Using Transaction Tracing to Determine Issues with Remote MQ Transactions

Richard Nikula
VP, Product Development and Support
INTRODUCTION
Introduction

- Richard Nikula
  - VP of Product Development and Support

- Developing software for management of “middleware” since 1985
  - MAINVIEW for CICS, MQ
  - BMC PATROL
  - IBM CICSPLEX System Manager (CPSM)

- Involved in “MQ” since early 90’s
  - Primarily at the technology layer
  - Various certifications

- About Nastel Technologies
  - Founded in 1994
  - Middleware-centric Application Performance Management software supplier
  - Core competency: Messaging Middleware, Java Application Servers, ESB's and other SOA technologies
Overview

- MQ transactions can run on a number of different platforms and locations. They typically interact with other environments such as IBM Integration Bus (Broker) and DataPower. It can be challenging to track the flow of activities in these environments.

- In this session, you will learn:
  - The facilities provided by MQ for tracking MQ activity without changing application code
  - The facilities provide by IIB/Broker for tracking activity within message flows
  - Techniques for correlating data between broker and MQ for cross platform visibility
  - How similar techniques could be used for extending tracking to application servers, DataPower or other platforms
PRESENTATION...
Time Travel

1993
Your workstation

Operating system PC DOS 4.01
CPU Intel 80386SX @ 16 MHz
Memory 2 MB ~ 6 MB

7 color
Reverse video and blink
Graphical display capable
Your Network

Token

HERES YOUR PROBLEM. THE CONNECTION TO THE NETWORK IS BROKEN.

UH-OH. IT'S A "TOKEN RING" LAN. THAT MEANS THE TOKEN FELL OUT AND IT'S IN THIS ROOM SOMEPLACE.

YOU ARE THE WIND BENEATH MY WINGS.

I'LL WAIT A WEEK THEN TELL HIM THE TOKEN MUST BE IN THE "ETHERNET."

Copyright © 1996 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited.
Your Enterprise
Your remote network

Network Speed: 2400/4800/9600 Baud
Reliability?
MQ Timeline from 1993 to Today

1993 1.0
1995 2.0
1997 5.0
2002 2005 5.3
2008 6.0
2011 2012 7.1
2014 2015 7.5
2015 M2000
2014 8.0

1997 5.0

Because we wouldn’t have had this discussion in the past. Today’s environments make remote operation of MQ a requirement.
CONCEPTS OF REMOTE MANAGEMENT
You are remote to the system being managed

Not Remote Desktop or Telnet

No install of software on the Server1 (M2000 doesn’t allow)
Types of Remote Access

- **Application**
  - **The reason MQ exists**
    - But new considerations in a fully remote environment

- **Administrative**
  - **Configuring MQ**
    - Traditional MQ Explorer or similar 3rd party tools

- **Diagnostic**
  - **Looking at queue managers, queues and messages**
    - Traditional MQ Explorer or similar 3rd party tools
  - **Tracking messages**
    - New tools are needed
Primary Consideration for Remote Access

- **Connectivity**
  - Have to have access
    - Firewalls and network

- **Security**
  - Must be authorized to perform the actions requested
    - Access control (e.g. setmqaut)
    - AMS (Advanced Message Security)
    - SSL (communication)
    - Channel Authorization (V7.1 and higher)
    - Connection Authentication (V8 and higher)

- **Performance**
  - Has to be able to provide the bandwidth required

- **Not always the right choice**
ADMINISTRATION
Remote Management Example

This slide was added to show how easily remote management is. This was an MQ appliance being demonstrated by IBM at the conference to which they provided access. It shows that the advantage of remote access is how easily it can be done and the disadvantage is how easily it can be done.
TRACKING USING
APPLICATION ACTIVITY
TRACE
Application Activity Trace

- Creates an event message for MQ calls
  - Similar to using MQ API before/after exits

- Introduced in MQ 7.1
  - Expanded with MQ Appliance
  - Continuing improvements being made in V8

- Not available for MQ on zOS 😞
Using in MQ V7.1, 7.5 and V8*

- Configure mqat.ini (to add application)
- Execute Application to be analyzed
- Configure mqat.ini
- Configure mqat.ini (to remove application)
  - If running, change something in QMGR
- Data captured on SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE
- View Data
  - amqsact
  - Amqsactz (“freeware”)
  - Ms0P
  - 3rd Party tooling
Global Monitoring

Alternately, turn on/off Activity trace for “everything”
- Alter QMGR ACTVTRC(ON/OFF)

Change mqat.ini to exclude “everything” by default
- Applications that you don’t want
- Monitoring tools
- IBM tools (amqsact)
**MQAT.INI**

`#*******************************************************************#
#* Module Name: mqat.ini  *
#* Type       : WebSphere MQ queue manager configuration file     *
#* Function   : Define the configuration of application activity  *
#*             : trace for a single queue manager.                *
#*******************************************************************#`

AllActivityTrace: # Global settings stanza
  ActivityInterval=1 # Time interval between trace messages
  #   Values: 0-99999999 (0=off)
  #   Default: 0
  ActivityCount=100  # Number of operations between trace msgs
  #   Values: 0-99999999 (0=off)
  #   Default: 0
  TraceLevel=MEDIUM   # Amount of data traced for each operation
  #   Values: LOW | MEDIUM | HIGH
  #   Default: MEDIUM
  TraceMessageData=0  # Amount of message data traced
  #   Values: 0-104857600
  #   Default: 0
  StopOnGetTraceMsg=ON # Stop trace on get of activity trace message
  #   Values: ON | OFF
  #   Default: ON`
MQAT.INI (application specific)

