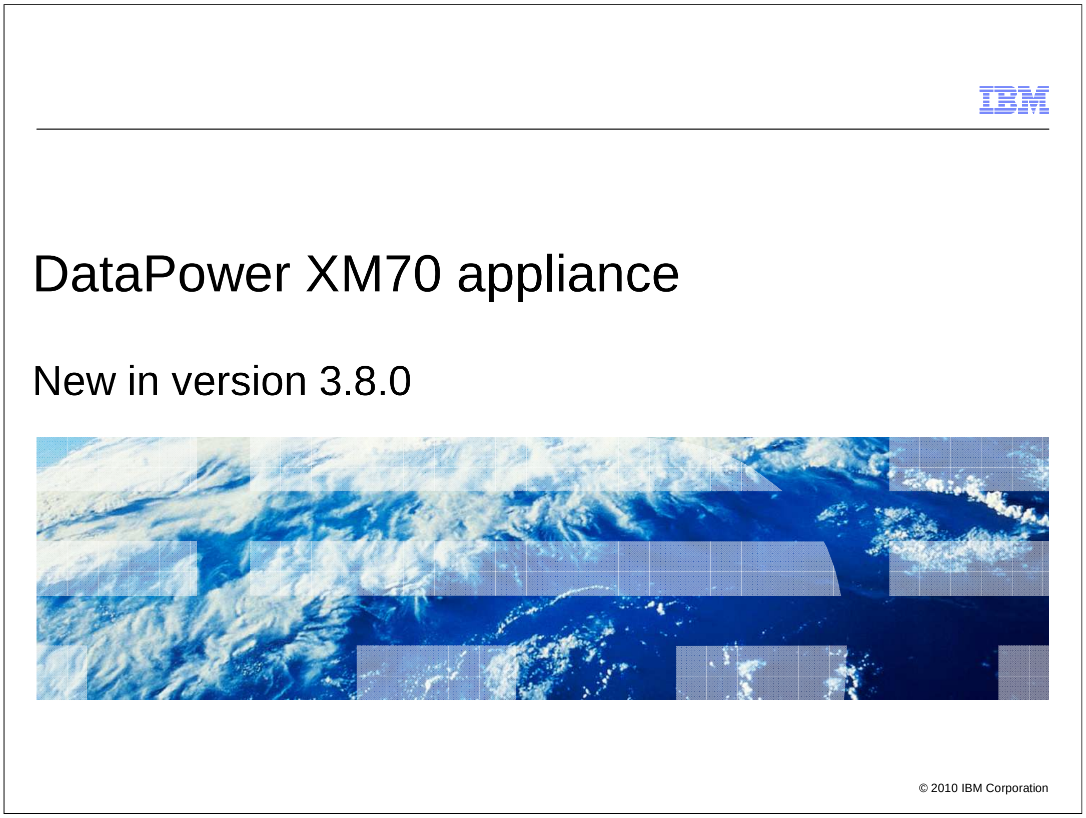
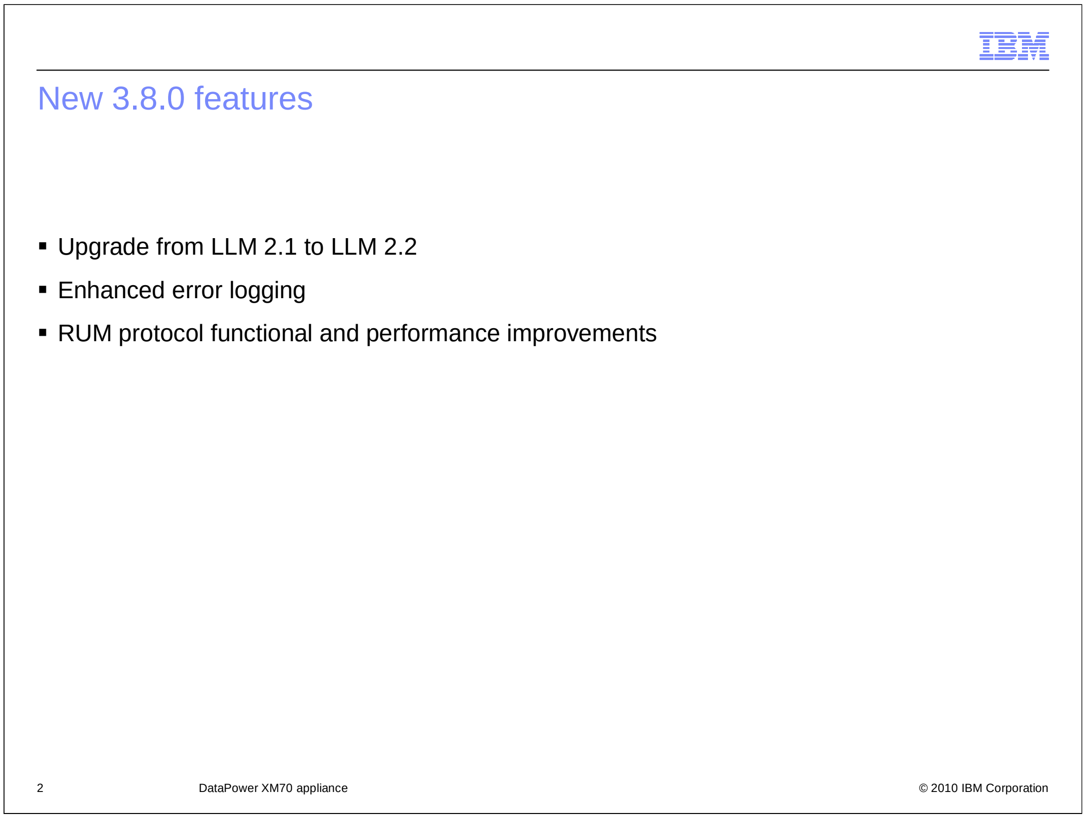


