



IBM Software Group

# Integrating WebSphere MQ V7 Publish/ Subscribe Feature with WebSphere DataPower SOA Appliance

Chin Sahoo ([chintam3@us.ibm.com](mailto:chintam3@us.ibm.com))  
DataPower SOA Appliances Support  
17 August 2010



WebSphere® Support Technical Exchange



# Agenda

- MQ V7 Publish/Subscribe Features
- MQ V7 Configurations for pub/sub
- DataPower Configurations for pub/sub
- Trouble Shooting Techniques
- Useful Links
- Questions and Answers

# MQ V7 Pub/Sub Features

- MQ V7 Publish/Subscribe becomes an in-built part of the MQ API (Application Programming Interface)
- MQ V7 extends the MQ API to use the publish/subscribe models with ease. New verbs and changes to existing verbs are provided in the product
- MQ V7 also extends the administrative interfaces (MQSC and PCF) to allow administrators to manage Publish/Subscribe applications
- MQ V7 Explorer can now be used to create/modify/delete Topic, Queue, Publication and Subscription Objects in QMGR

# MQ V7 Pub/Sub Features Continued

- Publish/Subscribe is an application model in which the provider of information is decoupled from the consumers of that information
- Providers of information are called **publishers**
- Consumers of information are called **subscribers**
- New providers/consumers can be added without disruption

# MQ V7 Pub/Sub Features Continued

- MQ Publish/Subscribe model connects publishing and subscribing applications via the topic or subject which the publisher associates with the information
- Subscribers need to agree on the topic to become connected to the published message
- Subscribers nominate which types of information they want to receive by subscribing to specific topics

# MQ V7 Pub/Sub Features Continued

- Publishers of information are unaware of subscribers to the extent that they may publish information even if there are no subscribers
- Publishing and subscribing are completely dynamic processes. New subscribers and new publishers can be added to the system without disruption
- With respect to a given topic, all possible combinations of publishers/subscribers are possible, such as (one-to-one, one-to-many and many-to-many)



# MQ V7 Pub/Sub Features Continued

- Subscribers register with the queue manager to **receive** information relating to specific topics. They use the MQSUB verb
- Publishers provide information about specific topics by sending publications to the queue manager. They use the MQPUT verb
- The queue manager forwards each publication it receives to all subscribers with a subscription that matches the associated topic



# MQ V7 Pub/Sub Features Continued

- **Types of publications**
- Events:
  - ▶ Continuing succession of logically independent messages, for example:
    - ▶ Trades
    - ▶ Customer buying an airline ticket
    - ▶ Subscribers receive as available



# MQ V7 Pub/Sub Features Continued

- **Types of publications Continued..**
- State
  - ▶ Information that is being regularly updated or replaced, for example:
    - stock prices
    - furnace temperatures
  - ▶ Queue managers can retain copy of the last publication
  - ▶ Subscribers may receive immediately or check at their own initiative



# MQ V7 Pub/Sub Features Continued

- Messages are published to a topic string. When publishing, users have to specify one topic string only
- Subscribers can subscribe to the topic string. Once messages are published to the topic string, these messages are delivered to the subscribers
- Subscribers can subscribe to many topic strings at the same time. They can use wild cards in the topic string



# MQ V7 Pub/Sub Features Continued

- Topic String Example:
  - ▶ /Sports/# (pound sign) means all topic strings with the top level topic of “Sports”
- Topic strings are organized into a logical hierarchy, like /Sports/Tennis or /Sports/Golf
- When designing a set of topics, it is strongly recommended to use the / character to add structure to the topic strings



# MQ V7 Pub/Sub Features Continued

- **Subscription Type:**
- Subscription can be durable or non-durable
  - ▶ Non-durable subscriptions mean messages are delivered to subscribers only while they are connected. It means consumers will not receive message published to a topic when they are disconnected
  - ▶ Messages for durable subscriptions are delivered to subscribers when they collect next



# MQ Configurations for Pub/Sub

- Configuration Steps for MQ Server
- Create a Queue Manager
- Create Topic and Topic Strings
- Create Subscription Queues
- Create Subscriptions
- Test Publication on a Topic String
- Test Subscriptions

# Creating QMGR with MQ Explorer

**Create Queue Manager**

**Queue Manager**  
Enter basic values

**Queue manager name:** WSTEQM

Make this the default queue manager

Default transmission queue:

Dead-letter queue:

Max handle limit: 256

Trigger interval: 99999999

Max uncommitted messages: 10000

# Creating QMGR continued..

**Create Queue Manager**

Queue Manager

Enter data and log values

Queue manager name: WSTEQM

Use circular logging  
 Use linear logging

Log file size: (x4KB) 4096

Log primary files: 3

Log secondary files: 2

Data and Log paths

Use default paths

Data path: C:\Program Files\IBM\WebSphere MQ\qmgrs Browse...

