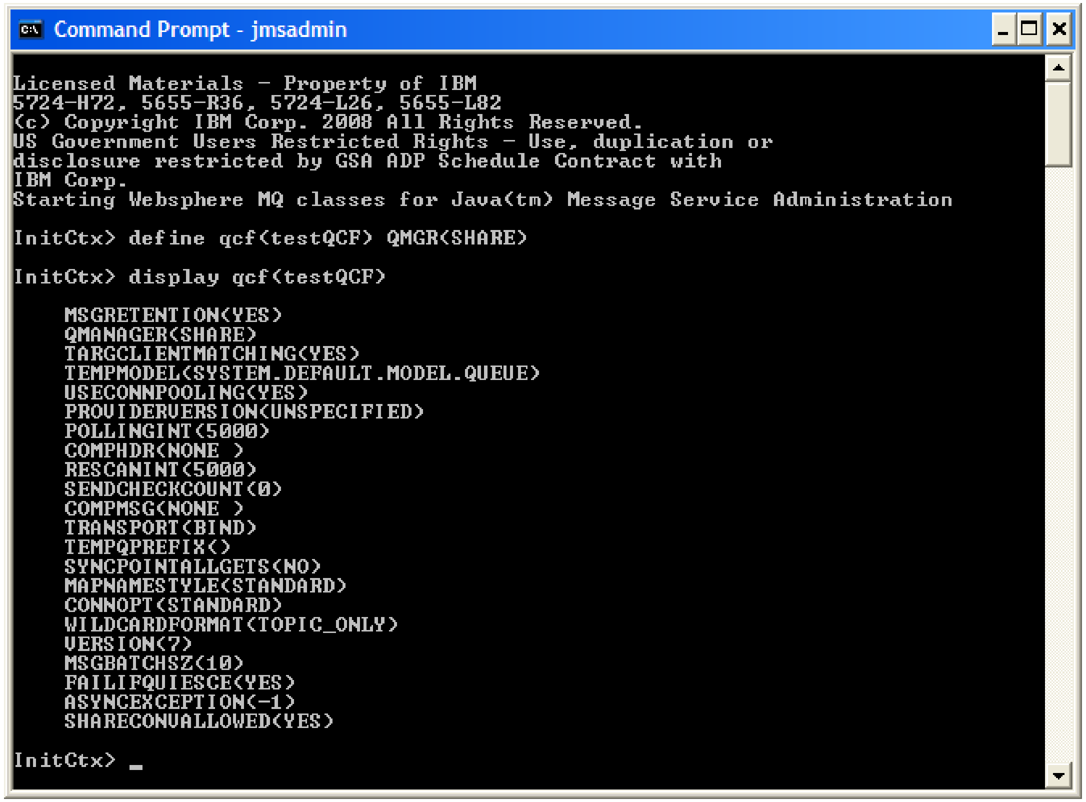
- JMS Administered Objects
  - Connection Factories.
  - Destinations.
- WMQ JMS Administration Tools
  - JNDI Repositories
  - JMSAdmin
  - MQ Explorer
- Connection Factory Properties
- Destination Properties

Any questions?

- If you have any questions, or ideas for future topics, feel free to email me at **sgormley@uk.ibm.com**