DataPower XM70 appliance

New in version 3.8.0

© 2010 IBM Corporation

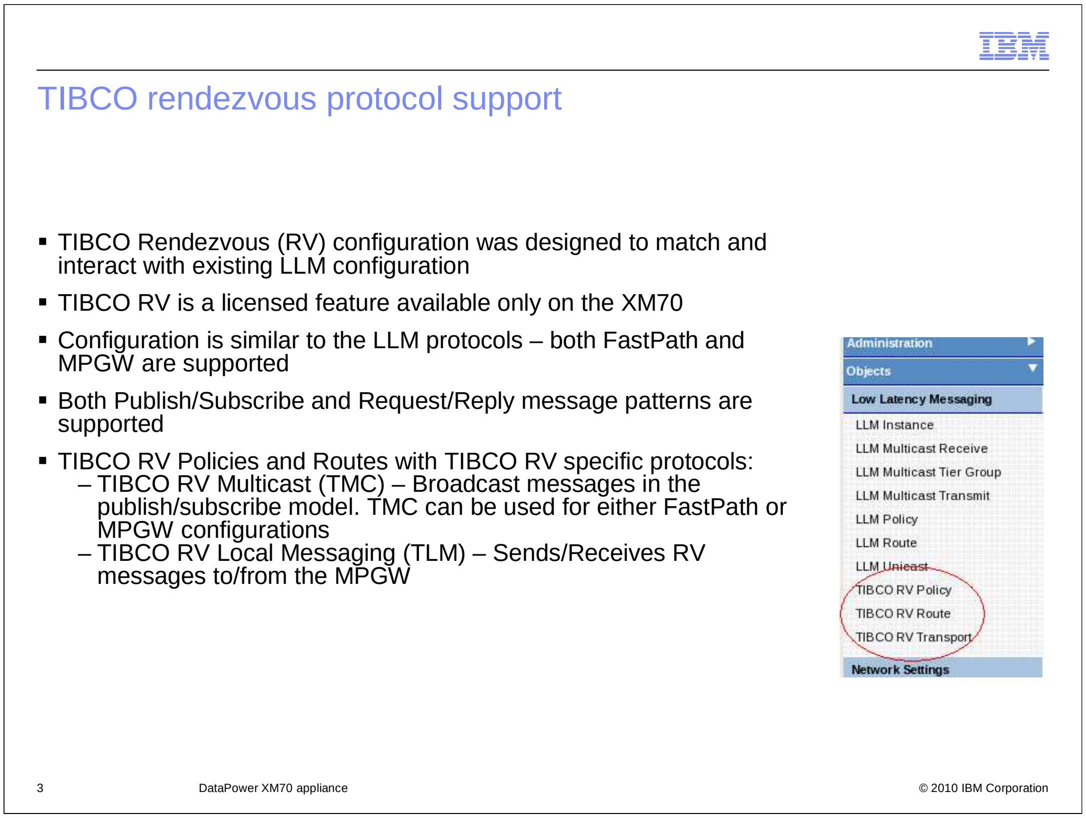


New 3.8.0 features

- Upgrade from LLM 2.1 to LLM 2.2
- Enhanced error logging
- RUM protocol functional and performance improvements

The first set of changes for 3.8.0 was for the base LLM function. Most of the changes to LLM were somewhat in the background, and not really noticeable unless you know what you are looking for. The LLM libraries have been from version 2.1 to 2.2, which is the latest available level. This brings the code in line with the WebSphere® MQ Low Latency Messaging product.

The error logging was also enhanced and expanded, and quite a few bugs were fixed. Finally, there were several functional and performance improvements made to the RUM protocol code, which brings it more in line with the RMM protocol code.



TIBCO rendezvous protocol support

- TIBCO Rendezvous (RV) configuration was designed to match and interact with existing LLM configuration
- TIBCO RV is a licensed feature available only on the XM70
- Configuration is similar to the LLM protocols – both FastPath and MPGW are supported
- Both Publish/Subscribe and Request/Reply message patterns are supported
- TIBCO RV Policies and Routes with TIBCO RV specific protocols:
  - TIBCO RV Multicast (TMC) – Broadcast messages in the publish/subscribe model. TMC can be used for either FastPath or MPGW configurations
  - TIBCO RV Local Messaging (TLM) – Sends/Receives RV messages to/from the MPGW

3 DataPower XM70 appliance © 2010 IBM Corporation

By far the biggest change for the XM70 made for the 3.8.0 release is the addition of protocol support for TIBCO Rendezvous, which is referred to as TIBCO RV. TIBCO RV should not be confused with TIBCO EMS – they are completely different protocols.

One of the main design points of the TIBCO RV support is that it should function and be configured in a similar way to how the existing LLM protocols work in the appliance. The slides that follow show that the configuration screens are very similar.

So TIBCO RV is a separately licensed feature available only on the XM70, meaning that customers will have to pay extra for it. It can be used in both FastPath and multiprotocol gateway configurations, just like LLM. But unlike LLM, TIBCO RV supports both publish/subscribe and request/reply and messaging patterns.

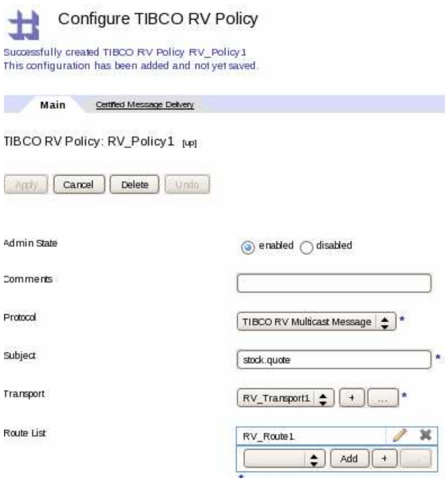
Similar to the way that LLM currently works, there are two main types of objects that deal with messages: policies and routes. Policies receive messages, and routes transmit messages. So generally a policy will be on the front-end, and a route will be on the back-end. For 3.8.0, the TIBCO RV Multicast protocol has been implemented. Like the name suggests, this protocol multicasts messages to its receivers, and can be used for both FastPath and MPGW configurations.

Also implemented is the local messaging protocol for TIBCO RV, known as TLM, which is similar to RLM for LLM. TLM sends and receives messages between the route and policy objects and the MPGW.

There are two other protocols that TIBCO RV supports that were not implement for 3.8.0: Point to point and Virtual circuits.

IBM

TIBCO RV configuration (1 of 3)

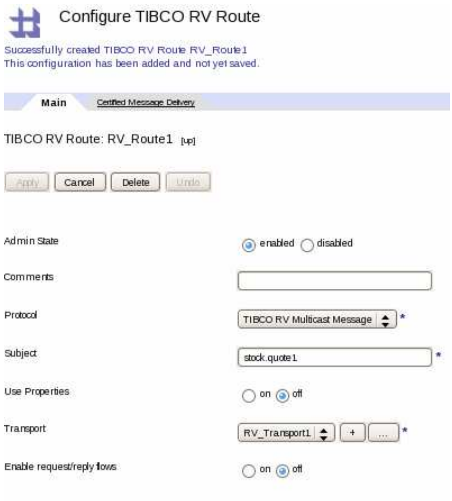


Configure TIBCO RV Policy

Successfully created TIBCO RV Policy RV_Policy1
This configuration has been added and not yet saved.