Log path: C:\Program Files\IBM\WebSphere MQ\log Browse...

< Back Next > Finish Cancel

# Creating Topic with MQ Explorer

**New Topic**

**Create a Topic**

Enter the details of the object you wish to create

Name:  
Sports.Topic

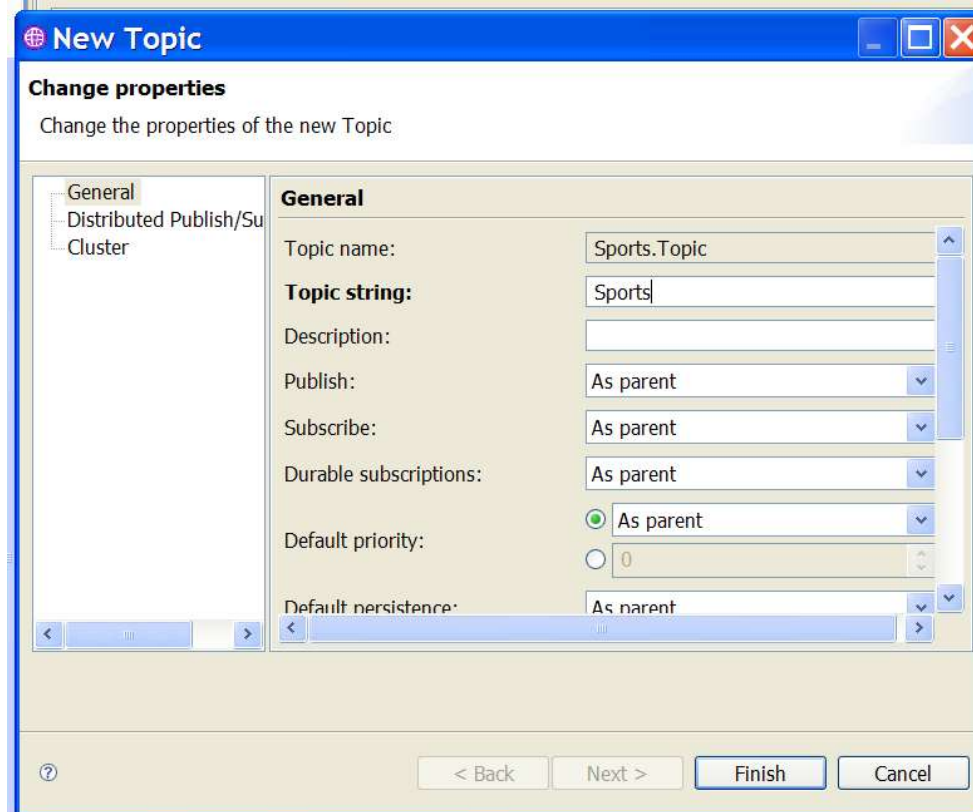
Select an existing object from which to copy the attributes for the new object.  
SYSTEM.DEFAULT.TOPIC Select...

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

? < Back Next > Finish Cancel



# Creating Topic continued..



**New Topic**

**Change properties**  
Change the properties of the new Topic

General  
Distributed Publish/Su  
Cluster

**General**

Topic name: Sports.Topic

**Topic string:** Sports

Description:

Publish: As parent

Subscribe: As parent

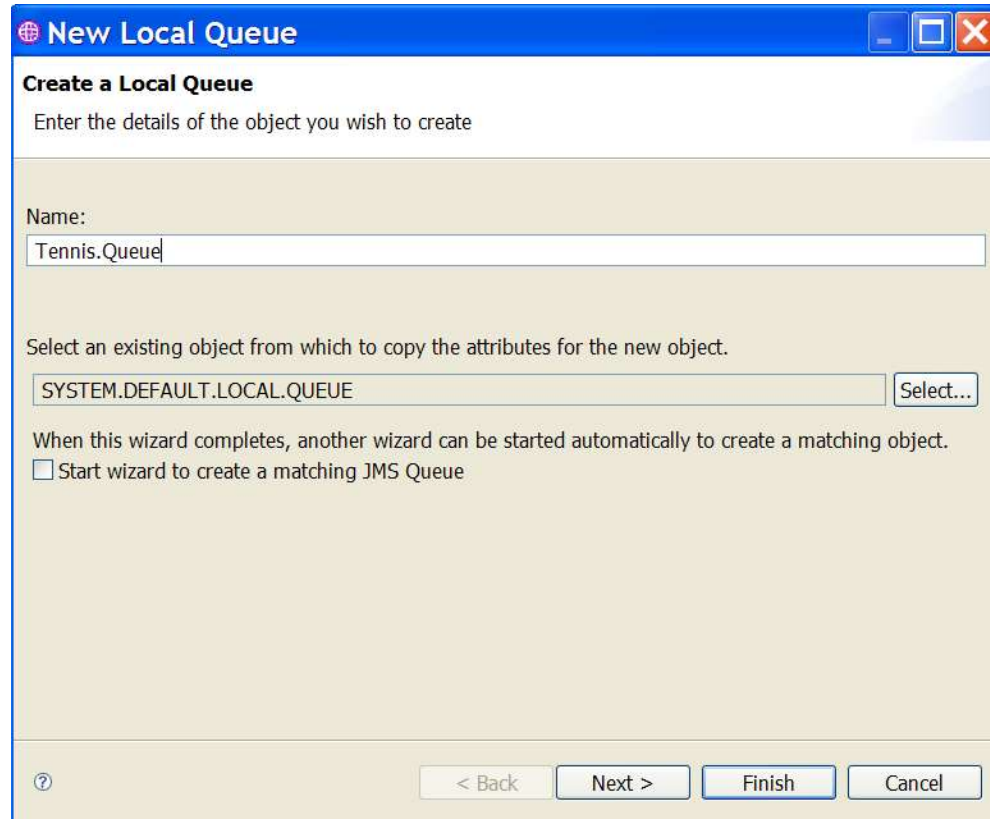
Durable subscriptions: As parent

Default priority:  As parent  0

Default persistence: As parent

< Back Next > Finish Cancel

# Creating MQ Queue with MQ Explorer



**New Local Queue**

**Create a Local Queue**

Enter the details of the object you wish to create

Name:  
Tennis.Queue

Select an existing object from which to copy the attributes for the new object.  
SYSTEM.DEFAULT.LOCAL.QUEUE

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

# Creating MQ Queue continued..

**New Local Queue**

**Change properties**  
Change the properties of the new Local Queue

General  
Extended  
Cluster  
Triggering  
Events  
Storage  
Statistics

**General**

Queue name: Tennis.Queue

Queue type: Local

Description:

Put messages: Allowed

Get messages: Allowed

Default priority: 0

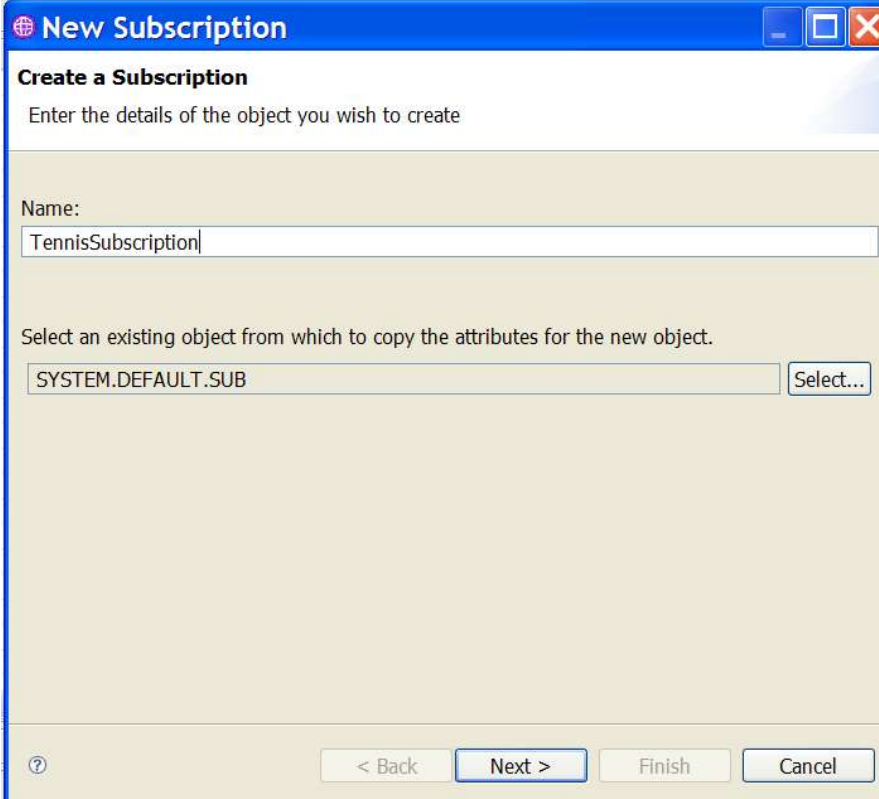
Default persistence: Not persistent

Scope: Queue manager

Usage: Normal

< Back   Next >   Finish   Cancel

# Create Subscription with MQ Explorer



The screenshot shows a dialog box titled "New Subscription" with a blue header bar. Below the title bar, the text "Create a Subscription" is displayed in bold, followed by the instruction "Enter the details of the object you wish to create".