JMSAdmin Demo Screenshots

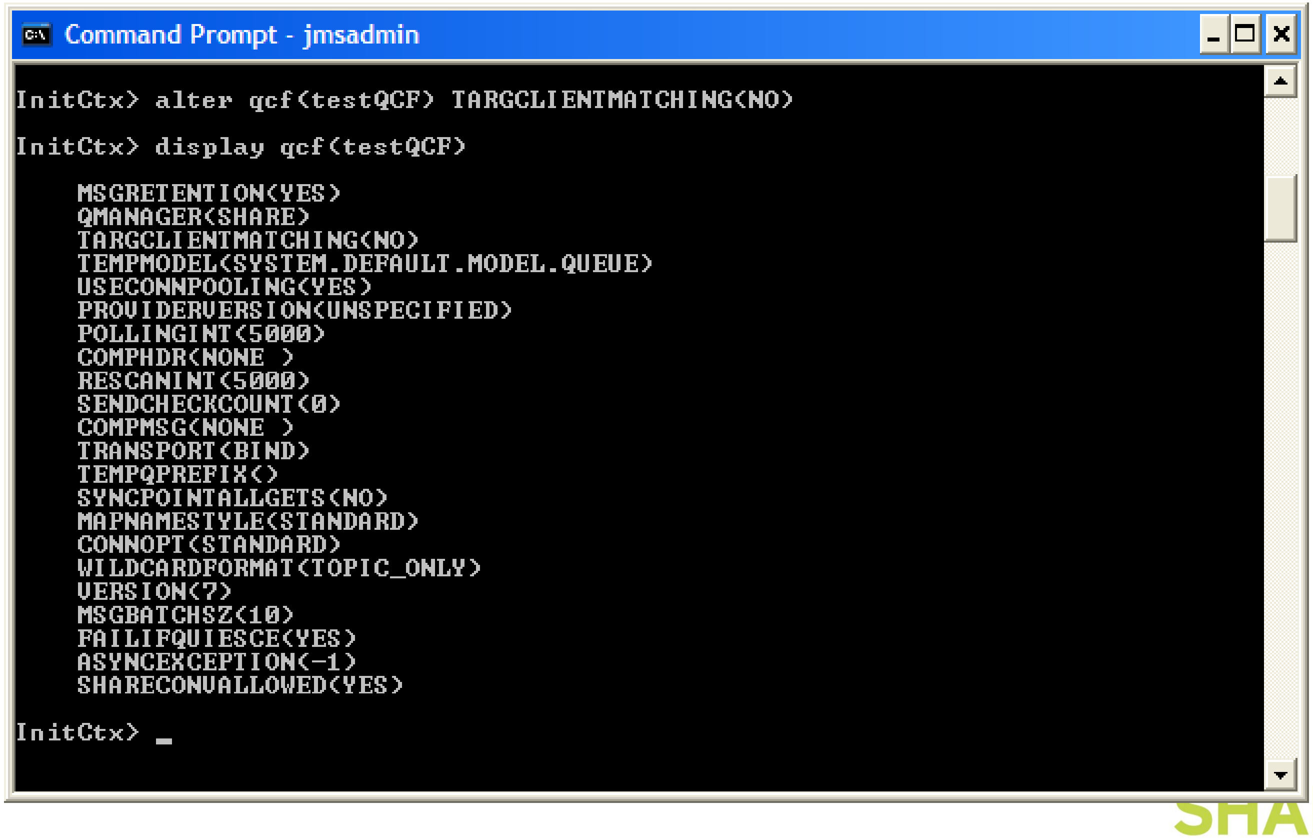
- After setting JMSAdmin.config, and running JMSAdmin, commands can be entered, such as `define qcf`



```
Licensed Materials - Property of IBM
5724-H72, 5655-R36, 5724-L26, 5655-L82
(c) Copyright IBM Corp. 2008 All Rights Reserved.
US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
Starting Websphere MQ classes for Java(tm) Message Service Administration

InitCtx> define qcf(testQCF) QMGR(SHARE)

InitCtx> display qcf(testQCF)

    MSGRETENTION(YES)
    QMANAGER(SHARE)
    TARGCLIENTMATCHING(YES)
    TEMPMODEL(SYSTEM.DEFAULT.MODEL.QUEUE)
    USECONNPOOLING(YES)
    PROVIDERVERSION(UNSPECIFIED)
    POLLINGINT(5000)
    COMPHDR(NONE )
    RESCANINT(5000)
    SENDCHECKCOUNT(0)
    COMPMSG(NONE )
    TRANSPORT(BIND)
    TEMPQPREFIX()
    SYNCPOINTALLGETS(NO)
    MAPNAMESTYLE(STANDARD)
    CONNOPT(STANDARD)
    WILDCARDFORMAT(TOPIC_ONLY)
    VERSION(7)
    MSGBATCHSZ(10)
    FAILIFQUIESCE(YES)
    ASYNCEXCEPTION(-1)
    SHARECONVALLOWED(YES)

InitCtx> _
```