TIBCO RV Policy: RV_Policy1 [up]

Admin State	● enabled ○ disabled
Comments	
Protocol	TIBCO RV Multicast Message *
Subject	stock.quote *
Transport	RV_Transport1 *
Route List	RV_Route1 *



Configure TIBCO RV Route

Successfully created TIBCO RV Route RV_Route1
This configuration has been added and not yet saved.

TIBCO RV Route: RV_Route1 [up]

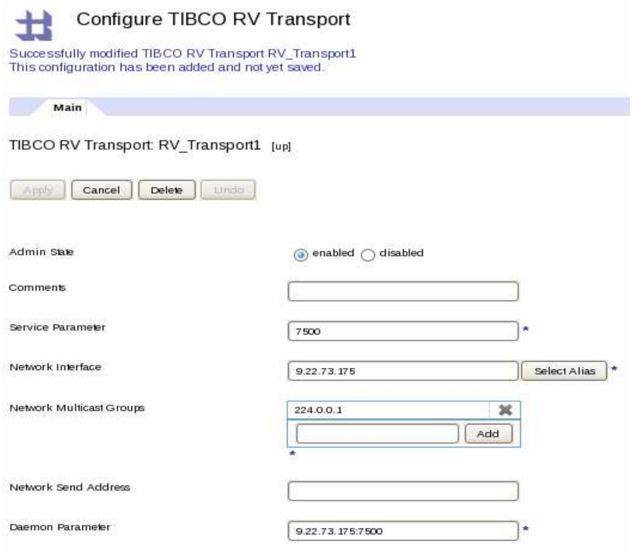
Admin State	● enabled ○ disabled
Comments	
Protocol	TIBCO RV Multicast Message *
Subject	stock.quote1 *
Use Properties	○ on ● off
Transport	RV_Transport1 *
Enable request/reply flows	○ on ● off

4 DataPower XM70 appliance © 2010 IBM Corporation

These are the configuration screens for TIBCO RV Policies and routes – Policy on the left and Route on the right. They are very similar to the LLM policy and route configuration screens, except obviously that the available protocols are different, and TIBCO RV uses a subject instead of a topic – but subjects and topics are similar concepts- they both identify a message type and allow messages to be filtered. A route object sends messages with the given subject, and a policy object listens for messages that conform to the given subject. When you listen for a subject, you can specify wildcard characters.

Another difference is that TIBCO RV routes and policies use a transport object instead of an instance object, but again, they are similar concepts as both define the transport characteristics for message transmission.

TIBCO RV configuration (2 of 3)



Configure TIBCO RV Transport

Successfully modified TIBCO RV Transport RV_Transport1
This configuration has been added and not yet saved.

TIBCO RV Transport: RV_Transport1 [up]

Admin State	● enabled ○ disabled
Comments	
Service Parameter	7500 *
Network Interface	9.22.73.175 *
Network Multicast Groups	224.0.0.1 *
Network Send Address	
Daemon Parameter	9.22.73.175:7500 *

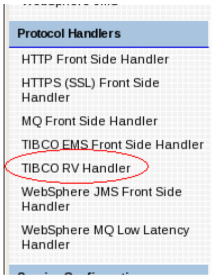
- **TIBCO RV Transport Configuration**
  - Allows the configuration of RV service, network and daemon parameters
  - RV daemon can only be accessed on a remote system (it does not run locally on the appliance). RV applications connect to an RV daemon to communicate with other RV applications across the network.

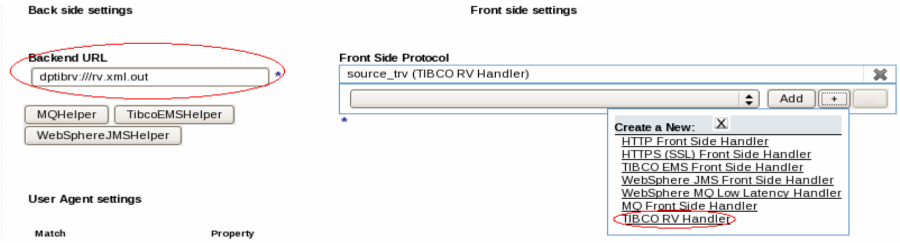
This is the configuration screen for the TIBCO RV Transport object. The main point is that these transport values have to match what the external applications in the network are using. The policy object on the front-end will generally use a different set of transport values then the route object on the back-end.

The most interesting parameter is the last one, the daemon parameter. This parameter defines where in the network the RV daemon is located. In most RV configurations, applications don’t send messages directly to each other, but instead the messages are routed through daemon processes that run within the networks. DataPower® must be configured to use external daemons, meaning that an RV daemon does not run on the appliance itself.

TIBCO RV Configuration (3 of 3)

- TIBCO RV Front Side Protocol Handler and URL Opener
  - Provide a subject to listen on for the FSPH
  - Provide a subject to send a request to on the URL Opener (dptibrv:///<your-subject-goes-here>)


Protocol Handlers
HTTP Front Side Handler
HTTPS (SSL) Front Side Handler
MQ Front Side Handler
TIBCO EMS Front Side Handler
TIBCO RV Handler
WebSphere JMS Front Side Handler
WebSphere MQ Low Latency Handler