The dialog contains two main input sections:

- Name:** A text input field containing the text "TennisSubscription".
- Select an existing object from which to copy the attributes for the new object.** A dropdown menu showing "SYSTEM.DEFAULT.SUB" and a "Select..." button to the right.

At the bottom of the dialog, there is a row of four buttons: a help button (question mark icon), "< Back", "Next >" (highlighted with a blue border), "Finish", and "Cancel".

# Create Subscription continued..

General

Extended

**General**

Subscription name:

**Topic**

Topic name:

Topic string:

Wildcard usage:

Scope:

**Destination**

Destination class:

Destination queue manager:

Destination name:

Correlation identifier:

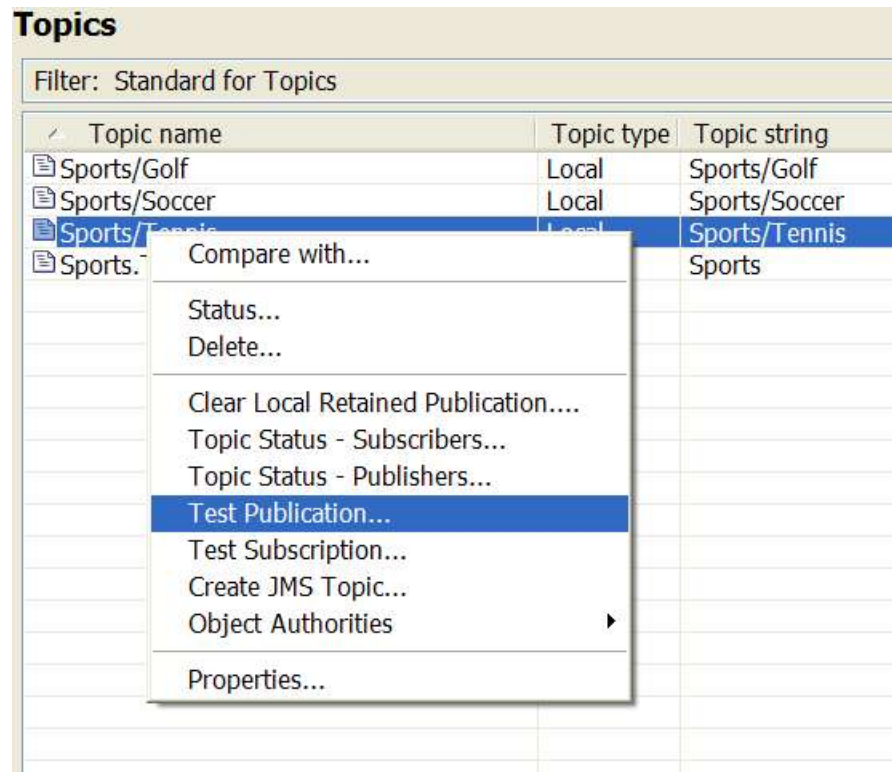
Properties:

User data:

Selector:



# Publication/Subscription Test with MQ Explorer



## DataPower MPGW Service Configuration for MQ Pub/Sub Traffic

- ▶ Request Rule only, no Response Rule
- ▶ May have Error Rule
- ▶ Process Backend Errors is “off” under the advanced tab of the Multi-Protocol Gateway (MPGW) Service
- ▶ Request Type as “XML”, “SOAP”, “non-XML” or “pass-thru”
- ▶ Response Type as “pass-thru”
- ▶ Request MQMD is not altered
- ▶ Backside MQ URL specifies (1) “PublishTopicString” and “SubscribeTopicString” (2) “PublishTopicString” (3) “PublishTopicString” and “ReplyQueue” for reply message

## DataPower Pub/Sub configuration – MQ FSH

- Specify **Subscribe Topic String**
- If response is needed, **Publish Topic String**

Get Queue	<input type="text"/>	*
Subscribe Topic String	<input type="text"/>	*
Put Queue	<input type="text"/>	
Publish Topic String	<input type="text"/>	
Subscription Name	<input type="text"/>	
CCSI	<input type="text" value="0"/>	
Get Message Options	<input type="text" value="4097"/>	
Selector	<input type="text"/>	
Parse Properties	<input type="radio"/> on <input checked="" type="radio"/> off	
Async Put	<input type="radio"/> on <input checked="" type="radio"/> off	