### Specific application activity trace entry from generating data

- **ApplicationTrace:**
  - **ApplClass=ALL**
    - Values: (USER | MCA | ALL)
    - Default: USER
  - **ApplName=amqsput**
    - (matched to app name without path)
    - Default: *
  - **ApplFunction=***
    - (matched to app function)
    - Default: *
  - **Trace=ON**
    - Values: (ON | OFF)
    - Default: OFF
  - **ActivityInterval=0**
    - Values: 0-99999999 (0=off)
    - Default: 0
  - **ActivityCount=0**
    - Values: 0-99999999 (0=off)
    - Default: 0
  - **TraceLevel=HIGH**
    - Values: LOW | MEDIUM | HIGH
    - Default: MEDIUM
  - **TraceMessageData=1000**
    - Values: 0-104857600
    - Default: 0
MQAT.INI (application specific)

# Prevent the sample activity trace program from generating data#

ApplicationTrace:  # Application specific settings stanza
   ApplClass=ALL   # Application type
      # Values: (USER | MCA | ALL)
      # Default: USER
   ApplName=amqsact*  # Application name (may be wildcarded)
      # (matched to app name without path)
      # Default: *
   ApplFunction=*  # Application function (may be wildcarded)
      # (matched to app function)
      # Default: *
   Trace=OFF  # Activity trace switch for application
      # Values: ( ON | OFF )
      # Default: OFF
   ActivityInterval=0  # Time interval between trace messages
      # Values: 0-999999999 (0=off)
      # Default: 0
   ActivityCount=0  # Number of operations between trace msgs
      # Values: 0-999999999 (0=off)
      # Default: 0
   TraceLevel=MEDIUM  # Amount of data traced for each operation
      # Values: LOW | MEDIUM | HIGH
      # Default: MEDIUM
   TraceMessageData=0  # Amount of message data traced
      # Values: 0-104857600
      # Default: 0
Usage: amqsact

[-m QMgrName]  # Queue manager to connect to
[-q QName]     # Override default queue name
[-t TopicString]  # Subscribe to event topic
[-b]           # Only browse records
[-v]           # Verbose output
[-d <depth>]   # Number of records to display
[-w <timeout>] # Time to wait (in seconds)
[-s <startTime>] # Start time of record to process
[-e <endTime>]  # End time of record to process

Example:

amqsact -m QMGR1 -b -v
### Sample AMQSACT amqsact –m LocalQM1 –v (edited)

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MonitoringType</td>
<td>MQI Activity Trace</td>
</tr>
<tr>
<td>Correl_id</td>
<td>000000000: 414D 5143 4C6F 6361 6C51 4D31 2020 2020 'AMQCLocalQM1....'</td>
</tr>
<tr>
<td>Host Name</td>
<td>'RNIKULA-PC'</td>
</tr>
<tr>
<td>CommandLevel</td>
<td>800</td>
</tr>
<tr>
<td>SeqNumber</td>
<td>1</td>
</tr>
<tr>
<td>ApplicationName</td>
<td>'bSphere MQ\bin64\amqsput.exe'</td>
</tr>
<tr>
<td>Application Type</td>
<td>MQAT_WINDOWS_NT</td>
</tr>
<tr>
<td>Application Pid</td>
<td>5004</td>
</tr>
<tr>
<td>UserId</td>
<td>'Richard'</td>
</tr>
<tr>
<td>Pointer size</td>
<td>8</td>
</tr>
<tr>
<td>Platform</td>
<td>MQPL_WINDOWS_NT</td>
</tr>
<tr>
<td>Operation Id</td>
<td>MQXF_PUT</td>
</tr>
<tr>
<td>Operation Date</td>
<td>'2015-05-31'</td>
</tr>
<tr>
<td>Operation Time</td>
<td>'17:21:46'</td>
</tr>
<tr>
<td>High Res Time</td>
<td>1433110906406803</td>
</tr>
<tr>
<td>QMgr Operation Duration</td>
<td>70</td>
</tr>
<tr>
<td>Completion Code</td>
<td>MQCC_OK</td>
</tr>
<tr>
<td>Reason Code</td>
<td>0</td>
</tr>
<tr>
<td>Msg length</td>
<td>28</td>
</tr>
<tr>
<td>Object_type</td>
<td>MQOT_Q</td>
</tr>
<tr>
<td>Object_name</td>
<td>'Q1'</td>
</tr>
<tr>
<td>Object_Q_mgr_name</td>
<td>''</td>
</tr>
<tr>
<td>Resolved_Q_Name</td>
<td>'Q1'</td>
</tr>
<tr>
<td>Resolved_Q_mgr</td>
<td>'LocalQM1'</td>
</tr>
<tr>
<td>Resolved_local_Q_name</td>
<td>'Q1'</td>
</tr>
<tr>
<td>Resolved_local_Q_mgr</td>
<td>'LocalQM1'</td>
</tr>
<tr>
<td>Resolved_type</td>
<td>MQOT_Q</td>
</tr>
<tr>
<td>Report Options</td>
<td>0</td>
</tr>
<tr>
<td>Msg_type</td>
<td>MQMT_DATAGRAM</td>
</tr>
<tr>
<td>Expiry</td>
<td>-1</td>
</tr>
<tr>
<td>Format_name</td>
<td>'MQSTR'</td>
</tr>
<tr>
<td>Priority</td>
<td>-1</td>
</tr>
<tr>
<td>Persistence</td>
<td>2</td>
</tr>
</tbody>
</table>
MQCFH (PCF Header)

Use this page to view the PCF values contained by the MQCFH structure for an activity trace message.