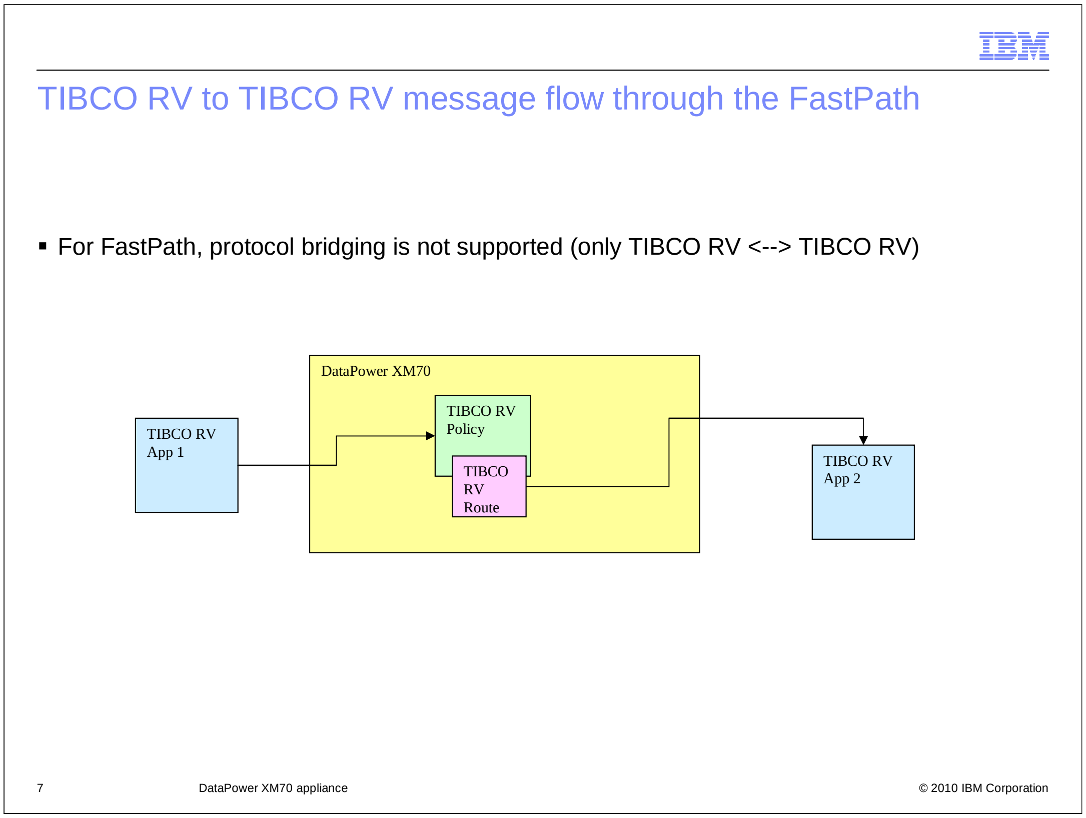


Backend URL: dptibrv:///rv.xml.out

Front Side Protocol: source_trv (TIBCO RV Handler)

TIBCO RV has its own front-side protocol handler and URL opener for when it's configured to run through a MPGW. The FSPH and URL opener actually don't do very much from a functional perspective, other than serving as attachment points to the MPGW. The actual protocol work is done in the RV policy and route objects.

The FSPH is configured with a subject name so that it can accept messages with that particular subject. The URL opener has a URL of DPTIBRV followed by another subject name, and that subject is listened to by a TLM policy.

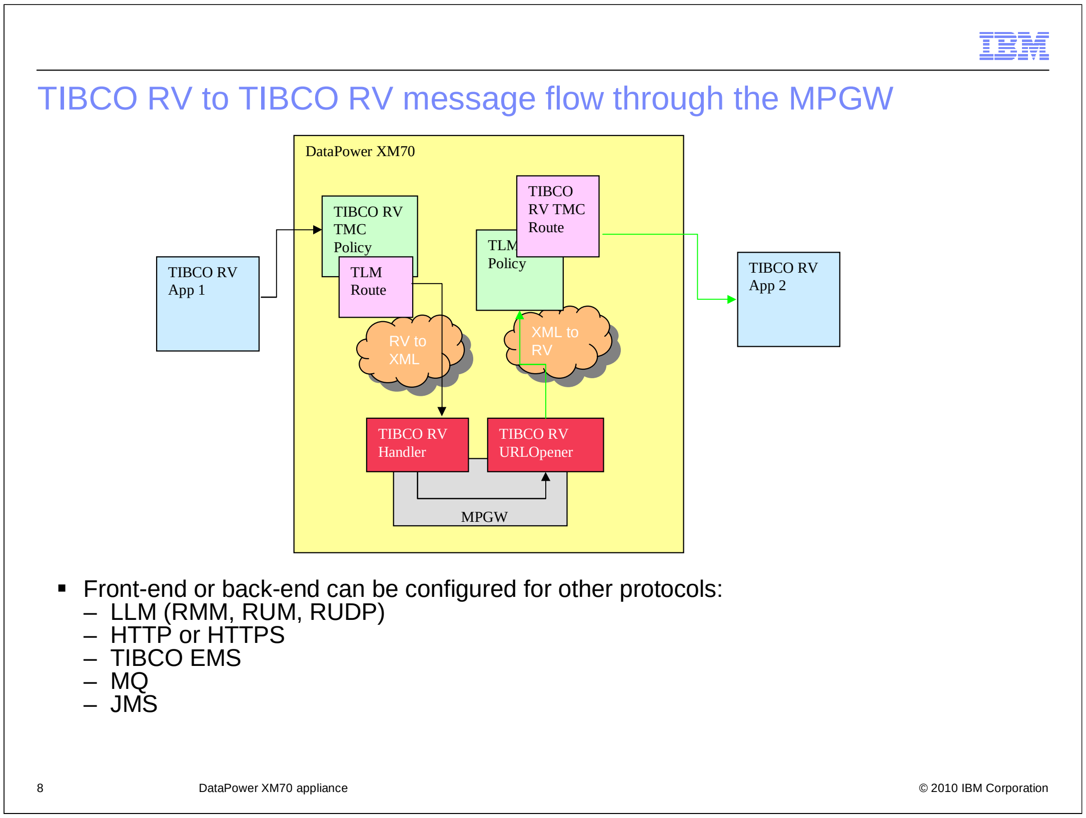


TIBCO RV to TIBCO RV message flow through the FastPath

- For FastPath, protocol bridging is not supported (only TIBCO RV <--> TIBCO RV)

© 2010 IBM Corporation

This is a block diagram of a FastPath configuration using TIBCO RV. Note the message is received in the front-end by the policy object, then routed directly from the policy to the route object, and ultimately transmitted out the back-end by the route object. Currently, FastPath only supports RV to RV configurations, meaning you can’t bridge between RV and LLM