## DataPower Pub/Sub configuration – MQ FSH

- Mix up with original Get Queue and Put Queue
  - ▶ user can have get, sub, get-put, get-pub, sub-put, sub-pub in the front side
  - ▶ If both Get Queue and Subscribe Topic String is present, will use Get Queue

Get Queue	<input type="text"/>	*
Subscribe Topic String	<input type="text"/>	*
Put Queue	<input type="text"/>	
Publish Topic String	<input type="text"/>	
Subscription Name	<input type="text"/>	
CCSI	<input type="text" value="0"/>	
Get Message Options	<input type="text" value="4097"/>	
Selector	<input type="text"/>	
Parse Properties	<input type="radio"/> on <input checked="" type="radio"/> off	
Async Put	<input type="radio"/> on <input checked="" type="radio"/> off	

# DataPower Pub/Sub configuration – Backend/url-open

- MQHelper of static backend will also have options for **Subscribe Topic String** and **Publish Topic String**

Build a MQ URL

Queue Manager:  
   \*

URI:

RequestQueue:  
 \*

**PublishTopicString:**  
 \*

ReplyQueue:  
 \*

**SubscribeTopicString:**  
 \*

**SubscriptionName:**

## DataPower Pub/Sub configuration – Backend/url-open

- Mix up with original RequestQueue and ReplyQueue
  - ▶ user can have request, reply, sub, pub, request-reply, request-sub, pub-reply, pub-sub in the MQ URL
  - ▶ DataPower does not match up MsgId/CoreId for pub-sub and request-sub

- Example

dpmq://QM/?PublishTopicString=xxxx;SubscribeTopicString=yyyy

dpmq://QM/?PublishTopicString=xxxx

dpmq://QM/?PublishTopicString=xxxx;ReplyQueue=yyyy



# MQ Message Properties - MQMP

- Message properties are optional user data with name/value pairs that can be added to message
- In Multistep, user can manipulate message properties
- Message properties and Message Selector allow user to select messages without accessing MQMD or MQRFH2 headers
- Message selector can get the messages that have some specific message properties



## Message Selector – MQ FSH and Backend URL

- Specify a selecting criteria in Selector field ex. color = 'red'
  - Specify Selector parameter for backend URL
- `dpmq://QM/?RequestQueue=xxx;Selector=color='red';`

Get Queue	<input type="text"/>	*
Subscribe Topic String	<input type="text"/>	*
Put Queue	<input type="text"/>	
Publish Topic String	<input type="text"/>	
Subscription Name	<input type="text"/>	
CCSI	<input type="text" value="0"/>	
Get Message Options	<input type="text" value="4097"/>	
Selector	<input type="text"/>	
Parse Properties	<input type="radio"/> on <input checked="" type="radio"/> off	
Async Put	<input type="radio"/> on <input checked="" type="radio"/> off	

# DataPower MPGW Service Configuration for Publication

Configure Multi-Protocol Gateway

General
Advanced
Stylesheet Params
Headers
Monitors
WS-Addressing
WS-ReliableMessaging

Apply Cancel Delete
Export | View Log | View Status | Show Probe | Validate Conformance | Help

Multi-Protocol Gateway status: [up]

### General Configuration

**Multi-Protocol Gateway Name**  
mq-pub-sub-example \*

**Summary**  
dpmq://DP4/?PublishTopicString=T

**Type**  
 dynamic-backend  
 static-backend  
 \*

**XML Manager**  
default + ... \*

**Multi-Protocol Gateway Policy**  
mq-pub-sub-example-policy + ... \*

**URL Rewrite Policy**  
(none) + ...

---

**Back side settings**

**Backend URL**  
?PublishTopicString=Sports/Tennis \*

MQHelper TibcoEMSHelper  
WebSphereJMSHelper IMSConnectHelper

**Front side settings**

**Front Side Protocol**  
mq-pub-fsh (MQ Front Side Handler) ✕  
http-pub-fsh (HTTP Front Side Handler) ✕

+ Add + ... \*

**User Agent settings**

Match	Property
<i>Note:</i> To edit the User Agent, please access via the XML Manager above.	