For an activity trace message, the MQCFH structure contains the following values:

**Type**

*Description:* Structure type that identifies the content of the message.
*Data type:* MQLONG.
*Value:* MQCFT_APP_ACTIVITY

**StrucLength**

*Description:* Length in bytes of MQCFH structure.
*Data type:* MQLONG.
*Value:* MQCFH_STRUC_LENGTH

**Version**

*Description:* Structure version number.
*Data type:* MQLONG.
*Values:* MQCFH_VERSION_3

**Command**

*Description:* Command identifier. This field identifies the category of the message.
*Data type:* MQLONG.
*Values:* MQCMD_ACTIVITY_TRACE

**MsgSeqNumber**

*Description:* Message sequence number. This field is the sequence number of the message within a group of messages.
*Data type:* MQLONG.
*Values:* 1

**Control**

*Description:* Control options.
*Data type:* MQLONG.
*Values:* MQCFC_LAST

**ComptCode**

*Description:* Completion code.
*Data type:* MQLONG.
*Values:* MQCC_OK

**Reason**

*Description:* Reason code qualifying completion code.
*Data type:* MQLONG.
*Values:* MQRC_NONE

**ParameterCount**

*Description:* Count of parameter structures. This field is the number of parameter structures that follow the MQCFH structure.
*A group structure (MQCFGR), and its included parameter structures, are counted as one structure only.*
*Data type:* MQLONG.
*Values:* 1 or greater
Notes

- What are you Tracing?
  - If you turn on at the Queue Manager Level, most applications
  - Unless application connects using MQCONNX using
    - MQCNO_ACTIVITY_TRACE_DISABLED
  - Need to edit MQAT.INI to specific applications
    - How to know which applications read which queues?

- Who can view the trace?
  - Anyone with access to the queue
  - Information mixed with other users
  - Other users could remove your information
  - Tricky to get just your information
Using in MQ Appliance using V8*

- Configure mqat.ini to change defaults
- Use Dynamic mode to collect trace data
- View Data
  - amqsact (c)
  - Ms0P
  - 3rd Party tooling
M2000(mqcli)# dspmqini -m MQAPP_QM1
AllActivityTrace:
   ActivityInterval   = 1
   ActivityCount      = 100
   TraceLevel         = MEDIUM
   TraceMessageData   = 0
   StopOnGetTraceMsg  = ON
   SubscriptionDelivery = BATCHED
Setmqini

M2000(mqcli)# setmqini
Usage: setmqini -m QMgrName -s StanzaName -k KeyName [-d | -v Value]

- d   Delete the attribute.
- k   Key name of attribute to set.
- m   Queue manager name.
- s   Stanza name.
- v   Attribute value to set.

Example:
  setmqini -m QMAPP_QM1 -s AllActivityTrace -k TraceLevel -v HIGH
Use Case for Application Activity Tracing (V8*)

- Opens up new use cases
  - Traditionally tracing an administration function
  - Now can be done at individual developer level
Amqsact(c)

Usage: amqsact(c)

[-m QMgrName] # Queue manager to connect to
[-a ApplName] # Name of application to trace
[-c ChannelName] # Name of channel to trace
[-i ConnId] # Unique connection id to trace
[-q QName] # Override default queue name
[-t TopicString] # Subscribe to event topic
[-b] # Only browse records
[-v] # Verbose output
[-d <depth>] # Number of records to display
[-w <timeout>] # Time to wait (in seconds)
[-s <startTime>] # Start time of record to process
[-e <endTime>] # End time of record to process

Example:
amqsact -m QMGR1 -w 30 -a amqsput.exe
Dynamic Subscription

amqsact -mLocalQM1 -w 60 -a amqsput.exe

Subscribing to the activity trace topic:
'SYS/MQ/INFO/QMGR/LocalQM1/ActivityTrace/AppIName/amqsput.exe'

<table>
<thead>
<tr>
<th>General</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscription name:</td>
</tr>
<tr>
<td>Topic</td>
</tr>
<tr>
<td>Topic name:</td>
</tr>
<tr>
<td>Topic string:</td>
</tr>
<tr>
<td>Wildcard usage:</td>
</tr>
<tr>
<td>Scope:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination class: Managed</td>
</tr>
<tr>
<td>Destination queue manager: LocalQM1</td>
</tr>
<tr>
<td>Destination name: SYSTEM.MANAGED.NDURABLE.556B867103220020</td>
</tr>
<tr>
<td>Correlation identifier:</td>
</tr>
</tbody>
</table>
Amqsact(c) remote access

```
amqsactc -m RemoteQM1 -w 60 -a amqsputc.exe -v
```

Subscribing to the activity trace topic:
```
'$SYS/MQ/INFO/QMGR/RemoteQM1/ActivityTrace/AplName/amqsputc.exe'
```

```
MonitoringType: MQI Activity Trace
Correl_id:
00000000: 414D 5120 5265 6D6F 7465 514D 3120 2020 'AMQ.RemoteQM1..'
00000010: 0D4D 6B55 1023 0020 '.MkU.#..
QueueManager: 'RemoteQM1'
CommandLevel: 801
SeqNumber: 0
ApplicationName: 'Sphere MQ\bin64\amqsputc.exe'
Application Type: MQAT_WINDOWS_NT
ApplicationPid: 14666
UserId: 'mqm'
API Caller Type: MQXACT_EXTERNAL
API Environment: MQXE_MCA_SVRCONN
Channel Name: 'SYSTEM.ADMIN.SVRCONN'
ConnName: '192.168.75.1'
Channel Type: MQCHT_SVRCONN
```
Tracking activity from a Business Partner

```
amqsactc -m RemoteQM1 -w 60 -c From.BP0302.Primary -v
```

Subscribing to the activity trace topic:

'\$SYS/MQ/INFO/QMGR/RemoteQM1/ActivityTrace/ChannelName/From.BP0302.Primary'

---

MonitoringType: MQI Activity Trace
Correl_id:
00000000: 414D 5120 5265 6D6F 7465 514D 3120 2020 'AMQ.RemoteQM1...
00000010: 0D4D 6B55 0429 0020 '.MkU.)..
QueueManager: 'RemoteQM1'
ApplicationName: 'amqrmppa'
Application Type: MQAT_QMGR
ApplicationPid: 14666
UserId: 'mqsystem'
API Caller Type: MQXACT_INTERNAL
API Environment: MQXE_MCA
Channel Name: 'From.BP0302.Primary'
ConnName: '192.168.75.1'
Channel Type: MQCHT_RECEIVER
Operation 0