JMSAdmin Demo Screenshots

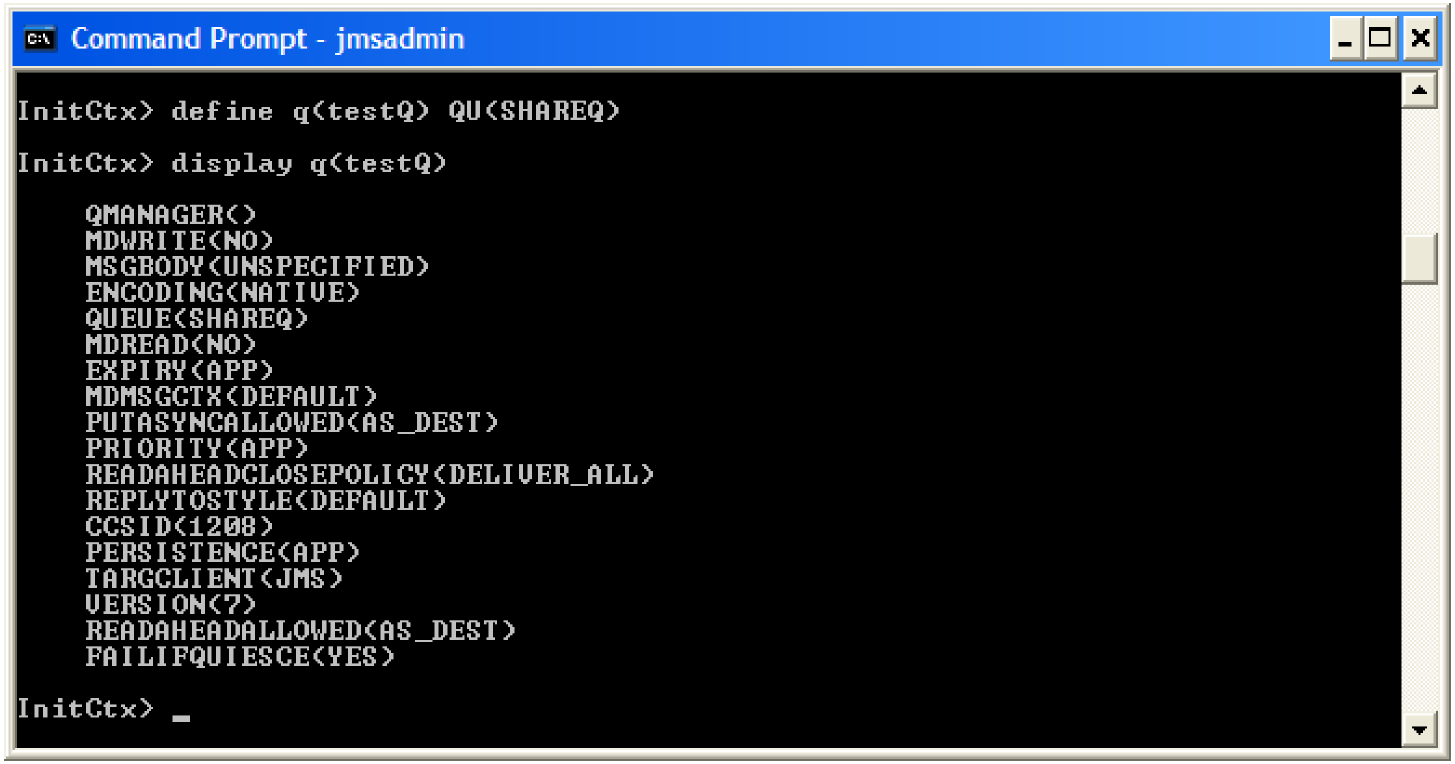
- Use the `alter qcf` command to make modifications after the object has been defined



```
InitCtx> alter qcf(testQCF) TARGCLIENTMATCHING(NO)

InitCtx> display qcf(testQCF)

    MSGRETENTION(YES)
    QMANAGER(SHARE)
    TARGCLIENTMATCHING(NO)
    TEMPMODEL(SYSTEM.DEFAULT.MODEL.QUEUE)
    USECONNPOOLING(YES)
    PROVIDERVERSION(UNSPECIFIED)
    POLLINGINT(5000)
    COMPHDR(NONE )
    RESCANINT(5000)
    SENDCHECKCOUNT(0)
    COMPMSG(NONE )
    TRANSPORT(BIND)
    TEMPQPREFIX()
    SYNCPOINTALLGETS(NO)
    MAPNAMESTYLE(STANDARD)
    CONNOPT(STANDARD)
    WILDCARDFORMAT(TOPIC_ONLY)
    VERSION(7)
    MSGBATCHSZ(10)
    FAILIFQUIESCE(YES)
    ASYNCEXCEPTION(-1)
    SHARECONVALLOWED(YES)

InitCtx> _
```