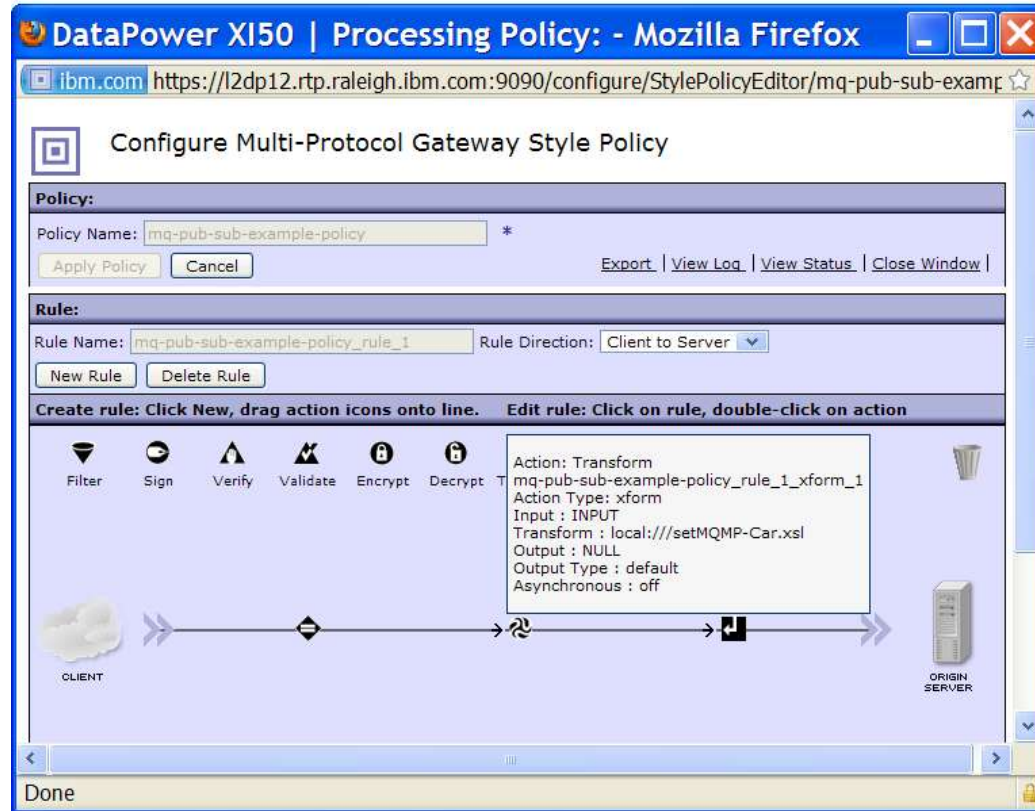
**SSL Client Crypto Profile**  
(none) + ...

**Response Type**  
 Non-XML  
 Pass-Thru  
 SOAP  
 XML

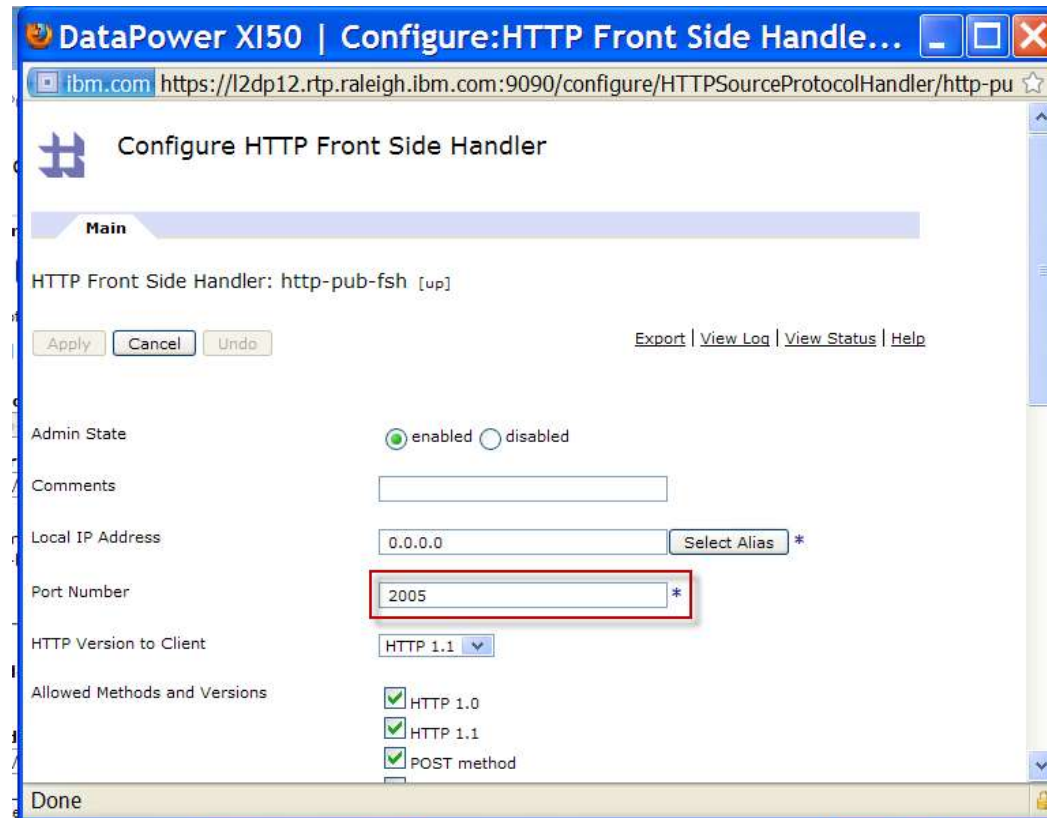
**Request Type**  
 Non-XML  
 Pass-Thru  
 SOAP  
 XML