MQI Operation: 0
Operation Id: MQXF_CONNNX
ApplicationTid: 21
OperationDate: '2015-05-31'
OperationTime: '14:33:35'
ConnectionId:
00000000: 414D 5143 5265 6D6F 7465 514D 3120 2020  'AMQCRemoteQM1...' 
00000010: 0D4D 6B55 0128 0020                      '.MkU.(..   '
QueueManager: 'RemoteQM1'
QMgr Operation Duration: 96098215
Completion Code: MQCC_OK
Reason Code: 0
Connect Options: 1
Operation 1

MQI Operation: 1
Operation Id: MQXF_OPEN
ApplicationTid: 21
OperationDate: '2015-05-31'
OperationTime: '14:33:35'
Object_type: MQOT_Q
Object_name: 'Q1'
Object_Q_mgr_name: 'RemoteQM1'

QMGr Operation Duration: 74
Completion Code: MQCC_OK
Reason Code: 0
Open_options: 43024
Object_type: MQOT_Q
Object_name: 'Q1'
Object_Q_mgr_name: 'RemoteQM1'

Resolved_Q_Name: 'Q1'
Resolved_Q_mgr: 'RemoteQM1'
Resolved_local_Q_name: 'Q1'
Resolved_local_Q_mgr: 'RemoteQM1'
Resolved_type: MQOT_Q
Operation 2

MQI Operation: 2
Operation Id: MQXF_PUT
ApplicationTid: 21
OperationDate: '2015-05-31'
OperationTime: '14:33:35'
High Res Time: 1433097215499226
QMgr Operation Duration: 44
Completion Code: MQCC_OK
Reason Code: 0
Hobj: 2
Put Options: 272388
Msg length: 20
Object_Q_mgr_name: 'RemoteQM1'
Resolved_Q_Name: 'Q1'
Resolved_Q_mgr: 'RemoteQM1'
Resolved_local_Q_name: 'Q1'
Resolved_local_Q_mgr: 'RemoteQM1'
Resolved_type: MQOT_Q
Expiry: -1
Format_name: 'MQSTR'
Considerations

- How to Pair up put and get of the same messages?
- Dealing with complex message flows.
## Sample display

- **Transaction Flow Diagram**
- **Transaction Timeline**
- **Transaction Trace**
- **Transaction Milestones**

### Show Hierarchy

<table>
<thead>
<tr>
<th>Time</th>
<th>Application</th>
<th>Operation Name</th>
<th>Resource</th>
<th>Resource Manager</th>
<th>Elapsed Time (usec)</th>
<th>Completion Code</th>
<th>Reason Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-06-01 10:52:15.000</td>
<td>amqsput</td>
<td>MQPUT</td>
<td>To.PayRoll.Primary</td>
<td>LocalQM1</td>
<td>69</td>
<td>Succeeded</td>
<td>0</td>
</tr>
<tr>
<td>2015-06-01 10:52:18.000</td>
<td>WMQ_ChannelPooling...</td>
<td>MQPUT</td>
<td>PayRoll.Primary</td>
<td>RemoteQM1</td>
<td>68</td>
<td>Succeeded</td>
<td>0</td>
</tr>
<tr>
<td>2015-06-01 10:52:24.000</td>
<td>amqsgetc</td>
<td>MQGET</td>
<td>PayRoll.Primary</td>
<td>RemoteQM1</td>
<td>39</td>
<td>Succeeded</td>
<td>0</td>
</tr>
</tbody>
</table>
FTE/MFT
What is it?

Agent1

Agent2

Coordinator (Queue Manager)
Subscribing to events

- SYSTEM.FTE/Log/#

Types of data

- Agents
- Transfers
- Schedules
- Logs
MQ Explorer Views

Transfer Log

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Completion State</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT1</td>
<td>AGENT2</td>
<td>Failed</td>
<td>RNIKULA@192.168.244.1</td>
</tr>
<tr>
<td>AGENT1</td>
<td>MB8BROKER.FTEHQ</td>
<td>Failed</td>
<td>RNIKULA@172.30.97.48</td>
</tr>
<tr>
<td>AGENT1</td>
<td>AGENT2</td>
<td>Successful</td>
<td>RNIKULA@192.168.244.1</td>
</tr>
<tr>
<td>AGENT1</td>
<td>AGENT2</td>
<td>Failed</td>
<td>RNIKULA@192.168.244.1</td>
</tr>
<tr>
<td>AGENT1</td>
<td>AGENT2</td>
<td>Successful</td>
<td>RNIKULA@172.30.97.197</td>
</tr>
<tr>
<td>AGENT1</td>
<td>AGENT2</td>
<td>Failed</td>
<td>RNIKULA@172.30.97.197</td>
</tr>
<tr>
<td>AGENT1</td>
<td>AGENT2</td>
<td>Failed</td>
<td>RNIKULA@192.168.244.1</td>
</tr>
<tr>
<td>AGENT1</td>
<td>AGENT2</td>
<td>Successful</td>
<td>RNIKULA@172.30.97.197</td>
</tr>
<tr>
<td>AGENT1</td>
<td>AGENT2</td>
<td>Successful</td>
<td>RNIKULA@172.30.97.197</td>
</tr>
</tbody>
</table>
Sample log Entry

```xml
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="6.00" ID="414d5120434f52442020202020202020a59e525520118c24" agentRole="sourceAgent" xmlns="http://www.w3.org/2001/XMLSchema-instance">
    <action time="2015-05-13T14:08:07.568Z">completed</action>
    <sourceAgent agent="AGNT1A" QMgr="AGNT1" agentType="STANDARD">
        <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
    </sourceAgent>
    <destinationAgent agent="AGNT2A" QMgr="AGNT2"/>
    <originator>
        <hostName>192.168.188.1</hostName>
        <userID>user</userID>
        <mqmdUserID>user</mqmdUserID>
    </originator>
    <status resultCode="49">
        <supplement>BFGEL0013E: The transfer with id '414d5120434f52442020202020202020a59e525520118c23' has failed. The agent has received a reason code of '2087' from WebSphere MQ when sending a message to destination queue 'SYSTEM.FTE.COMMAND.AGNT2A' on Queue Manager 'AGNT2'.
        </supplement>
    </status>
    <transferSet startTime="2015-05-13T14:08:07.552Z" total="1" bytesSent="0">
    </transferSet>
    <job>
        <name>Job1</name>
    </job>
</transaction>
```
Starting