TIBCO RV to TIBCO RV message flow through the MPGW

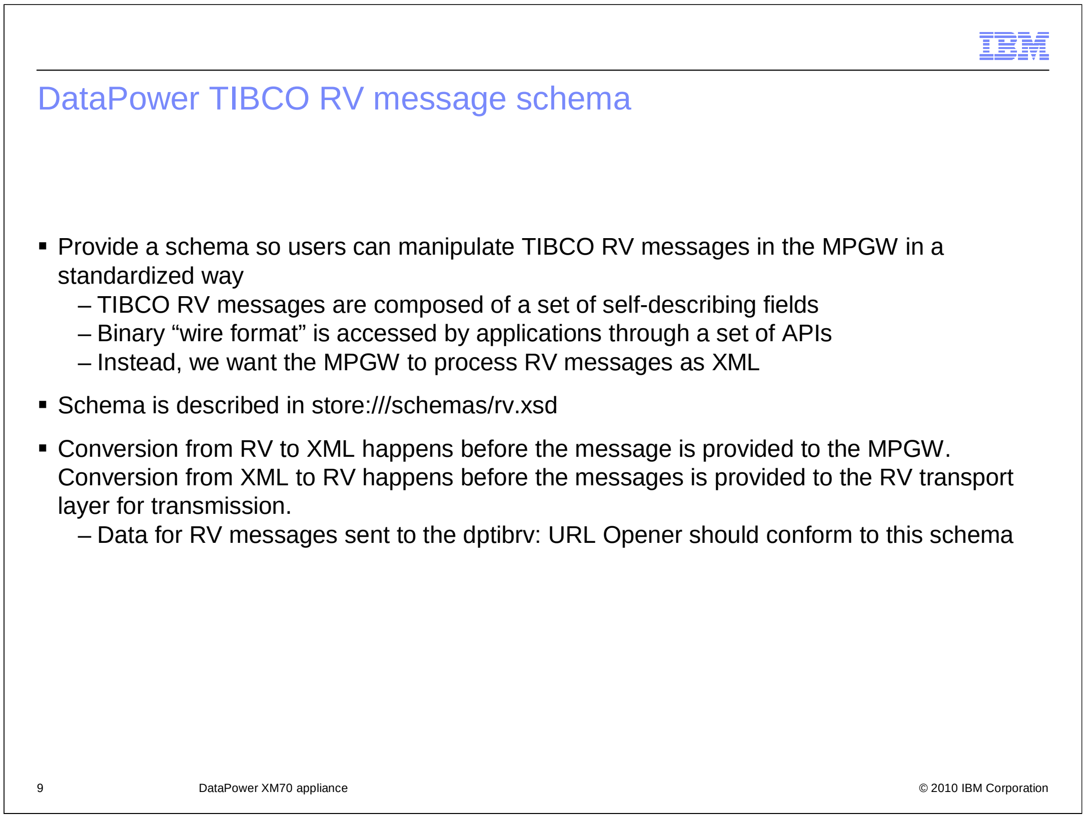
- Front-end or back-end can be configured for other protocols:
  - LLM (RMM, RUM, RUDP)
  - HTTP or HTTPS
  - TIBCO EMS
  - MQ
  - JMS

This is a block diagram of a MPGW configuration with TIBCO RV on both the front- and back-ends. On the front-end, the message is received by an RV multicast policy, then passed to a TLM route which sends it to the RV FSPH and into the gateway. After the message goes though the gateway, it's sent by the RV URL opener to a TLM policy then passed to an RV multicast route which transmits the message back out to the network.

On the front side, the TLM route sends the message with the same subject that the RV Handler is listening for, and on the back-end the URL opener transmits the message with the same subject that the TLM policy is listening for, and that's how they link up.

Another point is that the XM70 can bridge protocols in MPGW configurations, just like the other DataPower appliances. Either the front-end or back-end protocol can be configured as LLM, HTTP or HTTPS, TIBCO EMS, JMS or MQ.

You'll notice the orange cloud that the message goes through between the RV objects and the RV Handler and URL Opener. This is explained on the next page.

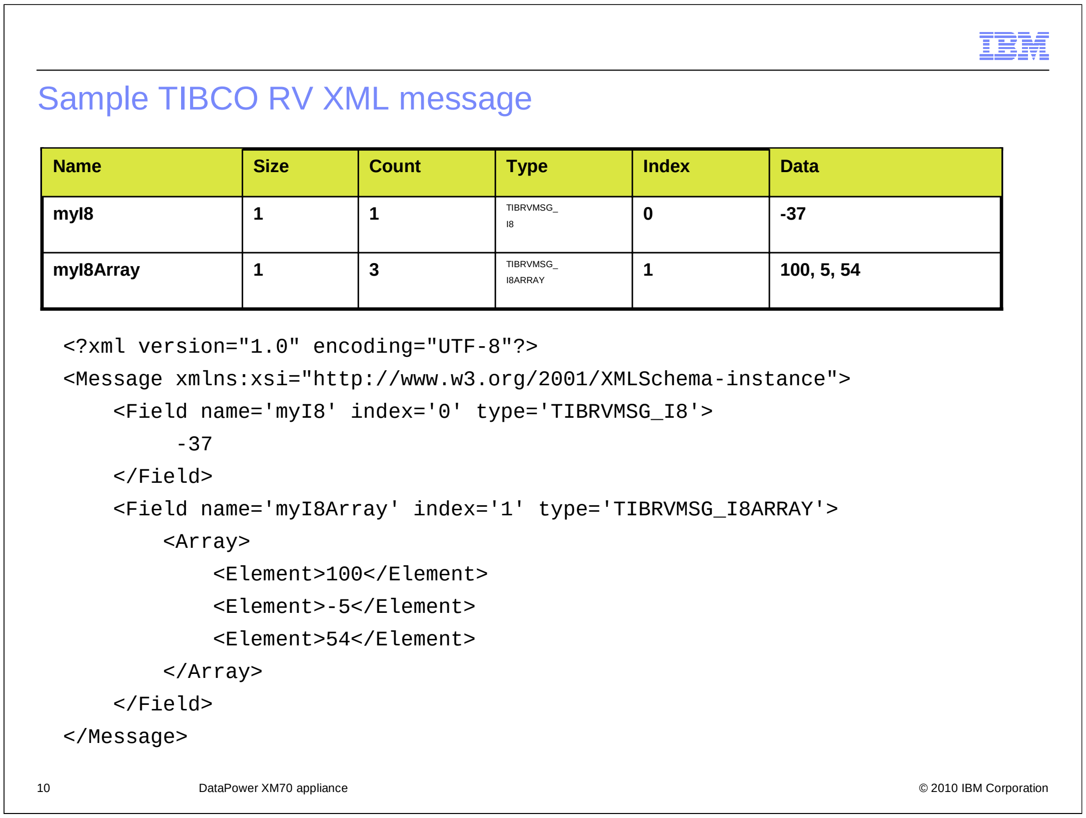


DataPower TIBCO RV message schema

- Provide a schema so users can manipulate TIBCO RV messages in the MPGW in a standardized way
  - TIBCO RV messages are composed of a set of self-describing fields
  - Binary “wire format” is accessed by applications through a set of APIs
  - Instead, we want the MPGW to process RV messages as XML
- Schema is described in store:///schemas/rv.xsd
- Conversion from RV to XML happens before the message is provided to the MPGW. Conversion from XML to RV happens before the messages is provided to the RV transport layer for transmission.
  - Data for RV messages sent to the dptibrv: URL Opener should conform to this schema

TIBCO RV messages use what is known as a self-describing format, comprised as a set of fields. These messages are transmitted over the network in a binary format that is not really conducive for processing by the MPGW. RV messages are really meant to be interrogated by using a set of APIs in the TIBCO RV libraries.