JMSAdmin Demo Screenshots

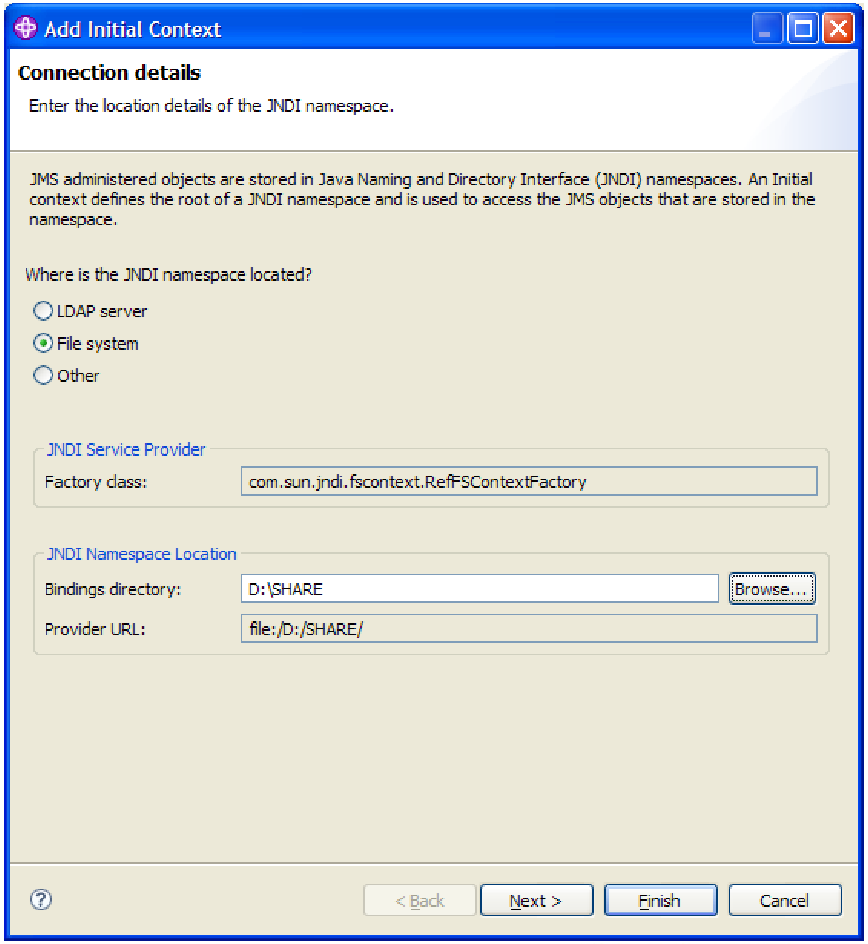
- Queue destination objects created by the `define q` command



```
InitCtx> define q(testQ) QU(SHAREQ)

InitCtx> display q(testQ)

    QMANAGER()
    MDWRITE(NO)
    MSGBODY(UNSPECIFIED)
    ENCODING(NATIVE)
    QUEUE(SHAREQ)
    MDREAD(NO)
    EXPIRY(APP)
    MDMSGCTX(DEFAULT)
    PUTASYNCALLOWED(AS_DEST)
    PRIORITY(APP)
    READAHEADCLOSEPOLICY(DELIVER_ALL)
    REPLYTOSTYLE(DEFAULT)
    CCSID(1208)
    PERSISTENCE(APP)
    TARGCLIENT(JMS)
    VERSION(7)
    READAHEADALLOWED(AS_DEST)
    FAILIFQUIESCE(YES)

InitCtx> _
```