<?xml version="1.0" encoding="UTF-8"?>
<transaction ID="414d51204d4238514d4752202020201d07c9552030d113" agentRole="sourceAgent" ...>
  <action time="2015-08-13T23:59:33.917Z">started</action>
  <sourceAgent QMgr="MB8QMGR" agent="AGENT1" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="MB8QMGR" agent="AGENT2"/>
  <originator>
    <hostName>192.168.244.1</hostName>
    <userID>RNIKULA</.userID>
    <mqmdUserID>RNIKULA</mqmdUserID>
  </originator>
  <transferSet bytesSent="0" startTime="2015-08-13T23:59:33.917Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfте.SourceAgent">AGENT1</metaData>
      <metaData key="com.ibm.wmqfте.DestinationAgent">AGENT2</metaData>
      <metaData key="com.ibm.wmqfте.MqmdUser">RNIKULA</metaData>
      <metaData key="com.ibm.wmqfте.OriginatingUser">RNIKULA</metaData>
      <metaData key="com.ibm.wmqfте.OriginatingHost">192.168.244.1</metaData>
      <metaData key="com.ibm.wmqfте.TransferId">414d51204d4238514d4752202020201d07c9552030d113</metaData>
      <metaData key="com.ibm.wmqfте.ScheduleId">10</metaData>
      <metaData key="com.ibm.wmqfте.JobName">AAAAA</metaData>
      <metaData key="com.ibm.wmqfте.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
  <job>
    <name>AAAAA</name>
  </job>
</transaction>
<xml version="1.0" encoding="UTF-8">
<transaction ID="414d5120d4238514d4752202020201d07c9552030d113" agentRole="sourceAgent" ...>
  <action time="2015-08-13T23:59:34.198Z">progress</action>
  <sourceAgent QMgr="MB8QMGR" agent="AGENT1" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="MB8QMGR" agent="AGENT2" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </destinationAgent>
  <originator>
    <hostName>192.168.244.1</hostName>
    <userID>RNIKULA</userID>
    <mqmdUserID>RNIKULA</mqmdUserID>
  </originator>
  <transferSet bytesSent="14285" index="0" size="1" startTime="2015-08-13T23:59:33.917Z" total="1">
    <item mode="binary">
      <source disposition="leave" type="file">
        <file last-modified="2014-12-10T14:57:18.000Z" size="14237">C:\TEMP\setup.ini</file>
        <checksum method="MD5">dec47d003341090ce9007ab5eca38623</checksum>
      </source>
      <destination exist="overwrite" type="file">
        <file last-modified="2015-08-13T23:59:34.073Z" size="14237">C:\nastel2\setup.ini</file>
        <checksum method="MD5">dec47d003341090ce9007ab5eca38623</checksum>
      </destination>
      <status resultCode="0"/>
    </item>
  </transferSet>
</transaction>
<xml version="1.0" encoding="UTF-8"/>
<transaction ID="414d51204d4238514d4752202020201d07c9552030d113" agentRole="sourceAgent">
  <action time="2015-08-13T23:59:34.213Z">completed</action>
  <sourceAgent QMgr="MB8QMGR" agent="AGENT1" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="MB8QMGR" agent="AGENT2" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </destinationAgent>
  <originator>
    ...
    <mqmdUserID>RIKULA</mqmdUserID>
  </originator>
  <status resultCode="0">
    <supplement>BFGRP0032I: The file transfer request has successfully completed.</supplement>
  </status>
  <transferSet bytesSent="14285" startTime="2015-08-13T23:59:33.917Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">AGENT1</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT2</metaData>
      ...
      <metaData key="com.ibm.wmqfte.TransferId">414d51204d4238514d4752202020201d07c9552030d113</metaData>
      <metaData key="com.ibm.wmqfte.ScheduleId">10</metaData>
      <metaData key="com.ibm.wmqfte.JobName">AAAAA</metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
  <job>
    <name>AAAAA</name>
  </job>
  <statistics>
    <actualStartTime>2015-08-13T23:59:34.010Z</actualStartTime>
    <retryCount>0</retryCount>
    <numFileFailures>0</numFileFailures>
    <numFileWarnings>0</numFileWarnings>
  </statistics>
</transaction>
<?xml version="1.0" encoding="UTF-8"?>
<translation>
<transaction ID="414d51204d4238514d475220202020dd843f552004d50c" agentRole="sourceAgent" ...>
  <action time="2015-08-13T23:59:01.121Z">completed</action>
  <sourceAgent QMgr="MB8QMGR" agent="AGENT1" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="MB8QMGR" agent="AGENT2" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </destinationAgent>
  <originator>
    <hostName>192.168.244.1</hostName>
    <userID>RNIKULA</userID>
    <mqmdUserID>RNIKULA</mqmdUserID>
  </originator>
  <status resultCode="40">
    <supplement>BFRP0034I: The file transfer request has completed with no files being transferred.</supplement>
  </status>
  <transferSet bytesSent="0" startTime="2015-08-13T23:58:59.247Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">AGENT1</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT2</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">RNIKULA</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">RNIKULA</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">192.168.244.1</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d51204d4238514d475220202020dd843f552004d50c</metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
  <statistics>
    <actualStartTime>2015-08-13T23:59:00.700Z</actualStartTime>
    <retryCount>0</retryCount>
    <numFileFailures>1</numFileFailures>
    <numFileWarnings>0</numFileWarnings>
  </statistics>
</transaction>
</translation>
### Trace Details