So to make message processing easier within the MPGW, a schema has been defined that describes RV messages, and the messages are automatically converted to XML defined by that schema before they enter the FSPH. Then after a message has been processed, it is automatically converted from XML back to an RV message when it goes back out the gateway through the URL opener.



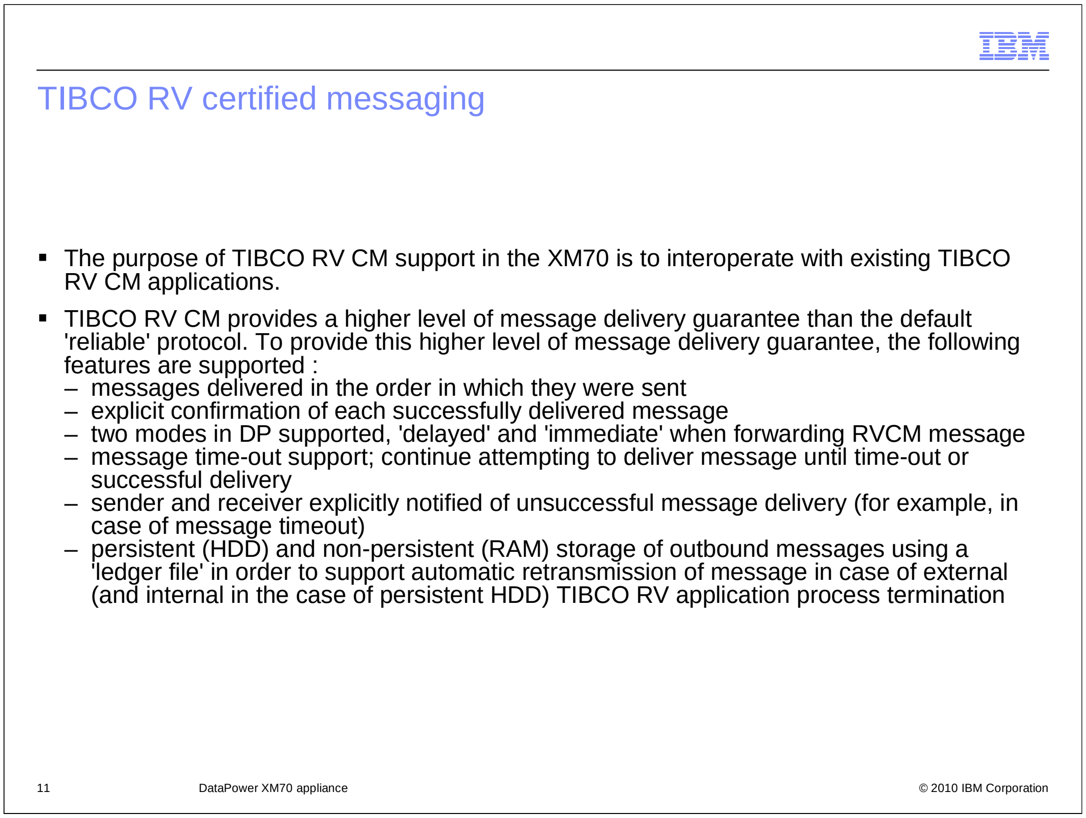
Sample TIBCO RV XML message

Name	Size	Count	Type	Index	Data
myI8	**1**	**1**	TIBRVMSG_ I8	**0**	**-37**
myI8Array	**1**	**3**	TIBRVMSG_ I8ARRAY	**1**	**100, 5, 54**

```
<?xml version="1.0" encoding="UTF-8"?>
<Message xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Field name='myI8' index='0' type='TIBRVMSG_I8'>
         -37
    </Field>
    <Field name='myI8Array' index='1' type='TIBRVMSG_I8ARRAY'>
        <Array>
            <Element>100</Element>
            <Element>-5</Element>
            <Element>54</Element>
        </Array>
    </Field>
</Message>
```

This is an example of an RV message. At the top is a block diagram of a message with two fields, the first containing a signed 8-bit integer, and the second containing an array of signed 8-bit integers. Underneath is the XML representation after the message has been converted according to the schema.

A TIBCO RV message field can contain scalar values, arrays, strings, XML and even embedded messages.



TIBCO RV certified messaging

- The purpose of TIBCO RV CM support in the XM70 is to interoperate with existing TIBCO RV CM applications.
- TIBCO RV CM provides a higher level of message delivery guarantee than the default 'reliable' protocol. To provide this higher level of message delivery guarantee, the following features are supported :
  - messages delivered in the order in which they were sent
  - explicit confirmation of each successfully delivered message
  - two modes in DP supported, 'delayed' and 'immediate' when forwarding RVCM message
  - message time-out support; continue attempting to deliver message until time-out or successful delivery
  - sender and receiver explicitly notified of unsuccessful message delivery (for example, in case of message timeout)
  - persistent (HDD) and non-persistent (RAM) storage of outbound messages using a 'ledger file' in order to support automatic retransmission of message in case of external (and internal in the case of persistent HDD) TIBCO RV application process termination

11 DataPower XM70 appliance © 2010 IBM Corporation

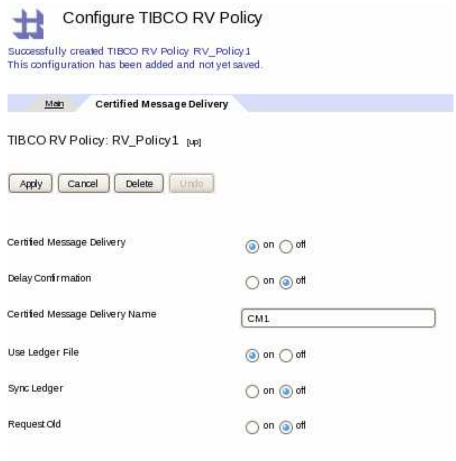
The last topic is Certified Messaging for TIBCO RV. Certified messaging behaves somewhat differently depending on whether you are the sender or receiver of the certified message. The sender wants to know that his message has been successfully delivered, and the receiver has to acknowledge back to the sender when it receives the message. Since the XM70 acts as a middleman for a certified message transactions, it has to handle the case of being a sender and receiver – a receiver on the front-end, and a sender on the back-end.