MQ Explorer Demo Screenshots

- Create a connection to a File-based JNDI repository. Only the directory needs to exist. Click Finish to create.



Connection details

Enter the location details of the JNDI namespace.

JMS administered objects are stored in Java Naming and Directory Interface (JNDI) namespaces. An Initial context defines the root of a JNDI namespace and is used to access the JMS objects that are stored in the namespace.

Where is the JNDI namespace located?

○ LDAP server
● File system
○ Other

JNDI Service Provider

Factory class: com.sun.jndi.fscontext.RefFSContextFactory

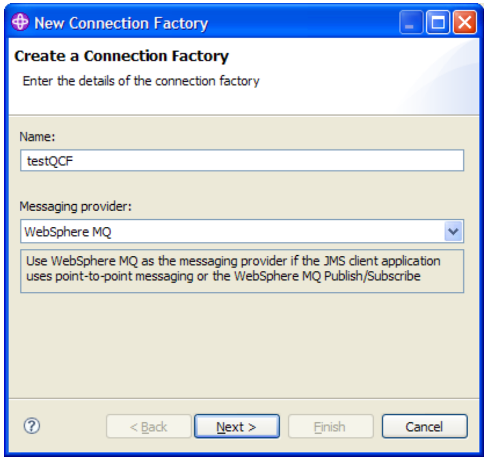
JNDI Namespace Location

Bindings directory: D:\SHARE

Provider URL: file:/D:/SHARE/

MQ Explorer Demo Screenshots

- Create a new connection factory, and provide a name


New Connection Factory

Create a Connection Factory

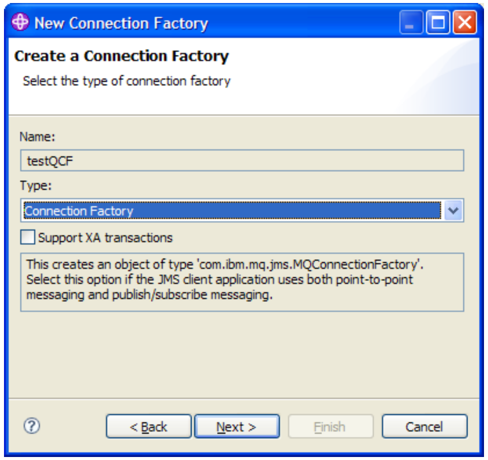
Enter the details of the connection factory

Name: testQCF

Messaging provider: WebSphere MQ

Use WebSphere MQ as the messaging provider if the JMS client application uses point-to-point messaging or the WebSphere MQ Publish/Subscribe

- Select a type of connection factory to create.