#### Trace

<table>
<thead>
<tr>
<th>Start Date</th>
<th>Applications</th>
<th>Transaction Status</th>
<th>SLA Status</th>
<th>SLA Status Text</th>
<th>Workload (RTIME:MS:mm)</th>
<th>Transaction Duration</th>
<th>Operations</th>
<th>Messages</th>
<th>Transaction ID</th>
<th>Transaction Groups</th>
<th>Servers</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>015-05-13 10:08:08</td>
<td>AGNT1A:AGNT1</td>
<td>Failed</td>
<td>Within SLA</td>
<td>✗</td>
<td>0.00:00.000</td>
<td>0.00:00.016</td>
<td>2</td>
<td>0</td>
<td>96581</td>
<td></td>
<td>192.168.188.1</td>
<td>Mes</td>
</tr>
<tr>
<td>015-05-13 10:08:08</td>
<td>AGNT1A:AGNT1</td>
<td>Failed</td>
<td>Within SLA</td>
<td>✗</td>
<td>0.00:00.000</td>
<td>0.00:00.000</td>
<td>2</td>
<td>0</td>
<td>96574</td>
<td></td>
<td>192.168.188.1</td>
<td>Mes</td>
</tr>
</tbody>
</table>

#### Transaction ID 96581

<table>
<thead>
<tr>
<th>Time</th>
<th>Operation Name</th>
<th>Resource</th>
<th>Elapsed Time (usec)</th>
<th>Message Id</th>
<th>Message Age (usec)</th>
<th>Completion Code</th>
<th>Server</th>
<th>Resource Manager Type</th>
<th>Resource Manager</th>
<th>Application</th>
<th>Total Runtime (RTIME:MS:mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>105-05-13 10:08:08.552</td>
<td>started</td>
<td>JOB:CHEVREOUX</td>
<td>0</td>
<td>0</td>
<td></td>
<td>192.168.188.1</td>
<td>Messaging Server</td>
<td>AGNT1</td>
<td>AGNT1A:AGNT1</td>
<td>0:00:00.000</td>
<td></td>
</tr>
<tr>
<td>105-05-13 10:08:08.568</td>
<td>completed</td>
<td>JOB:CHEVREOUX</td>
<td>0</td>
<td>0</td>
<td></td>
<td>192.168.188.1</td>
<td>Messaging Server</td>
<td>AGNT1</td>
<td>AGNT1A:AGNT1</td>
<td>0:00:00.000</td>
<td></td>
</tr>
</tbody>
</table>
IIB (BROKER)
Type of Broker Management Data Available Remotely

- **Resource Statistics**
  - Resources used by execution groups

- **Monitoring Statistics**
  - Usage Statistics of execution groups, nodes and threads

- **Flow Tracking**
  - Tracking of execution flow through message flows
Broker Monitoring Statistics

- The broker provides detailed statistics

  - **Message Flow Level (for the flow)**
    - Total Messages Processed
    - Total Messages in Error
    - CPU Time Spent
    - Message Statistics

  - **Node Level (for each processing node)**
    - Invocations
    - CPU Time
    - Information

  - **Thread Level (for each thread)**
    - Messages Processed
    - CPU Time Spent
    - Message Statistics
Tracking within the Message Flows

- The Broker Supports Tracking within the Message Flows
  - Transaction Start / Stop (default)
  - See when a given node was processed
  - See details about the message being processed by the flow
  - Track message flows in and across brokers

- Activated at the Message Flow and Node Level
  - Whether to collect
  - Data to Collect
## Configuring Message Flow Events

### Default Values for Message Flow Properties - SurfWatch

Configure monitoring events.

3 events defined. Events are defined via the Monitoring tab of a selected node in the message flow.

<table>
<thead>
<tr>
<th>Enabled</th>
<th>Node</th>
<th>Event Source</th>
<th>Event Source Address</th>
<th>Event Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Publish Reports</td>
<td>In terminal</td>
<td>Publish Reports.terminal.in</td>
<td>Publish Reports.InTerminal</td>
</tr>
<tr>
<td></td>
<td>SURFWATCH</td>
<td>Transaction start</td>
<td>SURFWATCH.transaction.Start</td>
<td>SURFWATCH.TransactionStart</td>
</tr>
<tr>
<td></td>
<td>SURFWATCH_FAIL</td>
<td>In terminal</td>
<td>SURFWATCH_FAIL.terminal.Exit</td>
<td>SURFWATCH_FAIL.InTerminal</td>
</tr>
</tbody>
</table>
Event Attributes – Basic

Event Source
Select the source of the event.
Transaction end

Event Source Address
The broker identifies an event source using an event source address. Use this value when you enable and disable event sources using runtime commands.
SURFWATCH:TransactionEnd

Event Name
Provide the name by which events emitted from this source are to be known. Specify either a literal name, or the location of a character field in the message tree or elsewhere in the message assembly.

Event Filter
Provide an expression to control whether the event is emitted. The expression must evaluate to true or false, and can reference fields in the message tree or elsewhere in the message assembly.
If you do not specify a value, the value true() is used.
true()

Event Payload
Most events need to contain data taken from fields in the message tree or from elsewhere in the message assembly. Data taken from simple fields or complex fields appears in the event in XML character format. An event can also contain bitstream data, which appears in the event as hexadecimal bytes.

Data location

Include bitstream data in payload

Content:  Encoding:  

Copyright © 2015 Nastel Technologies, Inc.
Event Attributes – Event Source

- Event Source
  - Select the source of the event.
  - Transaction end

- Event Source Address
  - The broker identifies an event source and can disable event sources using runtime.
  - SURFWATCH.transaction.End

- Event Name
  - Provide the name by which events can be identified, or the location of a character.
  - Literal: SURFWATCH.TransactionEnd

- Event Filter
  - Provide an expression to control whether the event is emitted. The expression must evaluate to true or false, and can reference fields in the message tree or elsewhere in the message assembly.
  - If you do not specify a value, the value true() is used.
  - true()

- Event Payload
  - Most events need to contain data taken from fields in the message tree or from elsewhere in the message assembly. Data taken from simple fields or complex fields appears in the event in XML character format. An event can also contain bitstream data, which appears in the event as hexadecimal bytes.
  - Data location

- Include Bitstream data in payload
  - Content: 
  - Encoding: 

Copyright © 2015 Nastel Technologies, Inc.
Event Attributes – Event Name

- **Event Name**: Provide the name by which events emitted from this source are to be known. Specify either a literal name, or the location of a character field in the message tree or elsewhere in the message assembly.
  - **Literal**: SURFWATCH.TransactionEnd
  - **Data location**: 

- **Event Filter**: Provide an expression to control whether the event is emitted. The expression must evaluate to true or false, and can reference fields in the message tree or elsewhere in the message assembly.
  - If you do not specify a value, the value true() is used.