# DataPower MPGW Style Policy for Publication



# DataPower XI50 HTTP FSH for Publication





# DataPower MPGW Style Policy for Subscription

The screenshot shows the DataPower XI50 web interface for configuring a Multi-Protocol Gateway (MPGW) Style Policy. The browser window title is "DataPower XI50 | Processing Policy: - Mozilla Firefox". The URL is "https://l2dp12.rtp.raleigh.ibm.com:9090/configure/StylePolicyEditor/mq-sub-example-policy".

The main heading is "Configure Multi-Protocol Gateway Style Policy".

**Policy:**  
 Policy Name:  \*  
 Buttons:   [Export](#) | [View Log](#) | [View Status](#) | [Close Window](#)

**Rule:**  
 Rule Name:  Rule Direction:   
 Buttons:

**Create rule: Click New, drag action icons onto line. Edit rule: Click on rule, double-click on action**

Available actions: Filter, Sign, Verify, Validate, Encrypt, Decrypt, Transform (selected), and a trash icon.

The rule flow diagram shows a sequence of actions between a "CLIENT" and an "ORIGIN SERVER". The selected "Transform" action is highlighted with a tooltip showing its configuration:

```

Action: Transform
mq-sub-example-policy_rule_0_xform_0
Action Type: xform
Input : INPUT
Transform : local:///setRoute-sub.xml
Output : NULL
Output Type : default
Asynchronous : off
    
```

The status bar at the bottom indicates "Done".



# DataPower Transaction Subscription List

The image displays three screenshots of the DataPower Transaction Subscription List interface, each shown in a Mozilla Firefox browser window. The interface includes a header with 'WebSphere. DataPower XI50' and a toolbar with buttons like 'Refresh', 'Flush', 'Disable Probe', 'Export Capture', 'View Log', 'Send Message', and 'Close'. Below the toolbar is a table with columns: 'view trans#', 'type', 'inbound-url', 'outbound-url', 'rule', and 'client-ip'.

**Top-Left Window: Transaction List for mq-sub-responder-red**

view trans#	type	inbound-url	outbound-url	rule	client-ip
505248	request	http://127.0.0.1:3002/Red	http://127.0.0.1:3002/Red	mq-sub-responder-red_request	127.0.0.1

**Top-Right Window: Transaction List for mq-sub-responder**

view trans#	type	inbound-url	outbound-url	rule	client-ip
161207	request	http://127.0.0.1:3001/Grey	http://127.0.0.1:3001/Grey	mq-sub-responder_request	127.0.0.1

**Bottom Window: Transaction List for mq-sub-example**

view trans#	type	inbound-url	outbound-url	rule	client-ip
116931	request	dpmq://WSTEQM/mq-sub-mp?SubscribeTopicString=Sports/Tennis	http://127.0.0.1:3002/Red	mq-sub-example-policy_rule_0	
509584	request	dpmq://WSTEQM/mq-sub-mp?SubscribeTopicString=Sports/Tennis	http://127.0.0.1:3001/Grey	mq-sub-example-policy_rule_0	

# DataPower Routing Configuration

```
<xsl:template match="/">
  <xsl:variable name="entries" select="dp:request-header('MQMP')"/>
  <xsl:variable name="header" select="dp:parse($entries)"/>
  <xsl:variable name="color" select="$header//*[ @name='car.color']/text()"/>
  <xsl:variable name="backend-url">
  <xsl:choose>
    <xsl:when test="contains($color, 'Grey')">
      <xsl:value-of select="concat('http://127.0.0.1:3001/', $color)"/>
    </xsl:when>
    <xsl:when test="contains($color, 'Red')">
      <xsl:value-of select="concat('http://127.0.0.1:3002/', $color)"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="concat('http://127.0.0.1:3002/', 'Black')"/>
    </xsl:otherwise>
  </xsl:choose>
  </xsl:variable>
  <dp:set-variable name=""var://service/routing-url"" value="$backend-url"/>
</xsl:template>
```