So what is the purpose of Certified Messaging? It provides a higher level of message delivery guarantee than the default ‘reliable’ protocol. It does this by doing the following: it delivers messages in the order that they were sent, and the receiver explicitly confirms receipt of each successfully delivered message.

There is an explicit timeout value that is set where the sender will continue to attempt to deliver the message, and RV will explicitly notify both the sender and receiver if the message is not delivered.

Certified messaging works by creating a ledger to maintain delivery state information, and this ledger can be either on the hard disk or in RAM. Using a RAM-based ledger provides better performance, at the cost of potentially losing messages if the appliance loses power.

TIBCO RV certified messaging (continued)



Configure TIBCO RV Policy

Successfully created TIBCO RV Policy RV_Policy1
This configuration has been added and not yet saved.

TIBCO RV Policy: RV_Policy1 [up]

Certified Message Delivery ● on ○ off

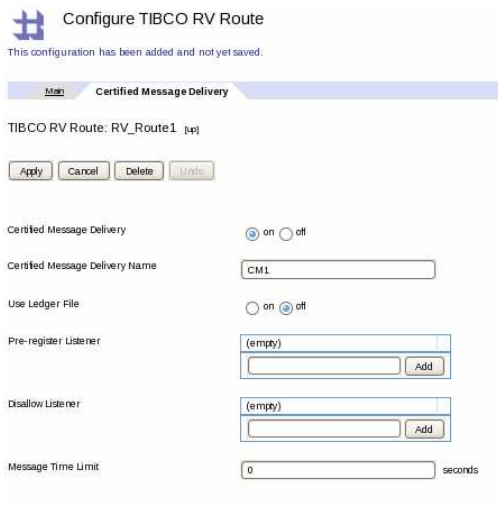
Delay Confirmation ○ on ● off

Certified Message Delivery Name CM1

Use Ledger File ● on ○ off

Sync Ledger ○ on ● off

Request Old ○ on ● off



Configure TIBCO RV Route

This configuration has been added and not yet saved.

TIBCO RV Route: RV_Route1 [up]

Certified Message Delivery ● on ○ off

Certified Message Delivery Name CM1

Use Ledger File ○ on ● off

Pre-register Listener (empty) Add

Disallow Listener (empty) Add

Message Time Limit 0 seconds

These are the screens for certified message configuration for RV policies and routes. Notice that the properties are different depending on whether you're the receiver which is the policy, or you're the sender which is the route.

Policy:

Certified Message Delivery – This is the main toggle that controls whether certified messaging is enabled

Delay confirmation – This controls whether the confirmation is sent back to the sender immediately after the route sends the message, or waits for the confirmation back from the receiver

CM delivery name – name of the CM transport, it controls whether a CM transport can be reused once it's destroyed

Use Ledger file – we've already talked about that – it controls whether the ledger is stored on the hard drive or in RAM

Sync ledger – controls whether write operations are synced to disk before continuing – this really slows things down

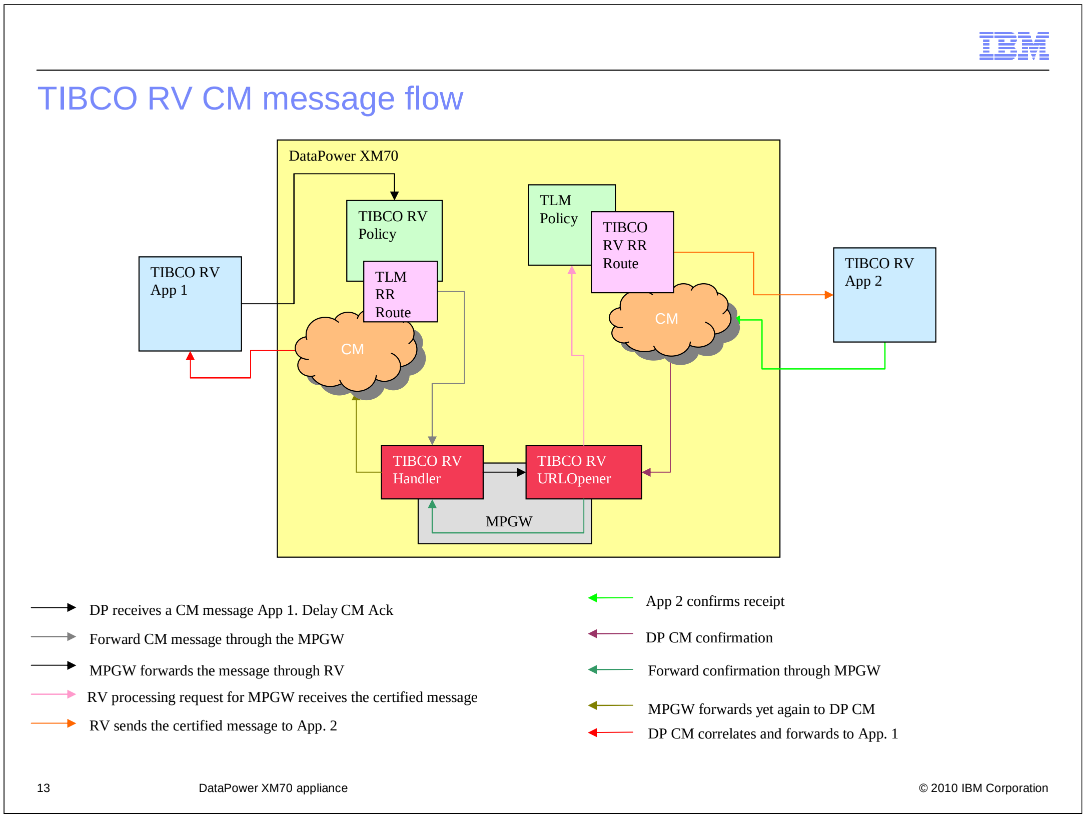
Request Old – specifies whether to request unacknowledged messages upon re-establishing the connection

Route:

Pre-register listener – allows CM transports which have request-old specified to receive all message transmitted before a CM agreement

Disallow listener – Disallows the specified name to register a CM delivery agreement between a listener and the RV route object

Message timeout limit – The interval of time allowed to certify message delivery. After the time expires, certified message delivery is not performed for the message.