- **Event Payload**: Most events need to contain data taken from fields in the message tree or from elsewhere in the message assembly. Data taken from simple fields or complex fields appears in the event in XML character format. An event can also contain bitstream data, which appears in the event as hexadecimal bytes.
  - **Data location**: 
  - **Include bitstream data in payload**: 
    - **Content**: 
    - **Encoding**: 

Event Attributes – Dynamic Event Name

XPath Expression Builder

Select the target from the Schema viewer or Operator viewer and drag and drop the nodes in the source viewer below.

Data Types Viewer

- $Root
- $Properties
- $LocalEnvironment
- $DestinationList
- $ExceptionList
- $Environment

Operators

| | /
<p>| &lt;= |
| &lt; |
| &gt;= |
| &gt; |
| = |
| != |
| and |
| or |</p>
<table>
<thead>
<tr>
<th>+</th>
</tr>
</thead>
</table>

XPath Expression

[Diagram of XPath Expression]

Checkbox: Show XML Schema groups

Namespace settings

Copyright © 2015 Nastel Technologies, Inc.
Event Attributes – Event Filter

Event Source
Select the source of the event.
Transaction end

Event Source Address
The broker identifies an event source using an event source address. Use this value when you enable and disable event sources using runtime commands.
SURFWATCH:TransactionEnd

Event Name
Provide the name by which events emitted from this source are to be known. Specify either a literal name, or the location of a character field in the message tree or elsewhere in the message assembly.

Event Filter
Provide an expression to control whether the event is emitted. The expression must evaluate to true or false, and can reference fields in the message tree or elsewhere in the message assembly. If you do not specify a value, the value true is used.
true()
Event Attributes – Event Payload

Event Source
Select the source of the event:
Transaction end

Event Source Address
The broker identifies an event source using an event source address. Use this value when you enable and disable event sources using runtime commands.
SURFWATCH.TransactionEnd

Event Name
Provide the name by which events emitted from this source are to be known. Specify either a literal name, or the location of a character field in the message tree or elsewhere in the message assembly.

- Literal: SURFWATCH.TransactionEnd
- Data location: [field location]

Event Filter
Provide an expression to control whether the event is emitted. The expression must evaluate to true or false, and can reference fields in the message tree or elsewhere in the message assembly. If you do not specify a value, the value true() is used.

true()

Event Payload
Most events need to contain data taken from fields in the message tree or from elsewhere in the message assembly. Data taken from simple fields or complex fields appears in the event in XML character format. An event can also contain bitstream data, which appears in the event as hexadecimal bytes.

Data location

- [Add, Edit, Delete]

Include bitstream data in payload
- Content: [field location]
- Encoding: [field location]
Event Attributes - Correlation

Event Correlation
A monitoring application uses event correlators to match events emitted by the same, or related, business transactions. A local transaction correlator links the events emitted by a single invocation of a message flow. A parent transaction correlator links the events from a message flow to a parent message flow or an external application. A global transaction correlator links events from a message flow to one or more related message flows or external applications. An event must contain a local transaction correlator, but need not contain a parent transaction correlator or global transaction correlator.

Local transaction correlator:
- Automatic
- Specify location of correlator

Description
The local correlator used by the most recent event for this message flow invocation will be used. If no local correlator exists yet, a new unique value will be generated.

Parent transaction correlator:
- Automatic
- Specify location of correlator

Description
The parent correlator used by the most recent event for this message flow invocation will be used. If no correlator exists yet, no parent correlator will be used.

Global transaction correlator:
- Automatic
- Specify location of correlator

Description
The global correlator used by the most recent event for this message flow invocation will be used. If no correlator exists yet, no global correlator will be used.
mqsichangeflowmonitoring Examples

- mqsichangeflowmonitoring BROKER -c active -g -j
  - > Activate event flow monitoring for all execution groups and flows

- mqsichangeflowmonitoring BROKER -c active -e default -k application1 -f myflow1
  - > Activate monitoring for message flow *myflow1* referenced by application *application1*, in execution group *default*

- mqsichangeflowmonitoring BROKER -c inactive -g -j
  - > Deactivate event flow monitoring for everything
Getting Tracking Data

- The statistics Tracking data is published

  - $SYS/Broker/<brokerName>/Monitoring/<executionGroupName>/<flowName>

- Example Subscriptions

  - $SYS/Broker/Broker1/Monitoring/#
  - $SYS/Broker/+/Monitoring/#
  - $SYS/Broker/Broker2/Monitoring/EGRP/Flow1

The resultant data is then processed directly by a subscribing application or placed on a queue for processing by an application.
<wmb:event
xmlns:wmb="http://www.ibm.com/xmlns/prod/websphere/messagebroker/6.1.0/monitoring/event">
<wmb:eventPointData>
<wmb:eventData wmb:productVersion="8001" wmb:eventSchemaVersion="6.1.0.2">
<wmb:eventIdentity wmb:eventName="transactionStart" />
<wmb:eventSequence wmb:creationTime="2015-09-25T21:06:10.008Z" wmb:counter="1" />
<wmb:eventCorrelation wmb:localTransactionId="414d51204d4238514d475220202020bf172454201558fe" wmb:parentTransactionId="" wmb:globalTransactionId="" />
</wmb:eventData>
<wmb:messageFlowData>
<wmb:broker wmb:name="MB8BROKER" wmb:UUID="61f8eda0-81f5-43b6-8cf5-b9a1fef8f91b" />
<wmb:executionGroup wmb:name="PagerExecutionGroup" wmb:UUID="a4f0ff6-4501-0000-0080-c644e460ccff" />
<wmb:messageFlow wmb:uniqueFlowName="MB8BROKER.PagerExecutionGroup.SurfWatch" wmb:name="SurfWatch" wmb:UUID="6c0000f7-4501-0000-0080-d6b3e1d5c115" wmb:threadId="10044" />
<wmb:node wmb:nodeLabel="SURFWATCH" wmb:nodeType="ComIbmMQInputNode" wmb:detail="SURFWATCH" />
</wmb:messageFlowData>
</wmb:event>
Tracking Data