New Connection Factory

Create a Connection Factory

Select the type of connection factory

Name: testQCF

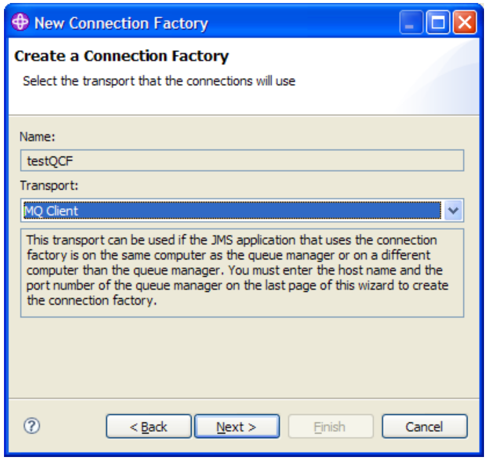
Type: Connection Factory

□ Support XA transactions

This creates an object of type 'com.ibm.mq.jms.MQConnectionFactory'. Select this option if the JMS client application uses both point-to-point messaging and publish/subscribe messaging.

MQ Explorer Demo Screenshots

- Select the type of transport to use



Create a Connection Factory

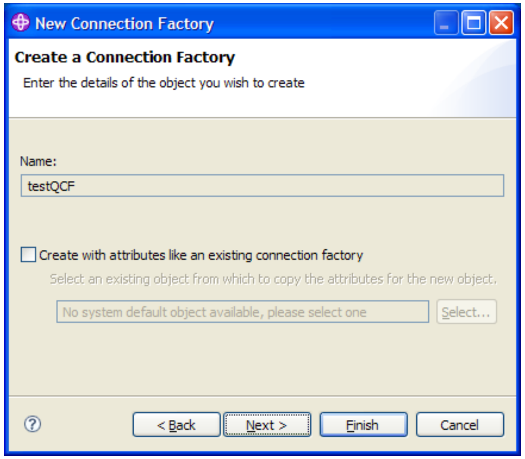
Select the transport that the connections will use

Name: testQCF

Transport: MQ Client

This transport can be used if the JMS application that uses the connection factory is on the same computer as the queue manager or on a different computer than the queue manager. You must enter the host name and the port number of the queue manager on the last page of this wizard to create the connection factory.

- Base the definition on a previously created object if needed.



Create a Connection Factory

Enter the details of the object you wish to create

Name: testQCF

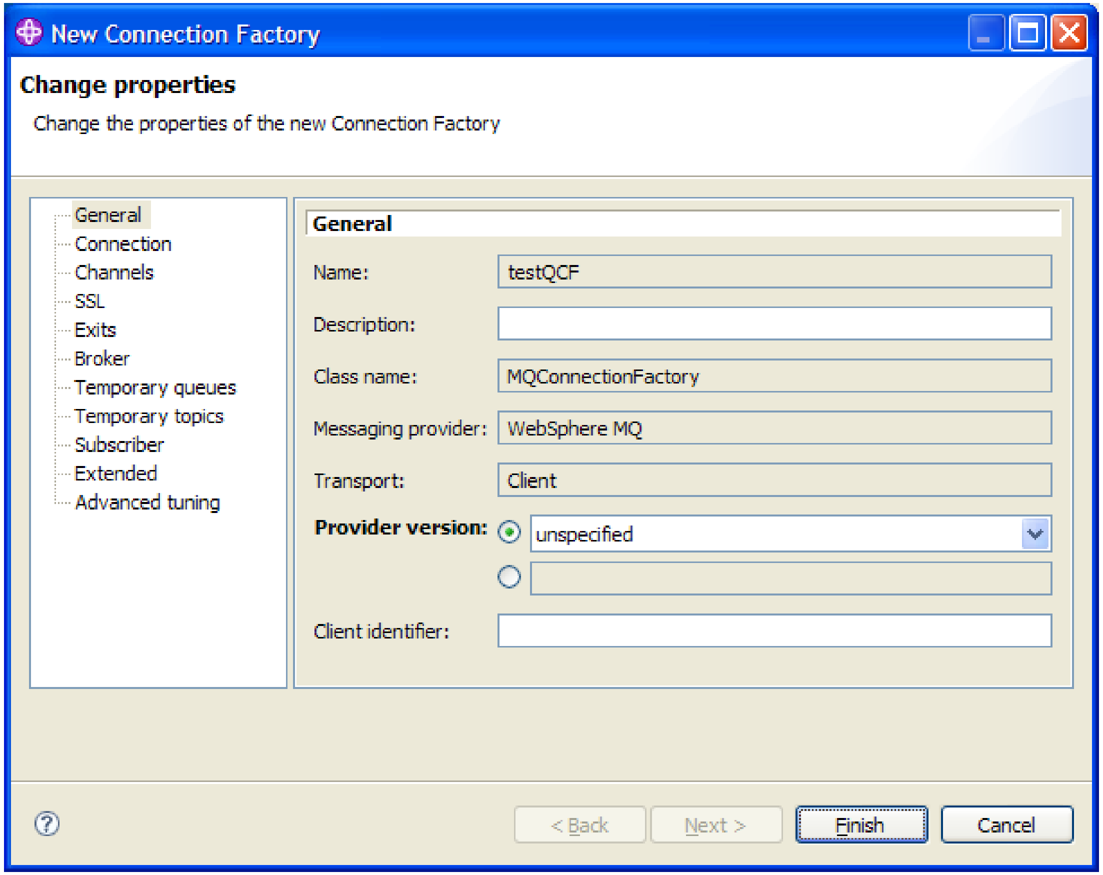
□ Create with attributes like an existing connection factory