TIBCO RV CM message flow

→ DP receives a CM message App 1. Delay CM Ack

→ Forward CM message through the MPGW

→ MPGW forwards the message through RV

→ RV processing request for MPGW receives the certified message

→ RV sends the certified message to App. 2

← App 2 confirms receipt

← DP CM confirmation

← Forward confirmation through MPGW

← MPGW forwards yet again to DP CM

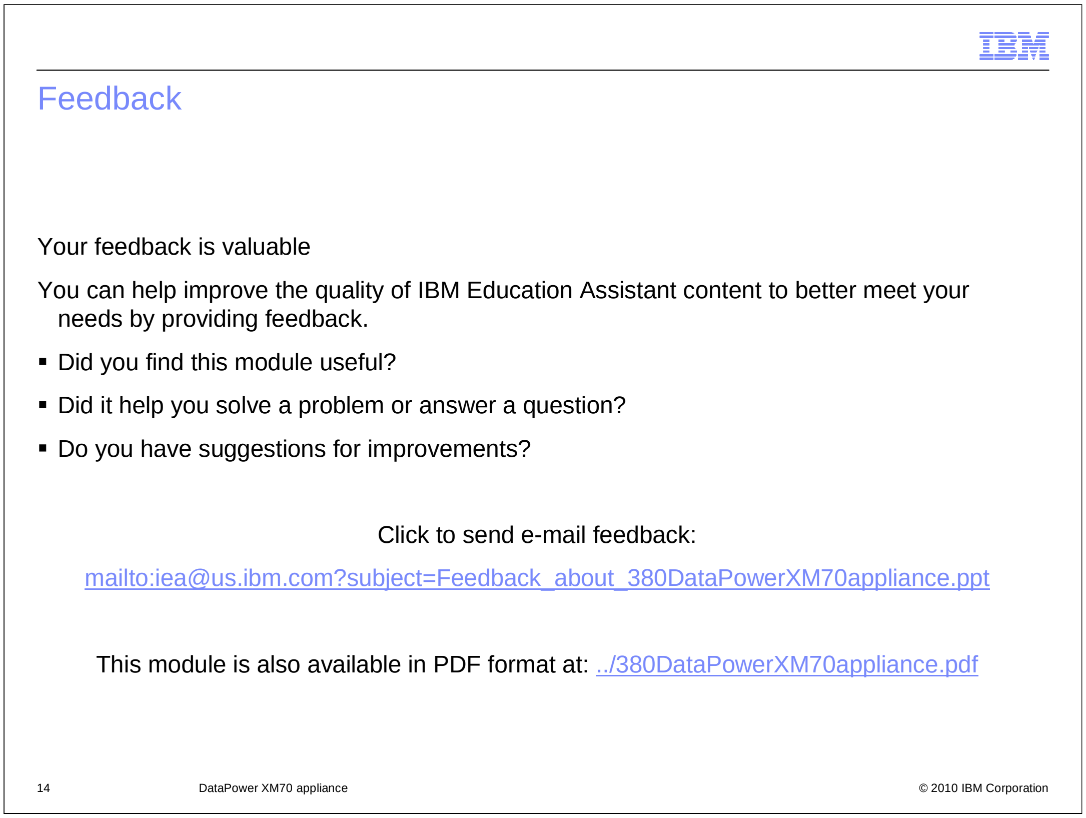
← DP CM correlates and forwards to App. 1

13 DataPower XM70 appliance

© 2010 IBM Corporation

This is the block flow diagram for a MPGW certified message transaction. Notice how it's quite a bit more complex than a normal message transaction.

Besides the normal messages flowing from left to right, you can see the CM confirmation messages flowing back from right to left.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

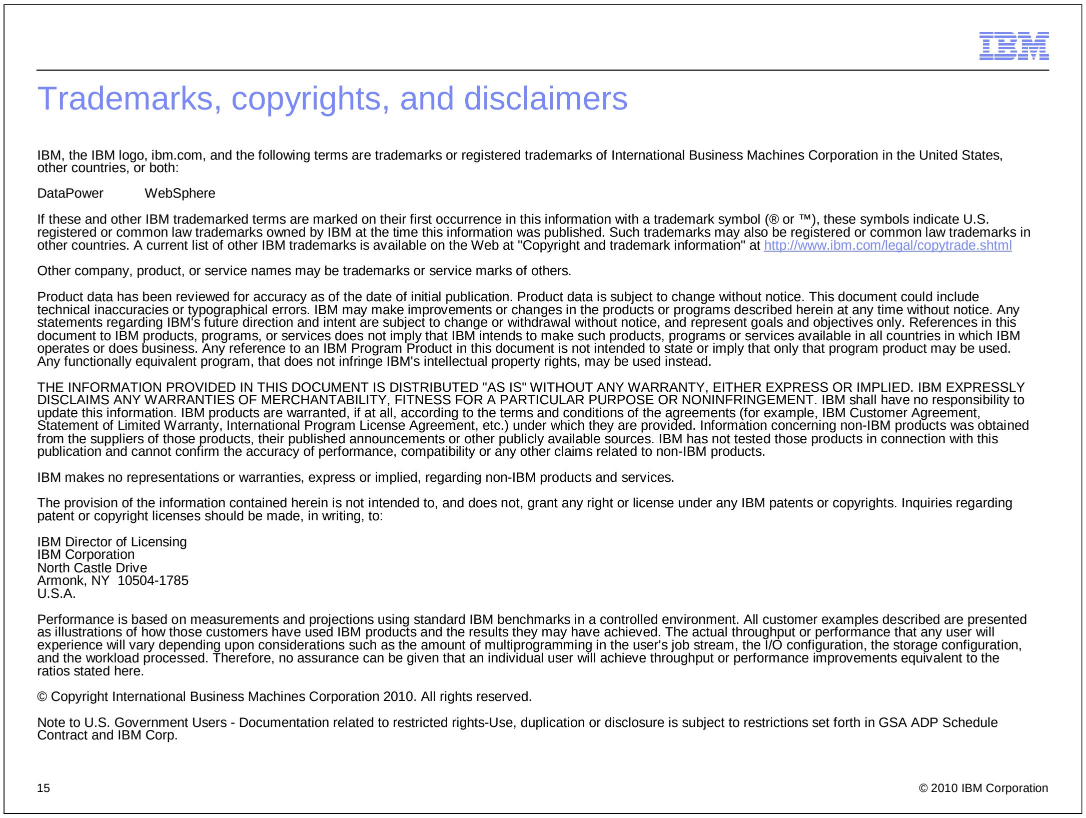
Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_380DataPowerXM70appliance.ppt

This module is also available in PDF format at: ../380DataPowerXM70appliance.pdf

© 2010 IBM Corporation

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

DataPower WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2010. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.