<wmb:event xmlns:wmb="http://www.ibm.com/xmlns/prod/websphere/messagebroker/6.1.0/monitoring/event">
  <wmb:eventPointData>
    <wmb:eventData wmb:productVersion="8001" wmb:eventSchemaVersion="6.1.0.2">
      <wmb:eventIdentity wmb:eventName="transactionEnd"/>
      <wmb:eventCorrelation wmb:localTransactionId="414d51204d4238514d47522020202020bf172454201558fe " wmb:parentTransactionId="" wmb:globalTransactionId=""/>
    </wmb:eventData>
    <wmb:messageFlowData>
      <wmb:broker wmb:name="MB8BROKER" wmb:UUID="61f8eda0-81f5-43b6-8cf5-b9a1fef8f91b"/>
      <wmb:executionGroup wmb:name="PagerExecutionGroup" wmb:UUID="a4f0fff6-4501-0000-0080-c644e460ccff"/>
      <wmb:messageFlow wmb:uniqueFlowName="MB8BROKER.PagerExecutionGroup.SurfWatch" wmb:name="SurfWatch" wmb:UUID="6c0000f7-4501-0000-0080-d6b3e1d5c115" wmb:threadId="10044"/>
      <wmb:node wmb:nodeLabel="SURFWATCH" wmb:nodeType="ComIbmMQInputNode" wmb:detail="SURFWATCH"/>
    </wmb:messageFlowData>
  </wmb:eventPointData>
</wmb:event>
Example

- Message Flows can be tracked by capturing the flow tracking events.
SUMMARY
Other Transaction Monitoring

1. PROCESSING POINT
   - IBM Data Power
   - IBM WBI (IIB)
   - IBM MFT (FTT)
   - WebSphere MQ
   - Java

2. AutoPilot® M6 TransactionWorks® ANALYZER
   - CEP
   - Transaction analysis

3. AutoPilot® M6 TransactionWorks® EXPLORER
   - Transaction exploration

GLOBAL TRANSACTION
Cross Application Tracking

<table>
<thead>
<tr>
<th>Time</th>
<th>Operation Name</th>
<th>Resource</th>
<th>Elapsed Time (usec)</th>
<th>Message Id</th>
<th>Message Age (ussec)</th>
<th>Completion Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-09-26 13:41:47.17</td>
<td>/trading/verification</td>
<td>/trading/verification</td>
<td>7875</td>
<td>0</td>
<td>Succeeded</td>
<td>L</td>
</tr>
<tr>
<td>2013-09-26 13:41:47.19</td>
<td>com/acme/trading/verify,start</td>
<td>java</td>
<td>6749</td>
<td>0</td>
<td>Succeeded</td>
<td>L</td>
</tr>
<tr>
<td>2013-09-26 13:41:47.19</td>
<td>HTTP/GET/trading/verification</td>
<td>/trading/verification</td>
<td>7665</td>
<td>2195</td>
<td>Success</td>
<td>L</td>
</tr>
<tr>
<td>2013-09-26 13:41:47.19</td>
<td>Statement.executeQuery(String)</td>
<td>MySQL5.1.0DB3en</td>
<td>1823</td>
<td>2196</td>
<td>Success</td>
<td>L</td>
</tr>
<tr>
<td>2013-09-26 13:41:47.20</td>
<td>com/acme/trading/verify/request java</td>
<td>1495</td>
<td>0</td>
<td>Succeeded</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td>2013-09-26 13:41:47.20</td>
<td>QueueSender.send</td>
<td>queue:/trading/</td>
<td>6565</td>
<td>2186</td>
<td>Success</td>
<td>L</td>
</tr>
<tr>
<td>2013-09-26 13:41:47.20</td>
<td>MQPUT</td>
<td>TradeVerification.</td>
<td>10107</td>
<td>2186</td>
<td>Success</td>
<td>L</td>
</tr>
<tr>
<td>2013-09-26 13:41:47.22</td>
<td>MQGET</td>
<td>TradeVerification.</td>
<td>10900</td>
<td>2186</td>
<td>Success</td>
<td>L</td>
</tr>
<tr>
<td>2013-09-26 13:41:47.22</td>
<td>MQPUT</td>
<td>TradeValidation.1</td>
<td>9099</td>
<td>2107</td>
<td>Success</td>
<td>D</td>
</tr>
<tr>
<td>2013-09-26 13:41:47.24</td>
<td>MQGET</td>
<td>TradeValidation.1</td>
<td>9507</td>
<td>2187</td>
<td>Success</td>
<td>Y</td>
</tr>
<tr>
<td>2013-09-26 13:41:47.24</td>
<td>MQPUT</td>
<td>TradeValidation.1</td>
<td>9803</td>
<td>2188</td>
<td>Success</td>
<td>Y</td>
</tr>
<tr>
<td>2013-09-26 13:42:23.22</td>
<td>MQGET</td>
<td>TradeValidation.1</td>
<td>05970000</td>
<td>2188</td>
<td>Success</td>
<td>B</td>
</tr>
</tbody>
</table>
Additional Information

- For additional information on products provided by Nastel including AutoPilot services for IBM MQ, Integration Bus, Managed File Transfer, DataPower and WebSphere Application Server

- See http://www.nastel.com

- Or contact info@nastel.com

- Thank You
Questions