Select an existing object from which to copy the attributes for the new object.

No system default object available, please select one

MQ Explorer Demo Screenshots

- Set the properties for the connection, and click Finish



New Connection Factory

Change properties

Change the properties of the new Connection Factory

General

Name: testQCF

Description:

Class name: MQConnectionFactory

Messaging provider: WebSphere MQ

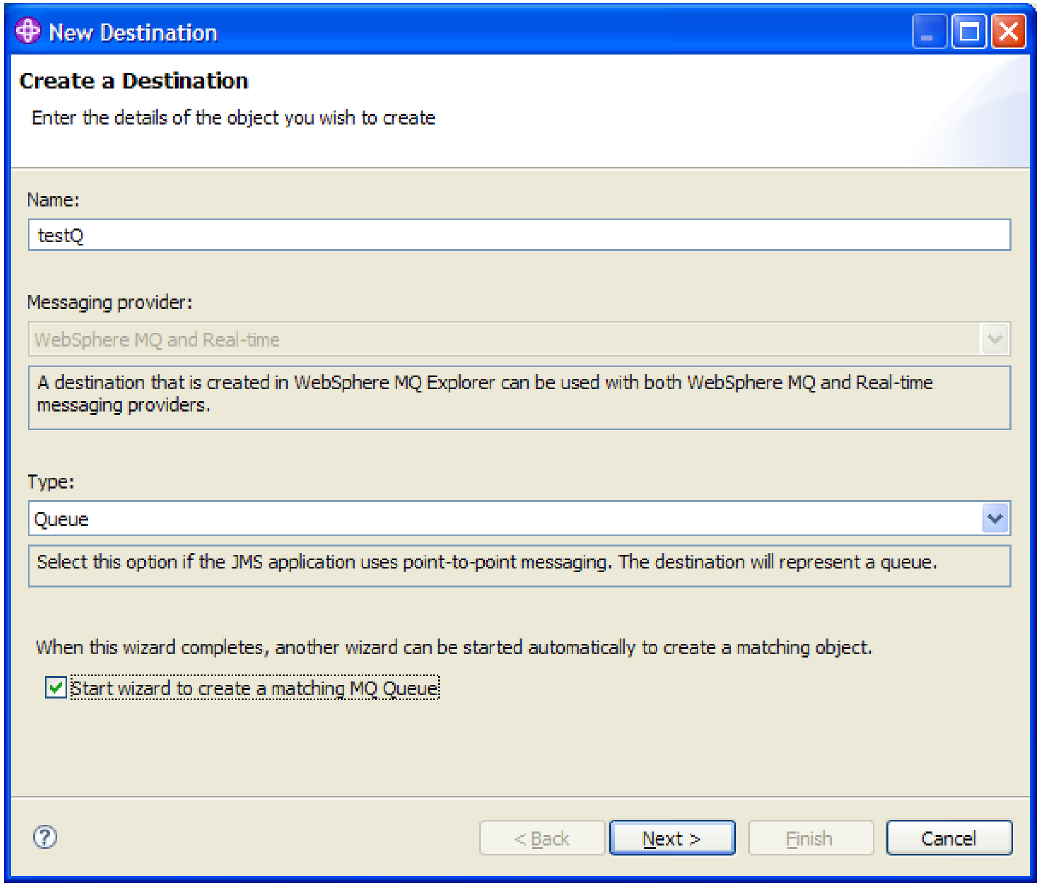
Transport: Client

Provider version: ● unspecified
○

Client identifier:

MQ Explorer Demo Screenshots

- Create a New JMS Queue, and tick the box to create the underlying MQ Queue



Create a Destination

Enter the details of the object you wish to create

Name: testQ

Messaging provider: WebSphere MQ and Real-time

A destination that is created in WebSphere MQ Explorer can be used with both WebSphere MQ and Real-time messaging providers.

Type: Queue

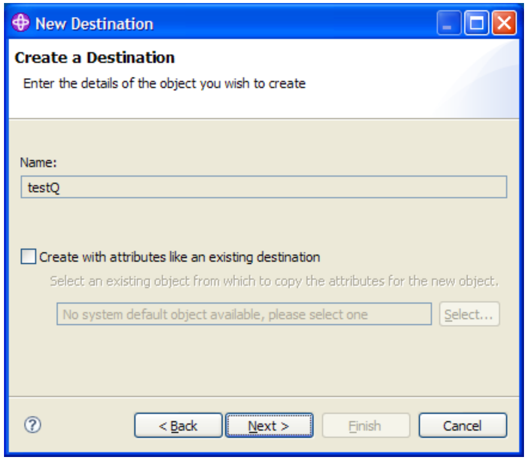