# DataPower MQ FSH Selector for Subscription

**DataPower XI50 | Configure:MQ Front Side Handler - ...**

https://l2dp12.rtp.raleigh.ibm.com:9090/configure/MQSourceProtocolHandler/mq-sub-n

**Publish and Subscribe**

Subscribe Topic String: Sports/Tennis \*

Subscription Name:

Publish Topic String:

**Properties and Headers**

Parse Properties:  on  off

Selector: car.color='Grey' or car.color='Red'

Exclude Message Headers:

- CICS Bridge Header (MQCIH)
- Dead Letter Header (MQDLH)
- IMS Information Header (MQIIH)
- Rules and Formatting Header (MQRFH)
- Rules and Formatting Header (MQRFH2)
- Work Information Header (MQWIH)

Header to extract Content-Type: None

**Advanced**

Done

# DataPower MQMP Manipulation

```
<xsl:variable name="car-color">
  <xsl:choose>
    <xsl:when test="contains(dp:variable('var://service/URI'), 'grey')">
      <xsl:value-of select="'Grey'"/>
    </xsl:when>
    <xsl:when test="contains(dp:variable('var://service/URI'), 'red')">
      <xsl:value-of select="'Red'"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="'Black'"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```



# DataPower MQMP Manipulation Continued

```
<xsl:variable name="newMQMP">
  <MQMP>
    <Property name="car.color" type="string">
      <xsl:value-of select="$car-color"/>
    </Property>
    <Property name="car.year" type="int32">2001</Property>
    <Property name="car.domestic" type="boolean">TRUE</Property>
    <Property name="car.code" type="hexstr">44445546</Property>
  </MQMP>
</xsl:variable>
<xsl:variable name="ser-mqmp">
  <dp:serialize select="$newMQMP" omit-xml-decl="yes"/>
</xsl:variable>
<xsl:message dp:priority="debug">
  <xsl:value-of select="concat('The New MQMP Header : ', $ser-mqmp)"/>
</xsl:message>
<dp:set-request-header name=""MQMP"" value="$ser-mqmp"/>
```

# Troubleshooting – DataPower Side

- Enable log level to “debug” using trouble shooting icon on the control panel
- Enable probe for the particular MPGW service
- Run few transactions and observe the system log
- Look for MQ Reason Code(s) and errors in the system log
- Understand the MQ Reason Code(s) using MQ supportpac “[ma0k](#)” available at <http://www-01.ibm.com/support/docview.wss?uid=swg24000652>

## **MQ supportpac Link:**

<http://www-01.ibm.com/support/docview.wss?rs=977&uid=swg27007205>



# Troubleshooting – MQ Server Side

- MQ function
- Access via: `su – mqm`
- Display queue status
  - ▶ Processes – how many connections on get/put
  - ▶ Queue depth
  - ▶ Uncommitted messages present?
  - ▶ Queue Handles
    - Who has open connections?
    - Are those connections input or output?



# Summary

- MQ V7 Pub/Sub Features are discussed
- MQ V7 Configuration steps are provided
- DataPower Pub/Sub configurations are presented
- Dynamic Routing in DataPower is explained
- MQ Message Properties (MQMP) and Selector configuration examples are discussed
- DataPower and MQ server side trouble shooting techniques are provided

# References

- IBM® WebSphere DataPower SOA Appliance Selected product documents
- IBM WebSphere DataPower SOA Appliances webGUI Guide
- IBM WebSphere DataPower SOA Appliances Reference Guide
- IBM WebSphere DataPower XSLT extension elements, extension functions, and variables Guide

# References Continued..

- [Message Properties Explained](#)
- [MQ V7 Information Center](#)
- [MQ V7 Features and Enhancements](#)
- [MQ Application Programming Reference](#)



# Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:  
[http://www.ibm.com/software/websphere/support/supp\\_tech.html](http://www.ibm.com/software/websphere/support/supp_tech.html)
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:  
<http://www.ibm.com/developerworks/websphere/community/>
- Join the Global WebSphere User Group Community:  
<http://www.websphere.org>
- Access key product show-me demos and tutorials by visiting IBM Education Assistant:  
<http://www.ibm.com/software/info/education/assistant>
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:  
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My Notifications emails:  
<http://www.ibm.com/software/support/einfo.html>

# We Want to Hear From You!

## Tell us about what you want to learn

Suggestions for future topics  
Improvements and comments about our webcasts  
We want to hear everything you have to say!

**Please send your suggestions and comments to:**  
[wsehelp@us.ibm.com](mailto:wsehelp@us.ibm.com)

# Questions and Answers