Select this option if the JMS application uses point-to-point messaging. The destination will represent a queue.

When this wizard completes, another wizard can be started automatically to create a matching object.

☑ Start wizard to create a matching MQ Queue

MQ Explorer Demo Screenshots

- Optionally create like an existing destination



New Destination

Create a Destination

Enter the details of the object you wish to create

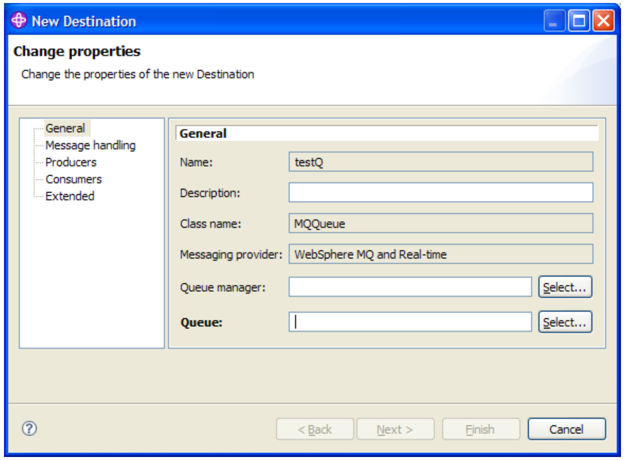
Name: testQ

□ Create with attributes like an existing destination

Select an existing object from which to copy the attributes for the new object.

No system default object available, please select one

- Set the properties for the destination object. The wizard to create the MQ Queue will start after clicking Finish.



New Destination

Change properties

Change the properties of the new Destination

General

Name: testQ

Description:

Class name: MQQueue

Messaging provider: WebSphere MQ and Real-time

Queue manager:

Queue: