

WebSphere MQ File Transfer Edition – Administration and Topologies

Chris Harris
WMQFTE L3 Team Lead
IBM Hursley
harriscr@uk.ibm.com



SHARE in Boston

Agenda

- Recap
- Configuration
- Message paths
- Accessing Information about WebSphere MQ File Transfer Edition
- Resilience
- Example Topologies
- Demo

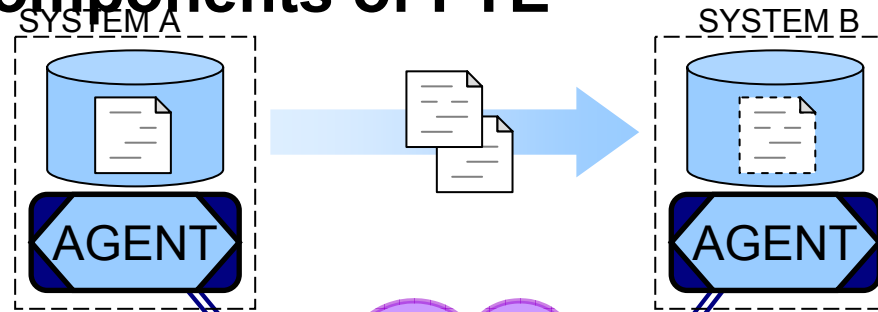


Recap

Recap: 3 Main Components of FTE

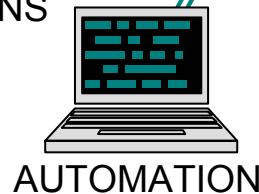
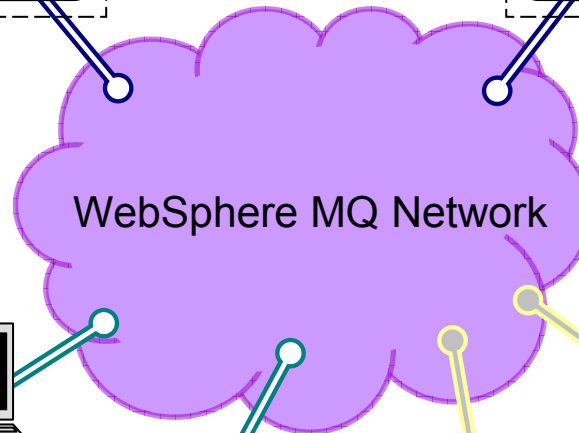
Agents:

- MQ applications
- End-points for file transfers
- Transfer files using MQ
- Multi-threaded
- Capable of simultaneous bi-directional file transfer



Monitoring / Logging

- Collect data about file transfers



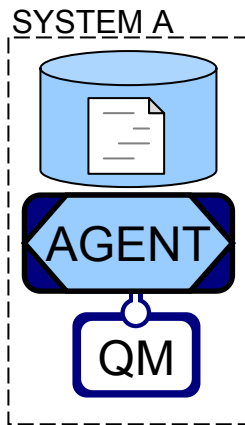
Commands:

- Instruct agents what to do
- Encode command as a message

Connecting Agents to MQ

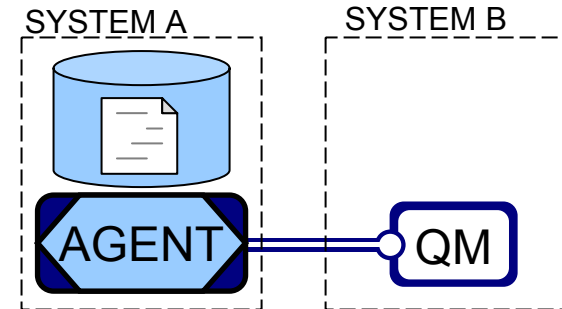
- On distributed platforms, agents can connect to MQ either directly, or using an *MQ Client* style connection

Direct connection to MQ



Both agent and queue manager must run on the same machine

“MQ Client” style connection



Agent and queue manager can run on different machines, connected by a TCP/IP network

- When agents connect as clients:
 - MQ “sees” the agent as a client
 - No dependency on MQ client libraries

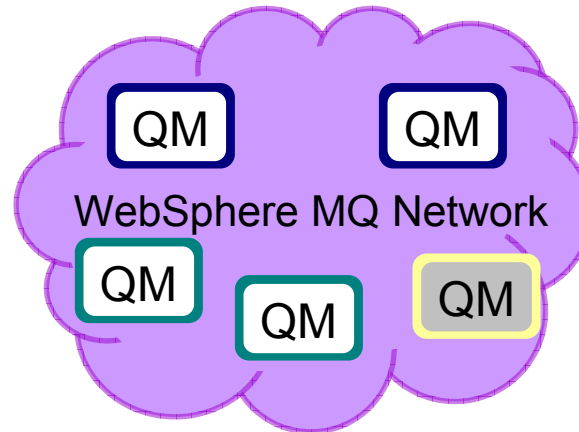
Technical considerations when connecting agents

- Performance
 - Connecting to a local queue manager offers the best possible performance
- Recovery time
 - Connecting to a local queue manager offers faster recovery from loss of network connectivity
- **BUT:** The reliability of transfers is not affected by whether the agent connects to a local queue manager or connects as an MQ Client

Recap: Queue Manager Roles

Agent Queue Manager Role

- Connection point for agent
- Hosts queues belonging to the agent



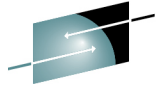
Command Queue Manager Role

- Connection point for commands

Coordination Queue Manager Role

- Gathers logging and monitoring information
- Publishes this information to interested parties

- An MQ queue manager can play between zero and all three of these roles!



SHARE
Technology · Connections · Results

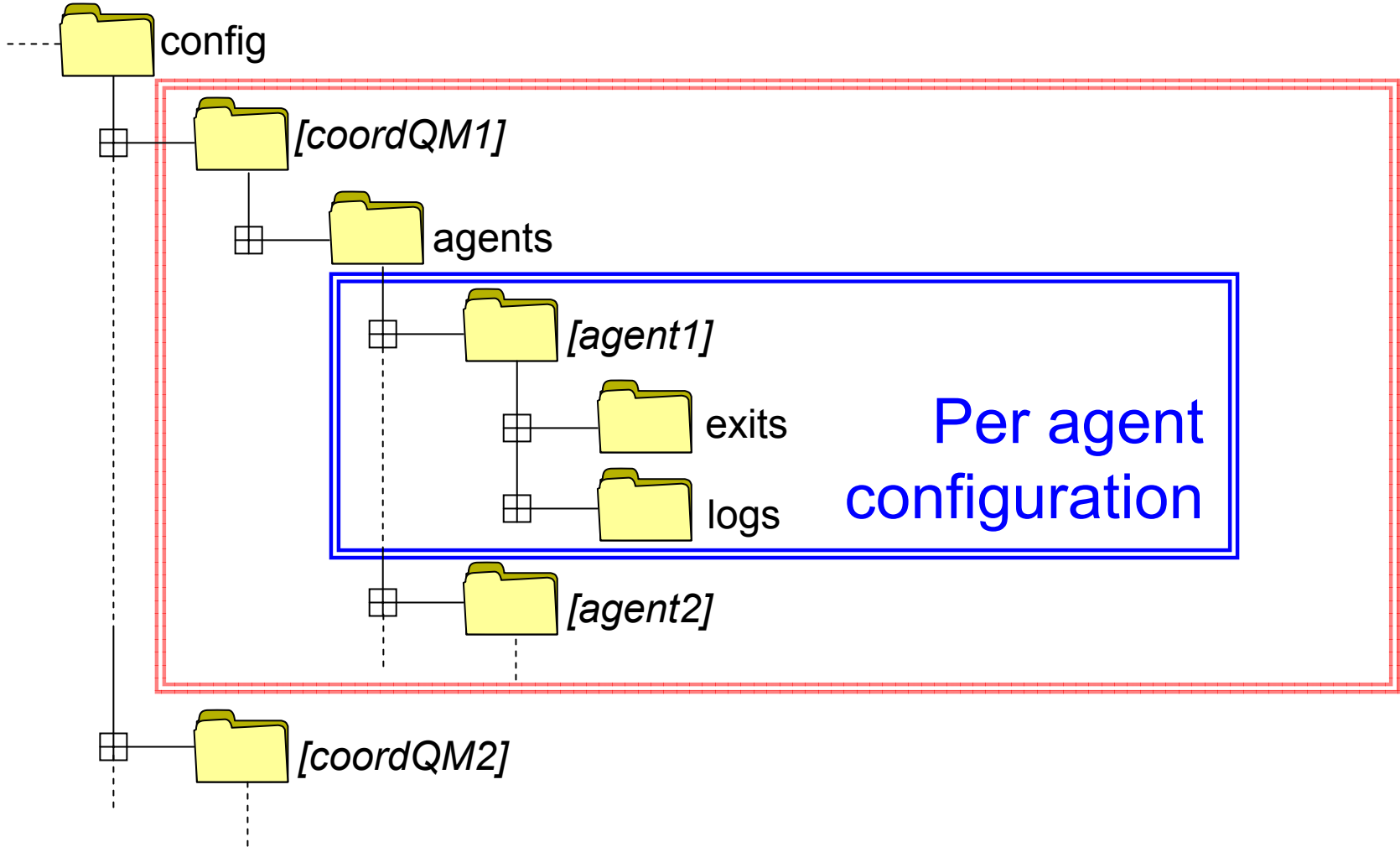


Configuration

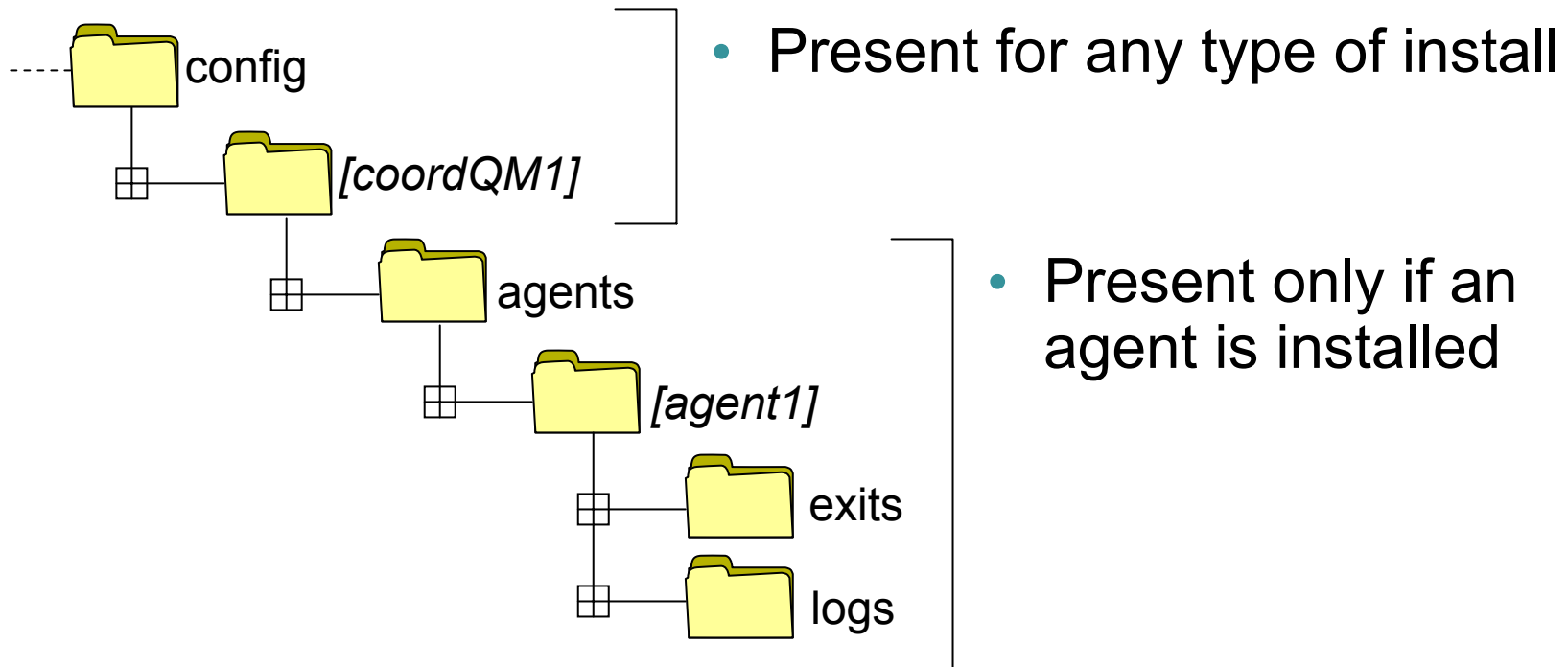
Configuration

- The agents, commands and monitoring/logging functions all require configuration that instructs them how to connect to their queue managers
- The configuration is held on the local file system where the components are installed
- Configuration commands:
 - fteSetupCoordination
 - fteSetupCommands
 - fteCreateAgent
 - fteCreateBridgeAgent
 - fteDeleteAgent

Configuration Directory Structure

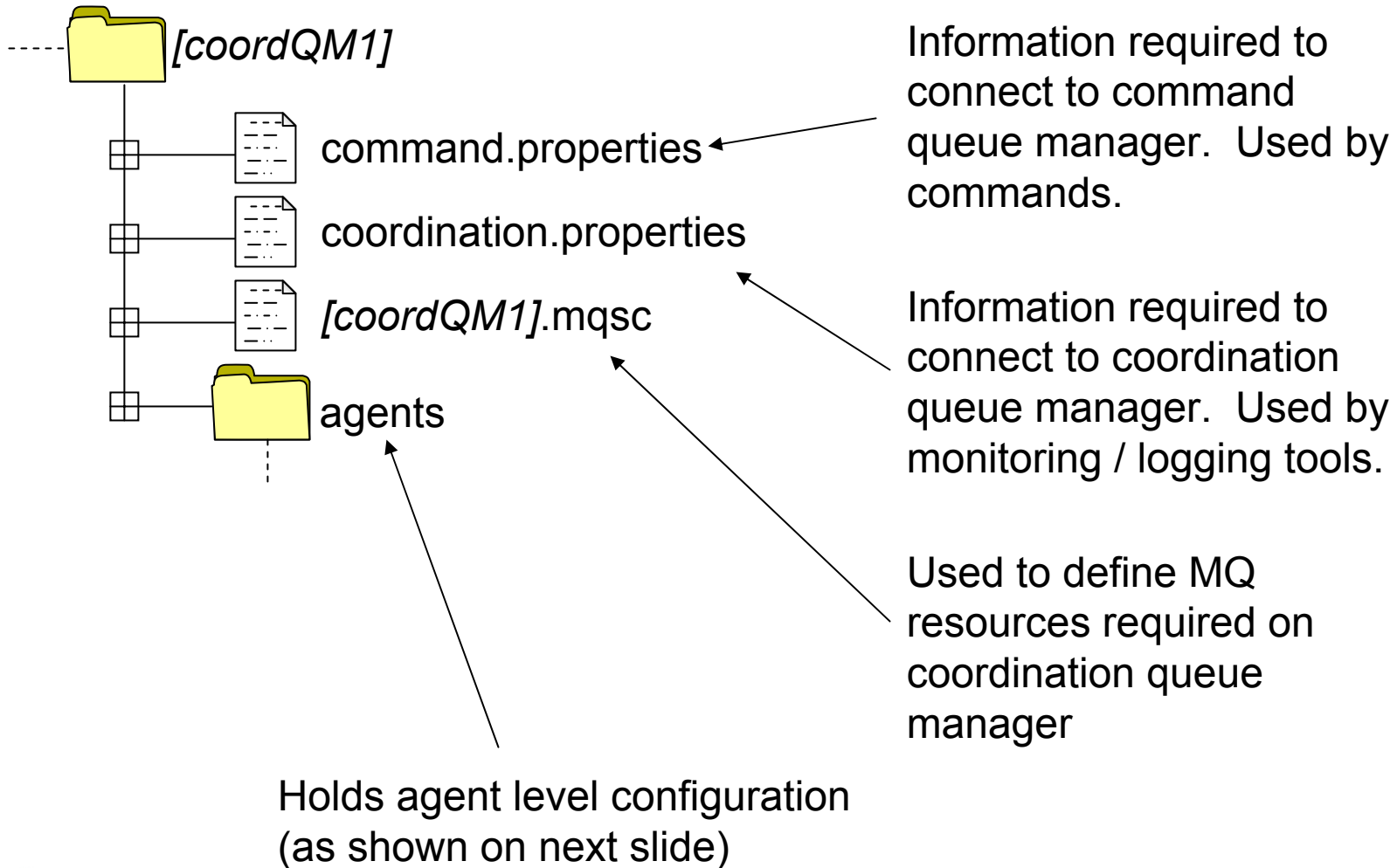


“Real World” Configuration

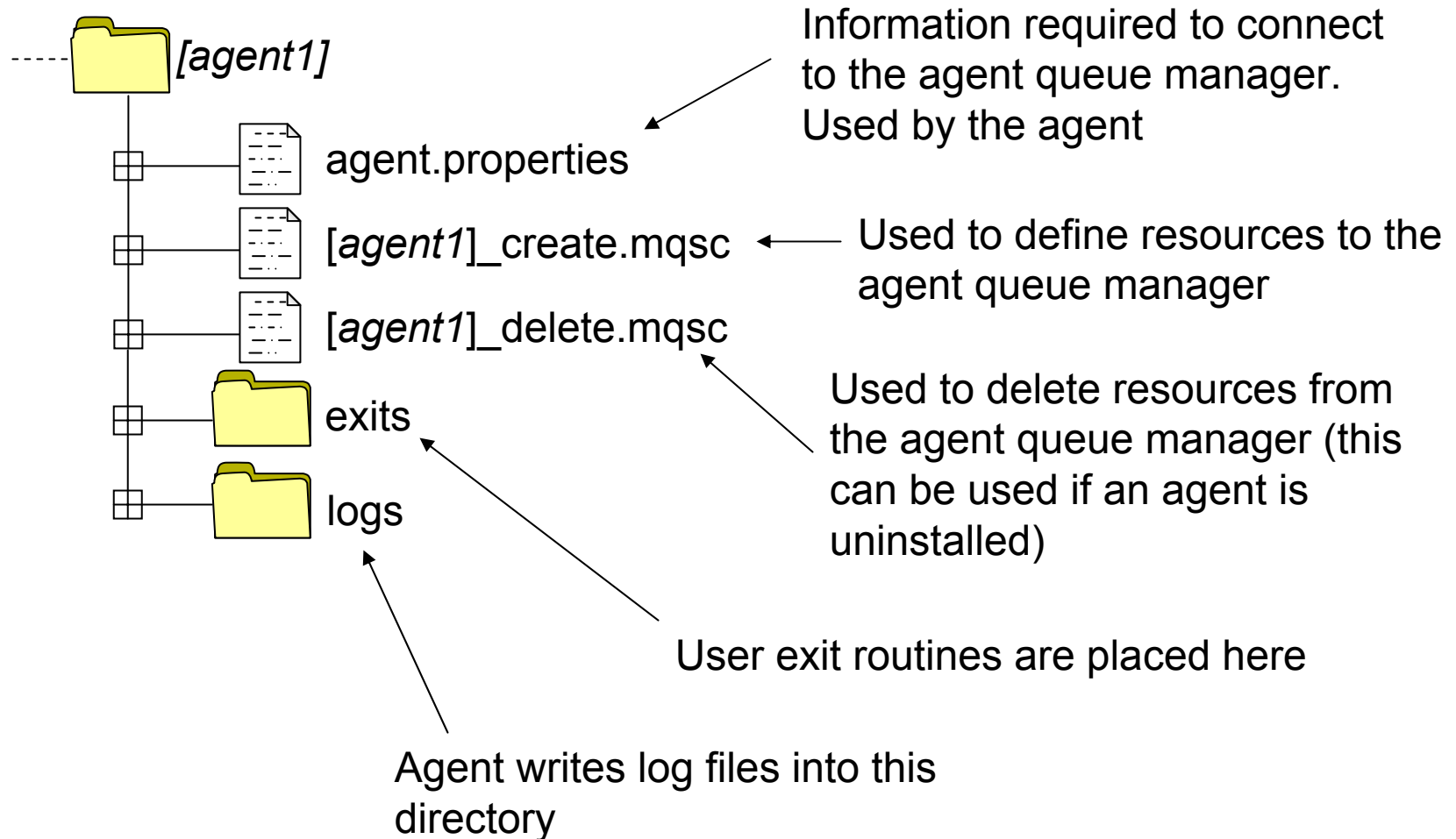


- Typically only a single agent (or a single set of commands) would be present on a system

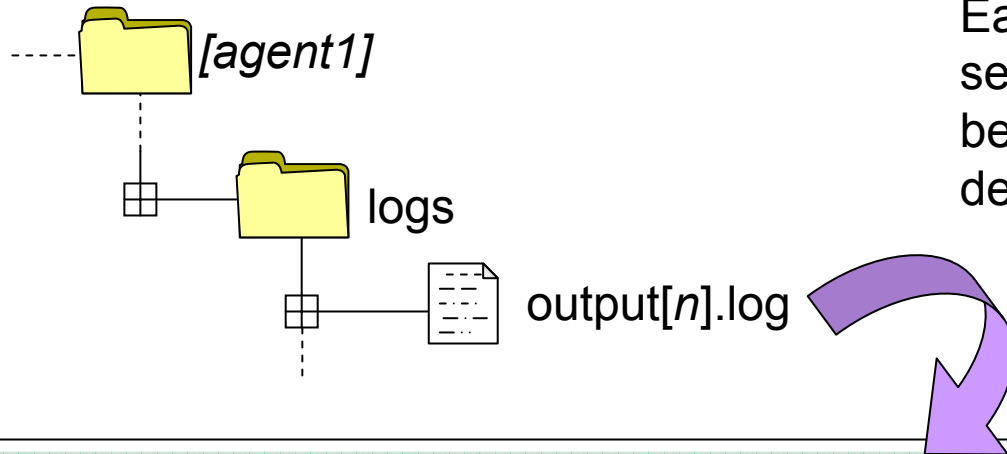
Coordination Level Configuration



Agent Level Configuration



Agent Event Log File



Each agent writes a wrapping sequence of log files that can be used for problem determination

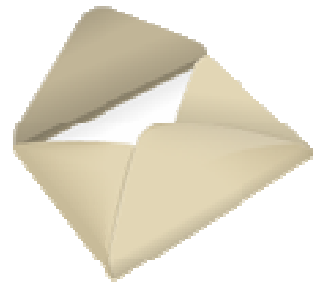
```

O ***** Start Display Current Environment ***** O
O Build level: f700-L081125.1 O
O Properties: O
O agentDesc=Running on Adrian's T43p, agentName=AGENT1, agentQMgr=qm1, coordinationQMgr=qm1 O
O transferRoot=C:\Documents and Settings\IBM_User O
O Test fixes loaded from: C:\Program Files\IBM\WMQFTE\testfix O
O 129424.jar O
O ***** End Display Current Environment ***** O
O [25/02/2009 14:44:35:640 GMT] 00000001 AgentRuntime I BFGAG0058I: The agent has been successfully initialized. O
O [25/02/2009 14:44:37:843 GMT] 00000001 AgentRuntime I BFGAG0059I: The agent has been successfully started. O
O [25/02/2009 14:44:43:031 GMT] 00000010 FTECommandIn I BFGCH0016I: A request to shut down this agent has been O
O [25/02/2009 14:44:43:031 GMT] 00000010 FTECommandIn I received. O
O [25/02/2009 14:44:43:062 GMT] 00000001 AgentRuntime I BFGAG0060I: The agent has completed its current transfers O
O [25/02/2009 14:44:43:062 GMT] 00000001 AgentRuntime I and is now stopping. O
O

```

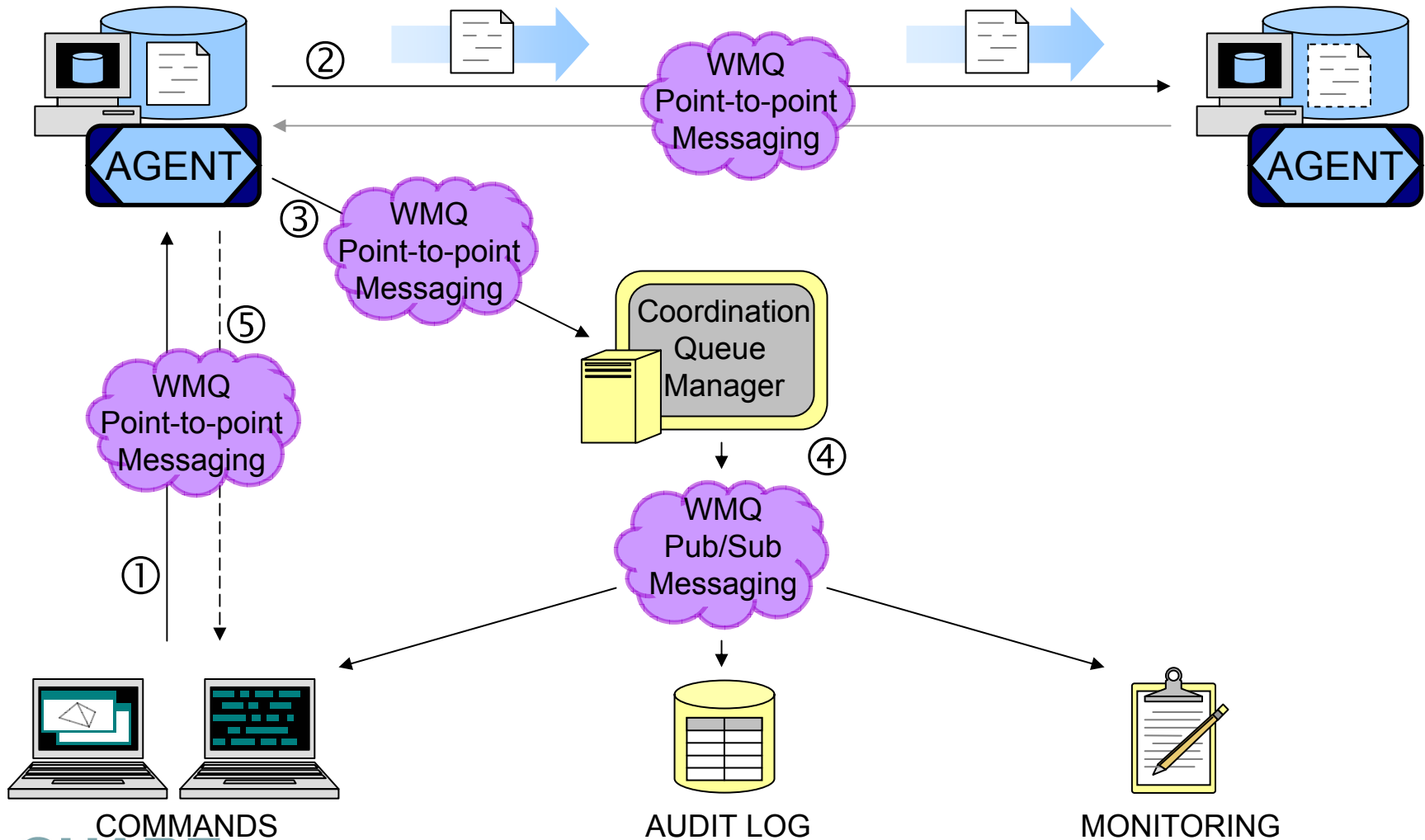
“But this seems awfully complex...”

- The directory structure used to store agent configuration can appear complex
 - It is designed so that groups of FTE agents can use a common shared filesystem to store this information
 - By using a symbolic link (to writable storage) for the logs directory, the configuration can even be mounted read-only.



Message Paths

Overview of Messaging Paths

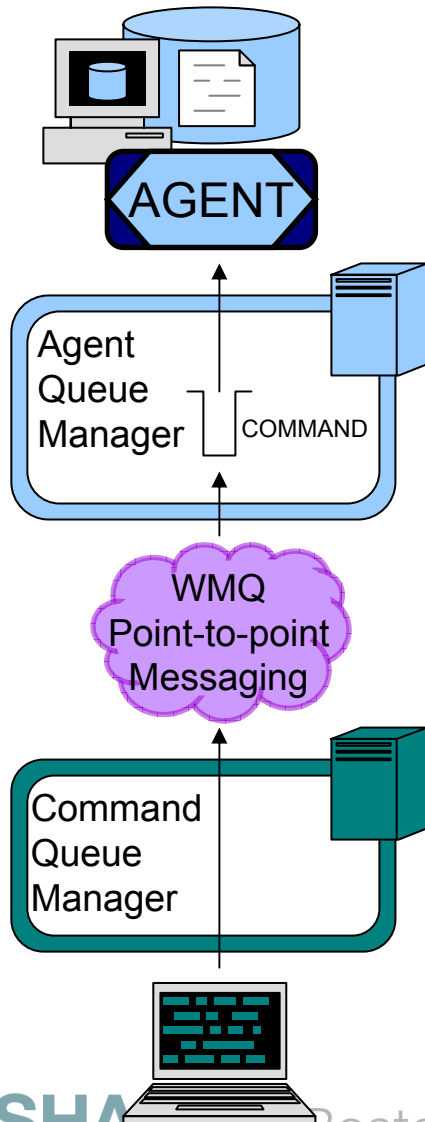




NOTES

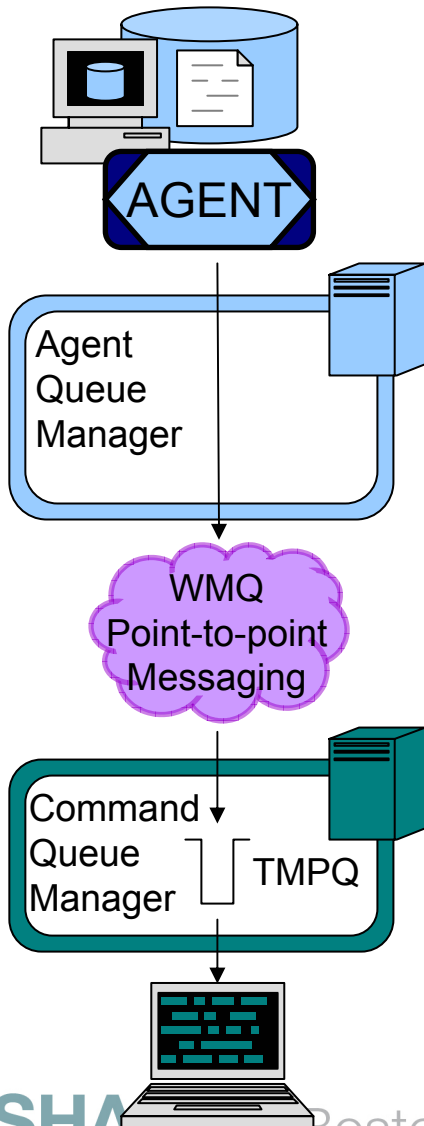
- Step 1: the command line, or GUI sends an MQ point-to-point message to the source agent. This message is routed via the MQ network to a queue which the source agent is monitoring and contains instructions about the file transfer to perform
- Step 2: based on the instructions in the message, the agent performs some checking (is this user allowed to do this?) and then starts transferring a file to the destination agent. The file data is transmitted by packaging it as MQ messages and routing it through the MQ network to the destination agent which re-assembles the data into a file. The protocol used to efficiently and reliably transfer the file data involves some acknowledgement messages flowing from the destination agent to the source agent.
- Step 3: while the transfer (step 2) is progressing the agent is also generating both monitoring and audit information. Example monitoring information would be: “I’m 10% through the transfer... I’m 20% through...”. Example audit information would be “transfer of file ‘xyz.txt’ has completed for user ‘bill’ with a cryptographic hash of 1234”. This audit information is sent using MQ messages to a queue manager which has been designated as performing the “coordination” queue manager role
- Step 4: at the coordination queue manager both the monitoring and audit information is published (using MQ publish subscribe) to interested parties that might include the GUI, a database for archiving log information or monitoring tools
- Step 5: when the agent has completed transferring the file it can optionally send a notification back to the process which requested the file transfer operation. This uses the standard MQ request/reply pattern.

Instructing the Agent



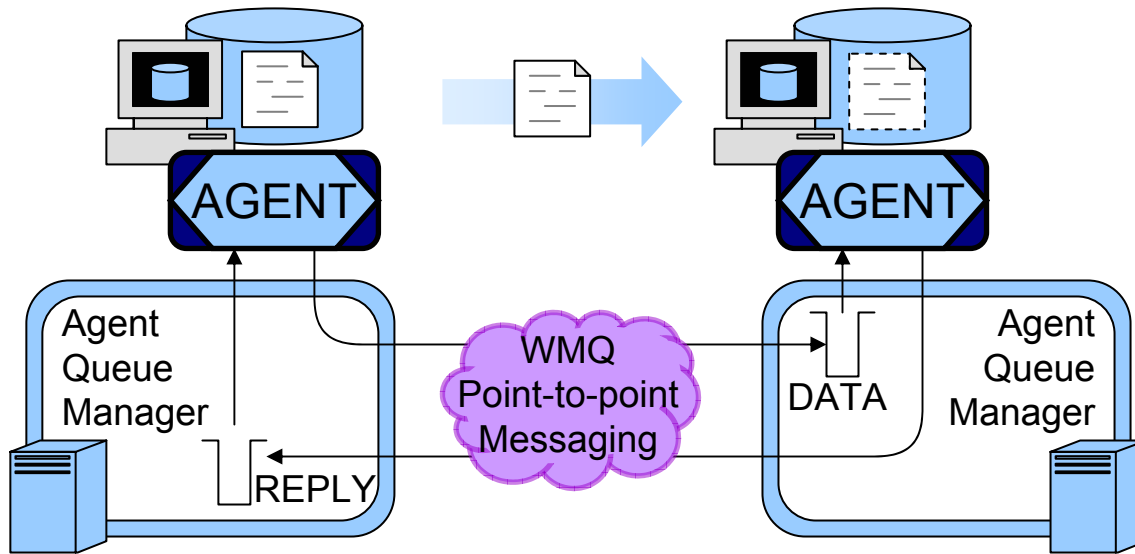
- Agents instructed via messages sent to their command queue
 - This is regardless of whether the GUI, command line, or scripts are used to start the transfer
- Supplied tools can send this via a nominated “command” queue manager
- Format of command message is documented (XML based)
 - Describes the files to send
 - Specifies the agent name and agent queue manager that will be the recipient

Instructing the Agent



- Command messages can specify a reply queue
 - Only applicable to new transfer request in 7.0.0
 - 7.0.1 extends this to a wider range of command messages
- Standard MQ request / response pattern is used to reply
- Reply contains information about the outcome of the operation
- Used to implement “*block while transfer completes*” semantics for command line

Transferring Files

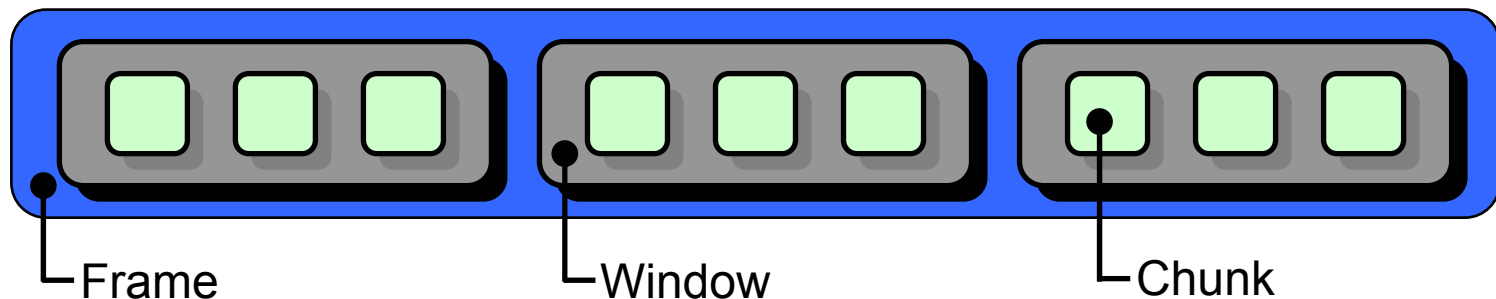


- Transfers involve a two way flow of messages between agents
- Source agent sends data to 'data' queue of destination agent
- Destination agent sends replies to 'reply' queue of source agent

- File data is sent as non-persistent messages
 - Minimises impact on queue manager logs
 - Permits prioritization of file traffic alongside messaging traffic
 - But non-persistent messages can be discarded, so...
- Bi-directional flow of messages allows agents to:
 - Handle non-deliver of non-persistent messages
 - Manage pacing of file transfer (this can be tuned)

Tuning the Transfer Protocol

- The transfer protocol can be tuned by specifying agent properties:
 - Size of messages transmitted (chunks)
 - Tune this to match the optimum message size for your network
 - Frequency of acknowledgements and thus the amount of queued data (windows)
 - Point at which transmission is blocked until an acknowledgment is received (frames)
 - Tune window size and frame size to match the speed and reliability of your network – as well as influence memory usage
 - Frequency at which agent saves state (in frames)
 - Tune this to match the reliability of your network and to influence memory usage



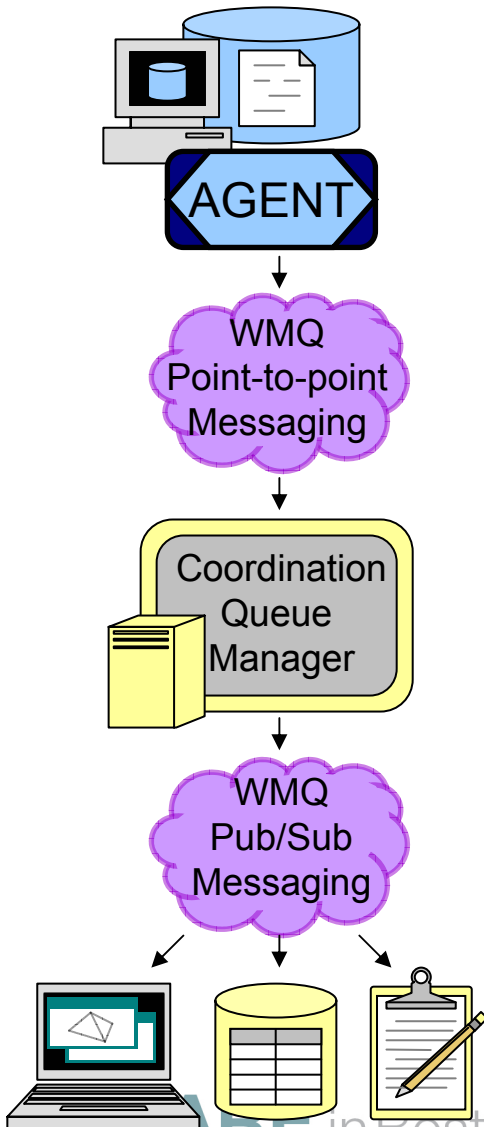


NOTES



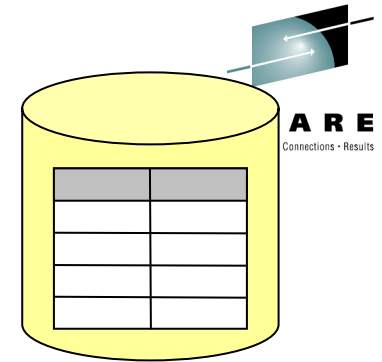
- The protocol used to transfer file data has four key concepts:
 - **Chunks:** correspond to MQ message payloads. A large file is split into *chunks* to transfer it. Multiple small files specified in a single transfer are combined into a *chunk*
 - **Windows:** are made up *chunks*. After each window has been transmitted, the source agent will ask the destination agent to send an acknowledgement back
 - **Frames:** are made up of *windows* and used for pacing. If no acknowledgements are received for any of the *windows* that make up the frame – then the agent will pause and wait for the destination agent to catch up.
 - **Checkpoints:** saving the state of a transfer onto an MQ queue. If the source or destination agent become unavailable they can pick up from where they left off by resuming transferring file data from the last common checkpoint. Between checkpoints the agent caches information in memory.
- There are a number of associated tuning parameters (see: <http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/topic/com.ibm.wmqfte.admin.doc/properties.htm>)
 - **agentChunkSize:** sets the chunk size in bytes. Tune this to the optimum MQ message payload size for the network you are using
 - **agentWindowSize:** window size (in chunks). Use this to determine how often acknowledgements are generated. The only reason to touch this parameter in isolation is because you think FTE is being too chatty (it does however interact with the *agentFrameSize* parameter – as described next)
 - **agentFrameSize:** frame size (in windows). Ultimately this determines both the amount data can be sent before the agent will stop and wait for an acknowledgement (the figure in bytes will be *agentChunkSize* x *agentWindowSize* x *agentFrameSize*) and also, in conjunction with *agentChunkSize* and *agentWindowSize*, the maximum amount of data which can be cached in memory. Tune *agentFrameSize* and *agentWindowSize* down if you are concerned that too much data is being placed onto MQ queues (especially if the destination agent becomes unavailable) or you are concerned about the agent's memory footprint. Tune *agentFrameSize* and *agentWindowSize* up if your network is fast, your disk is slow, and you have lots of RAM - as this will help keep the agent streaming data at all times.
 - **agentCheckpointInterval:** how frequently does the agent take a checkpoint (in frames). Tune this down if you have a very slow and unreliable network as it will minimise the amount of data re-transmitted if connectivity is lost. Tune this up if you have a fast reliable network as it will reduce the time spent persisting state – however note that un-persisted state is stored in memory so this will also increase the memory footprint of the agent.
- More information about the effect of tuning these parameters can be found in the FTE performance reports (FP11, FPL1 and FP61 - linked from here: <http://www-01.ibm.com/support/docview.wss?uid=swg27007197#4>)

Monitoring and Log Data



- Periodically, agents send transfer status to the coordination queue manager
 - Routed to coordination queue manager using point-to-point messaging
 - Coordination queue manager publishes to subscribing applications
- Log data, used for audit purposes, is sent using persistent messaging
- Monitoring data is more transient in nature and sent using non-persistent messaging
- Format of publications is an XML message payload

Log Messages



- Log data published using topic string:
 - `SYSTEM.FTE/Log/<agent>/<transfer_id>`
- To efficiently handle large numbers of messages in a transfer, each transfer generates:
 - A start log message
 - One or more intermediate log publications containing information about a batch of files transferred
 - An end log message
- Information gathered includes:
 - Time, agent names, identity of submitter, files moved, checksums, outcome
- Other key events also generate log messages:
 - Directory monitoring events, schedule events, receipt of malformed messages, etc.

Monitoring Messages

- Monitoring data published using topic string:
 - `SYSTEM.FTE/Transfers/<agent>/<transfer_id>`
- Messages are produced periodically during a file transfer
- Information gathered includes:
 - Bytes transferred / total bytes
 - Elapsed time
 - Current file / total files in transfer

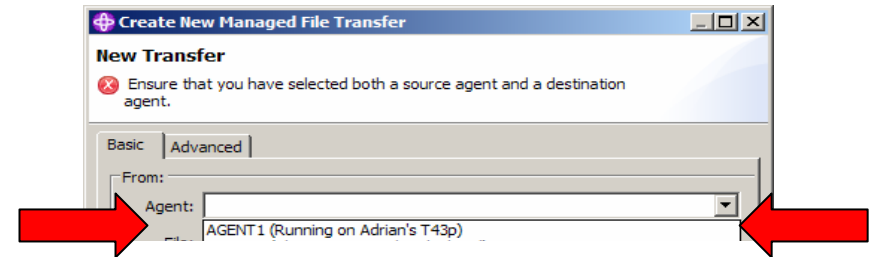


Retained Publications

- The agent also publishes retained publications containing:
 - Current agent status
 - Scheduling information
 - Directory monitoring information
- This allows a copy of agent status, scheduling and directory monitoring information to be held in a central place

Agent Status

- Agents periodically publish information about their status
- The GUI uses this to pre-fill certain fields



- The command line uses this for:
 - fteListAgents
 - fteShowAgentDetails

```
C:\Program Files\IBM\WMQFTE\bin>fteListAgents
5655-U80, 5724-R10 Copyright IBM Corp. 2008. ALL RIGHTS RESERVED
Agent Name:          Queue Manager Name:
AGENT1               qm1
AGENT2               qm2

C:\Program Files\IBM\WMQFTE\bin>fteShowAgentDetails AGENT1
5655-U80, 5724-R10 Copyright IBM Corp. 2008. ALL RIGHTS RESERVED
Agent Information:
Name:                AGENT1
Description:         Running on Adrian's T43p
Operating System:    Windows XP
Time Zone:           Greenwich Mean Time

Queue Manager Information:
Name:                qm1
Transport:           Bindings

C:\Program Files\IBM\WMQFTE\bin>
```

Agent Status Messages

- Published:
 - As XML message data
 - To a retained publication
 - Using topic string:
 - Topic: SYSTEM.FTE/Agents/<agent_name>
- Information includes:
 - Agent name, agent description, agent OS, agent queue manager name, agent time-zone information, agent version



Accessing Information About WebSphere MQ File Transfer Edition

Types of Information

- Information about agents
 - Agent names
 - Agent details
 - Transfer states
- Information about Transfers
 - Progress
 - Completion states
- Information about Schedules and Monitors
 - What exists in my system

Information about Agents – Agent Names

- The fteListAgents command

```
Command Prompt
D:\WMQFTE\bin>fteListAgents.cmd
5655-U80, 5724-R10 Copyright IBM Corp. 2008, 2009. ALL RIGHTS RESERVED
Agent Name:          Queue Manager Name:
IUT_BRG (SFTP Bridge)  Wakko
IUT_ONE               Wakko
IUT_RWO               Wakko
IUT_TWO               Wakko
WIBBLE                Wakko

D:\WMQFTE\bin>fteListAgents.cmd -v
5655-U80, 5724-R10 Copyright IBM Corp. 2008, 2009. ALL RIGHTS RESERVED
Agent Name:          Queue Manager Name:          Transfers:
                    (Source/Destination)
IUT_BRG (SFTP Bridge)  Wakko                      0/0
IUT_ONE               Wakko                      0/0
IUT_RWO               Wakko                      0/0
IUT_TWO               Wakko                      0/0
WIBBLE                Wakko                      0/0

D:\WMQFTE\bin>
```

Information about Agents – Agent Details

- The fteShowAgentDetails command

```
Command Prompt
D:\WMQFTE\bin>fteShowAgentDetails.cmd IUT_BRG
5655-U80, 5724-R10 Copyright IBM Corp. 2008, 2009. ALL RIGHTS RESERVED
Agent Information:
  Name:          IUT_BRG
  Description:
  Bridge Server: sftp://king.hursley.ibm.com:22
  Operating System: Windows XP
  Time Zone:     Greenwich Mean Time

Queue Manager Information:
  Name:          Wakko
  Transport:     Bindings

D:\WMQFTE\bin>fteShowAgentDetails.cmd -v IUT_BRG
5655-U80, 5724-R10 Copyright IBM Corp. 2008, 2009. ALL RIGHTS RESERVED
Agent Information:
  Name:          IUT_BRG
  Description:
  Bridge Server: sftp://king.hursley.ibm.com:22
  Operating System: Windows XP
  Time Zone:     Greenwich Mean Time
  Build Level:   f702-FP1-20100312-1632

Queue Manager Information:
  Name:          Wakko
  Transport:     Bindings

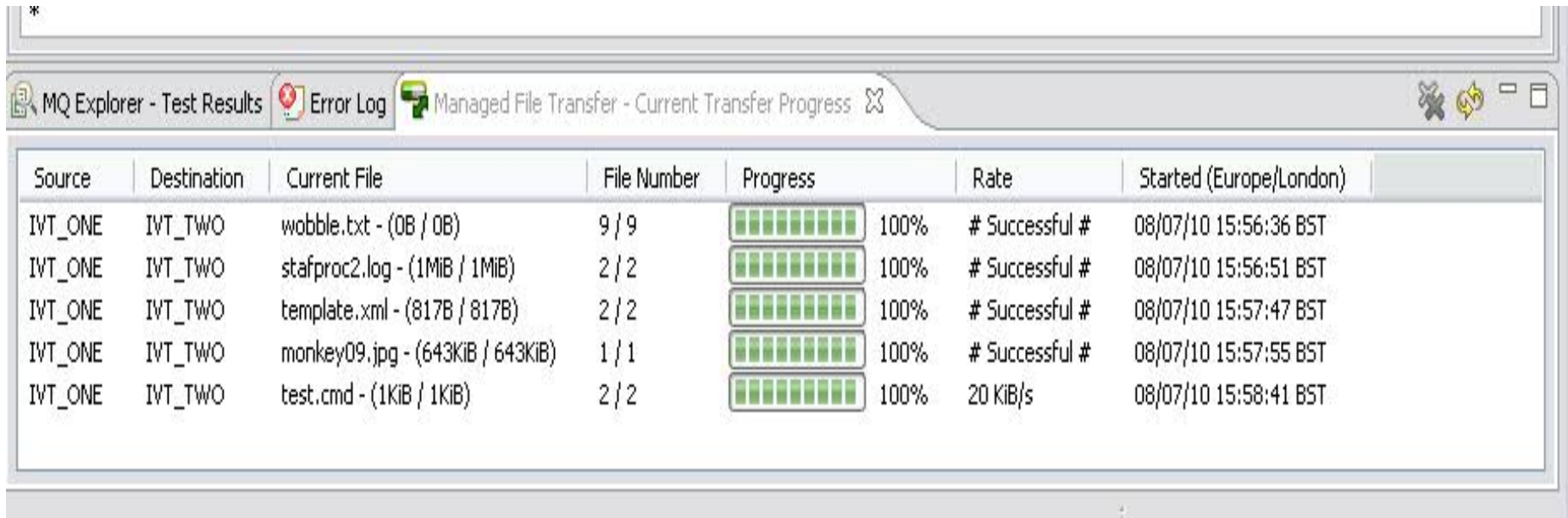
Maximum Number of Running Source Transfers: 25
Maximum Number of Queued Source Transfers: 1000
Source Transfer States:
  No current transfers






Maximum Number of Running Destination Transfers: 25
Destination Transfer States:
  No current transfers

D:\WMQFTE\bin>
```


Information about Transfers – Transfer Progress

- The transfer progress view
 - Snapshot of what is currently happening in the system



Source	Destination	Current File	File Number	Progress	Rate	Started (Europe/London)
IVT_ONE	IVT_TWO	wobble.txt - (0B / 0B)	9 / 9		100%	# Successful # 08/07/10 15:56:36 BST
IVT_ONE	IVT_TWO	stafproc2.log - (1MIB / 1MIB)	2 / 2		100%	# Successful # 08/07/10 15:56:51 BST
IVT_ONE	IVT_TWO	template.xml - (817B / 817B)	2 / 2		100%	# Successful # 08/07/10 15:57:47 BST
IVT_ONE	IVT_TWO	monkey09.jpg - (643KIB / 643KIB)	1 / 1		100%	# Successful # 08/07/10 15:57:55 BST
IVT_ONE	IVT_TWO	test.cmd - (1KIB / 1KIB)	2 / 2		100% 20 KIB/s	08/07/10 15:58:41 BST

Information about Transfers - Completion States

- The `-w` option on `fteCreateTransfer`



```
C:\ Command Prompt
D:\WMQFTE\bin>fteCreateTransfer.cmd -w -sa IUT_ONE -da IUT_TWO -dd d:\output -de
  overwrite "d:\input\*.log"
5655-U80, 5724-R10 Copyright IBM Corp. 2008, 2009. ALL RIGHTS RESERVED
BFGCL0035I: Transfer request issued. The request ID is: 414d512057616b6b6f20202
020202020657f354c2000a503
BFGCL0142I: The file transfer request has been submitted. The command is waiting
  for notification of the transfer's completion.
BFGCL0139I: The requested file transfer has successfully completed.

D:\WMQFTE\bin>
```


Information about Transfers - Completion states

- The Transfer Log

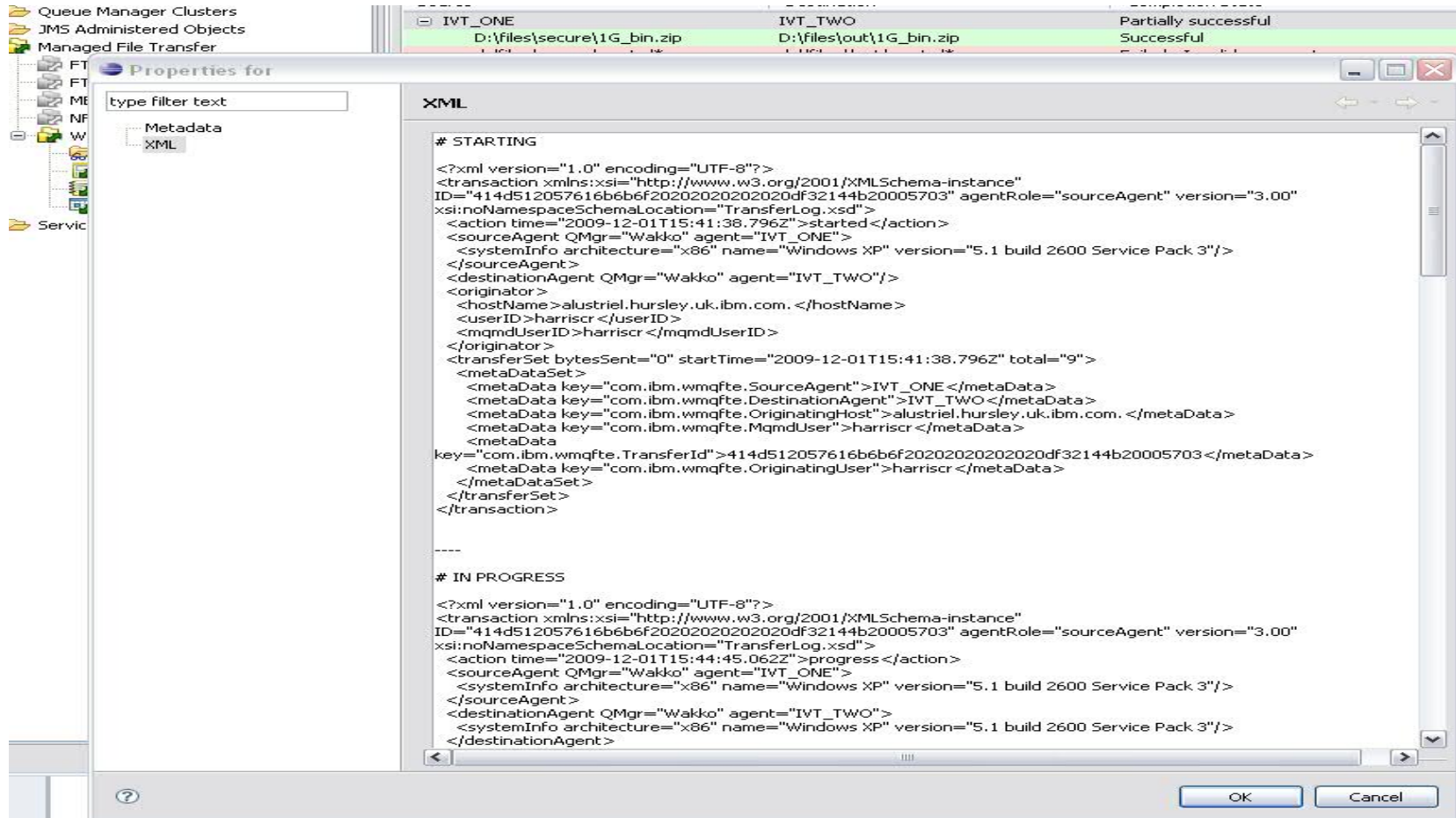
MQ Explorer - Content

Transfer Log

Source	Destination	Completion State	Owner
[-] IVT_ONE	IVT_TWO	Partially successful	harriscr @
D:\files\secure\1G_bin.zip	D:\files\out\1G_bin.zip	Successful	
d:\files\secure\empty*	d://files//out/empty*	Failed - Invalid argument	
D:\files\secure\temp\kitten.jpg	D:\files\out\temp\kitten.jpg	Successful	
D:\files\secure\temp\monkey22.jpg	D:\files\out\temp\monkey22.jpg	Successful	
D:\files\secure\trigger\firework.jpg	D:\files\out\trigger\firework.jpg	Successful	
D:\files\secure\trigger\kitten.jpg	D:\files\out\trigger\kitten.jpg	Successful	
D:\files\secure\trigger\monkey.jpg	D:\files\out\trigger\monkey.jpg	Successful	
D:\files\secure\trigger\test.jpg	D:\files\out\trigger\test.jpg	Successful	
d:\files\secure\triggers*	d://files//out/triggers*	Failed - Invalid argument	
[+] IVT_ONE	IVT_TWO	Failed	harriscr @
[+] IVT_ONE	IVT_TWO	Successful	harriscr @
[+] IVT_ONE	IVT_TWO	Successful	harriscr @
[+] IVT_ONE	IVT_TWO	Successful	harriscr @
[+] IVT_ONE	IVT_TWO	Successful	harriscr @
[+] IVT_ONE	IVT_ONE	Failed	harriscr @
[-] IVT_ONE	IVT_ONE	Partially successful	harriscr @
D:\input\ant.xml	D:\output\test\multifiles.txt	Successful	
D:\input\apache.rft.txt	D:\output\test\multifiles.txt	Failed - BFGIO0006E: File "D:\output\test\multifiles.txt" already exists.	
D:\input\fetch.cmd	D:\output\test\multifiles.txt	Failed - BFGIO0006E: File "D:\output\test\multifiles.txt" already exists.	
IVT_ONE	IVT_TWO	Cancelled	harriscr @
IVT_ONE	IVT_TWO	Cancelled	harriscr @
[+] IVT_ONE	IVT_TWO	Successful	harriscr @

Information about Transfers - Completion states

- The XML messages



The screenshot displays the IBM MQ Explorer interface. On the left, a tree view shows the hierarchy: Queue Manager Clusters, JMS Administered Objects, Managed File Transfer, and Services. The 'Managed File Transfer' folder is expanded, showing 'Properties for' with a 'type filter text' field and a tree view containing 'Metadata' and 'XML'. The main pane shows the XML content for two transfer messages.

Message 1: # STARTING

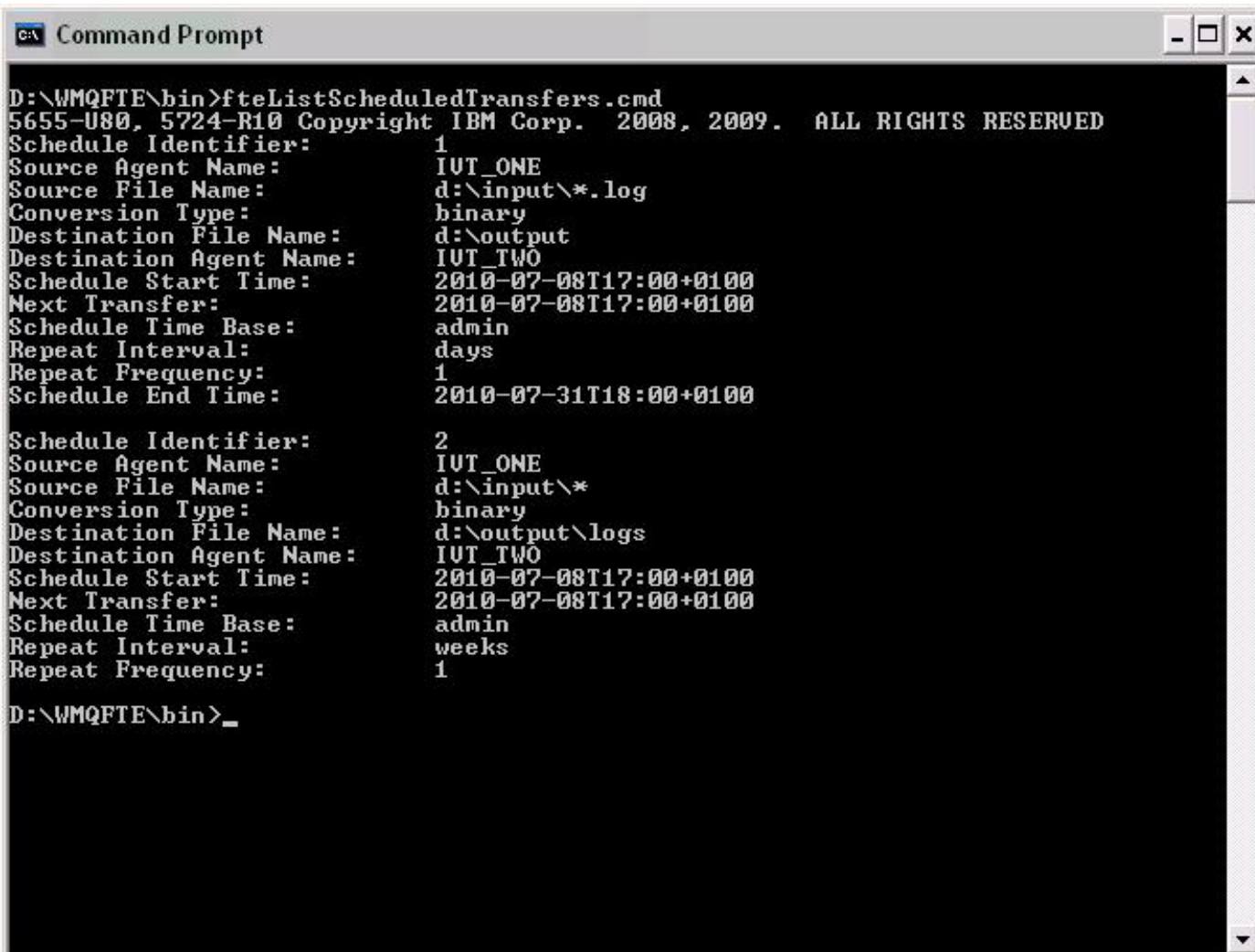
```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
ID="414d512057616b6b6f202020202020df32144b20005703" agentRole="sourceAgent" version="3.00"
xsi:noNamespaceSchemaLocation="TransferLog.xsd">
  <action time="2009-12-01T15:41:38.796Z">started</action>
  <sourceAgent QMgr="Wakko" agent="IVT_ONE">
    <systemInfo architecture="x86" name="Windows XP" version="5.1 build 2600 Service Pack 3"/>
  </sourceAgent>
  <destinationAgent QMgr="Wakko" agent="IVT_TWO"/>
  <originator>
    <hostName>alustriel.hursley.uk.ibm.com.</hostName>
    <userID>harriscr</userID>
    <mqmdUserID>harriscr</mqmdUserID>
  </originator>
  <transferSet bytesSent="0" startTime="2009-12-01T15:41:38.796Z" total="9">
    <metadataSet>
      <metadata key="com.ibm.wmqfte.SourceAgent">IVT_ONE</metadata>
      <metadata key="com.ibm.wmqfte.DestinationAgent">IVT_TWO</metadata>
      <metadata key="com.ibm.wmqfte.OriginatingHost">alustriel.hursley.uk.ibm.com.</metadata>
      <metadata key="com.ibm.wmqfte.MqmdUser">harriscr</metadata>
    </metadataSet>
    <key="com.ibm.wmqfte.TransferId">414d512057616b6b6f2020202020df32144b20005703</key>
    <metadata key="com.ibm.wmqfte.OriginatingUser">harriscr</metadata>
  </transferSet>
</transaction>
```

Message 2: # IN PROGRESS

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
ID="414d512057616b6b6f2020202020df32144b20005703" agentRole="sourceAgent" version="3.00"
xsi:noNamespaceSchemaLocation="TransferLog.xsd">
  <action time="2009-12-01T15:44:45.062Z">progress</action>
  <sourceAgent QMgr="Wakko" agent="IVT_ONE">
    <systemInfo architecture="x86" name="Windows XP" version="5.1 build 2600 Service Pack 3"/>
  </sourceAgent>
  <destinationAgent QMgr="Wakko" agent="IVT_TWO">
    <systemInfo architecture="x86" name="Windows XP" version="5.1 build 2600 Service Pack 3"/>
  </destinationAgent>
</transaction>
```

Information about Schedules

- The `fteListScheduledTransfer` command



```
C:\> Command Prompt

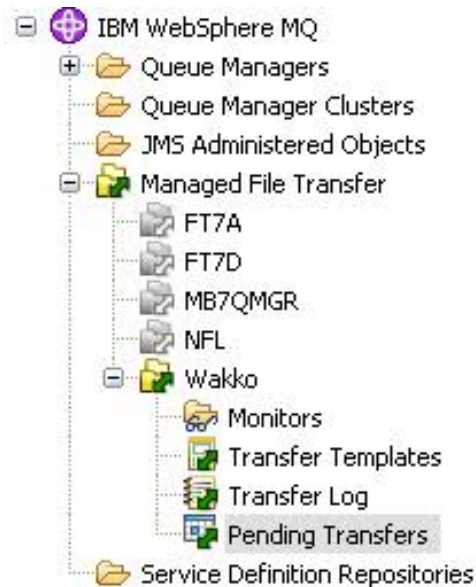
D:\WMQFTE\bin>fteListScheduledTransfers.cmd
5655-U80, 5724-R10 Copyright IBM Corp. 2008, 2009. ALL RIGHTS RESERVED
Schedule Identifier:      1
Source Agent Name:       IUT_ONE
Source File Name:        d:\input\*.log
Conversion Type:         binary
Destination File Name:   d:\output
Destination Agent Name:  IUT_TWO
Schedule Start Time:     2010-07-08T17:00+0100
Next Transfer:           2010-07-08T17:00+0100
Schedule Time Base:     admin
Repeat Interval:         days
Repeat Frequency:        1
Schedule End Time:       2010-07-31T18:00+0100

Schedule Identifier:     2
Source Agent Name:       IUT_ONE
Source File Name:        d:\input\*
Conversion Type:         binary
Destination File Name:   d:\output\logs
Destination Agent Name:  IUT_TWO
Schedule Start Time:     2010-07-08T17:00+0100
Next Transfer:           2010-07-08T17:00+0100
Schedule Time Base:     admin
Repeat Interval:         weeks
Repeat Frequency:        1

D:\WMQFTE\bin>_
```


Information about Schedules

- The pending transfers view



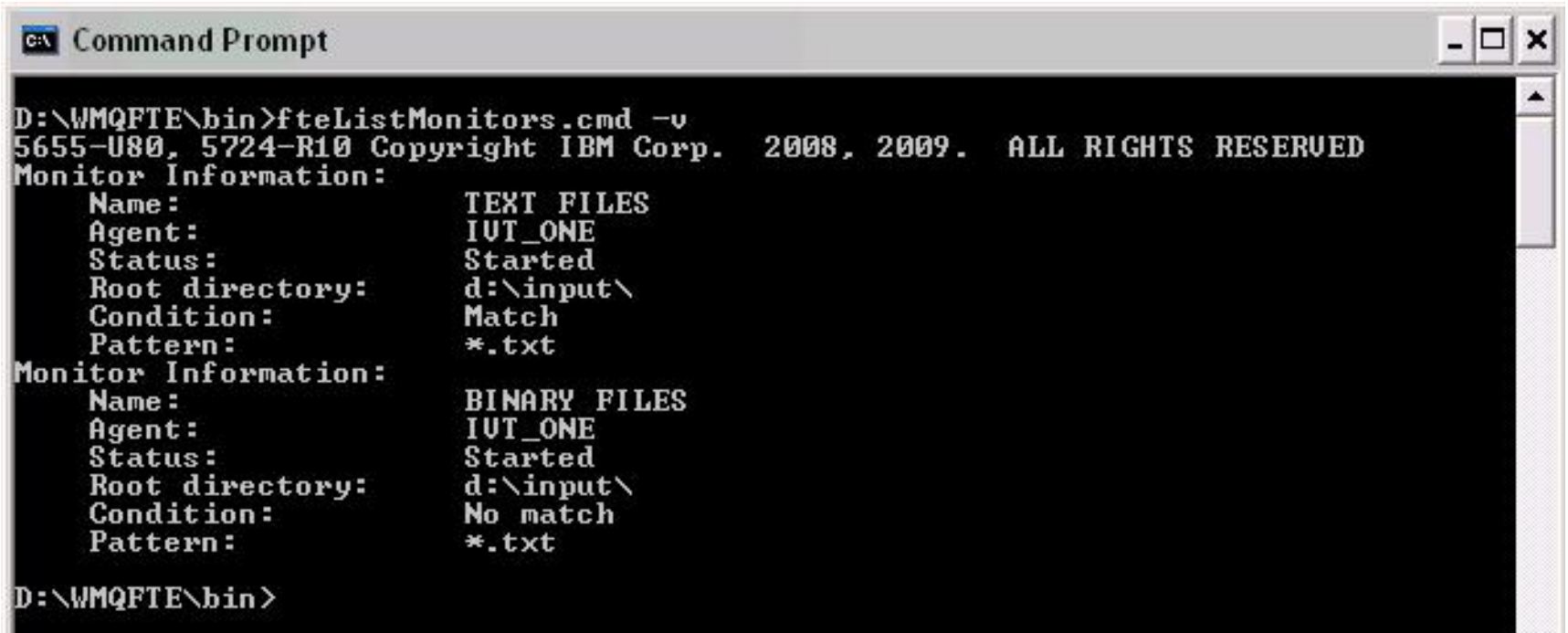
MQ Explorer - Content

Pending Transfers

Name	Source	Destination	Scheduled Start (Europe/London)	Repeat Every	Repeat Type	Repeat Until	Next Transfer (Europe/London)
2	IVT_ONE	IVT_TWO	08/07/10 17:00:00 BST	1	weeks		08/07/10 17:00:00 BST
1	IVT_ONE	IVT_TWO	08/07/10 17:00:00 BST	1	days	31/07/10 16:00:00 BST	08/07/10 17:00:00 BST

Information about Monitors

- The fteListMonitors command



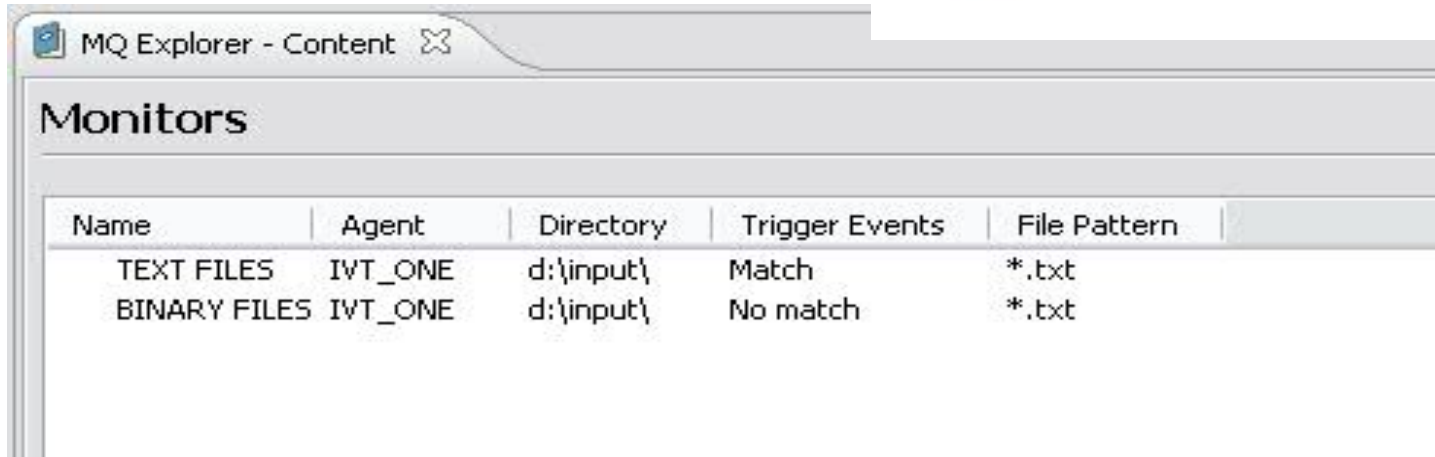
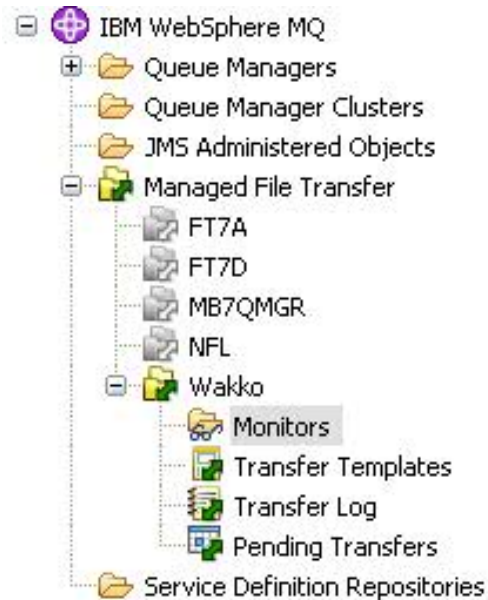
```
C:\> Command Prompt

D:\WMQFTE\bin>fteListMonitors.cmd -v
5655-U80, 5724-R10 Copyright IBM Corp. 2008, 2009. ALL RIGHTS RESERVED
Monitor Information:
  Name:          TEXT FILES
  Agent:         IUT_ONE
  Status:        Started
  Root directory: d:\input\
  Condition:     Match
  Pattern:       *.txt
Monitor Information:
  Name:          BINARY FILES
  Agent:         IUT_ONE
  Status:        Started
  Root directory: d:\input\
  Condition:     No match
  Pattern:       *.txt

D:\WMQFTE\bin>
```

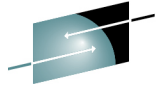
Information about Monitors

- The monitors view

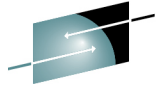


A screenshot of the MQ Explorer 'Monitors' view. The window title is 'MQ Explorer - Content'. The main area displays a table with the following data:

Name	Agent	Directory	Trigger Events	File Pattern
TEXT FILES	IVT_ONE	d:\input\	Match	*.txt
BINARY FILES	IVT_ONE	d:\input\	No match	*.txt



SHARE
Technology · Connections · Results



SHARE
Technology · Connections · Results



Resilience

Resilience to Outages

- FTE is designed to be resilient to both outages and connectivity issues:
 - Effects are localized, other parts of the FTE network can continue to transfer files
 - Check-pointing allows transfers to restart from the point a problem occurs.
- *The following slides describe how FTE function is affected by outages...*

Command
Queue
Manager



Command Queue Manager Becomes Unavailable

- Implications:
 - No new commands can be sent to agents
- Mitigation:
 - Agents continue to process any existing transfers
 - Including scheduled transfers + monitored directories
 - Have more than one queue manager performing the command queue manager role

Agent
Queue
Manager



Agent Queue Manager or Agent Becomes Unavailable

- Implications:
 - Transfers to/from the affected agent(s) cannot progress until the queue manager or agent is available
- Mitigations:
 - Transfers between other agents are unaffected
 - Once the issue is resolved transfers to the affected agent(s) will restart from the point the problem occurs
 - Agent queue manager can be made highly available

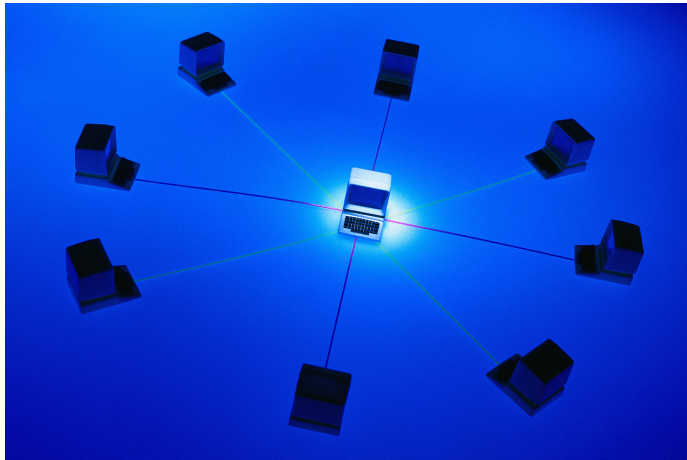


Coordination Queue Manager Becomes Unavailable

- Implications:
 - Logging and monitoring information for in-progress transfers not available
- Mitigations:
 - Agents can continue to transfer files
 - Commands can still be sent to agents
 - Logging and monitoring data is not lost, and will become available when the coordination queue manager is restarted
 - Coordination queue manager can be made highly available

Concurrency and transfers

- Multiple transfer requests are processed in parallel
 - Up to a configurable maximum
- Files specified in a transfer request are transferred in sequence
- However, be aware that:
 - Too many transfers happening in parallel can reduce performance
 - Under load some sequential behaviour emerges resulting in less parallelism than is theoretically possible
 - The size of message placed onto the state queue is proportional to the number of files in the transfer
 - If you wish to transfer very large numbers of files in a single transfer you will need to increase the MQ maximum message size



Topologies

Example Topologies

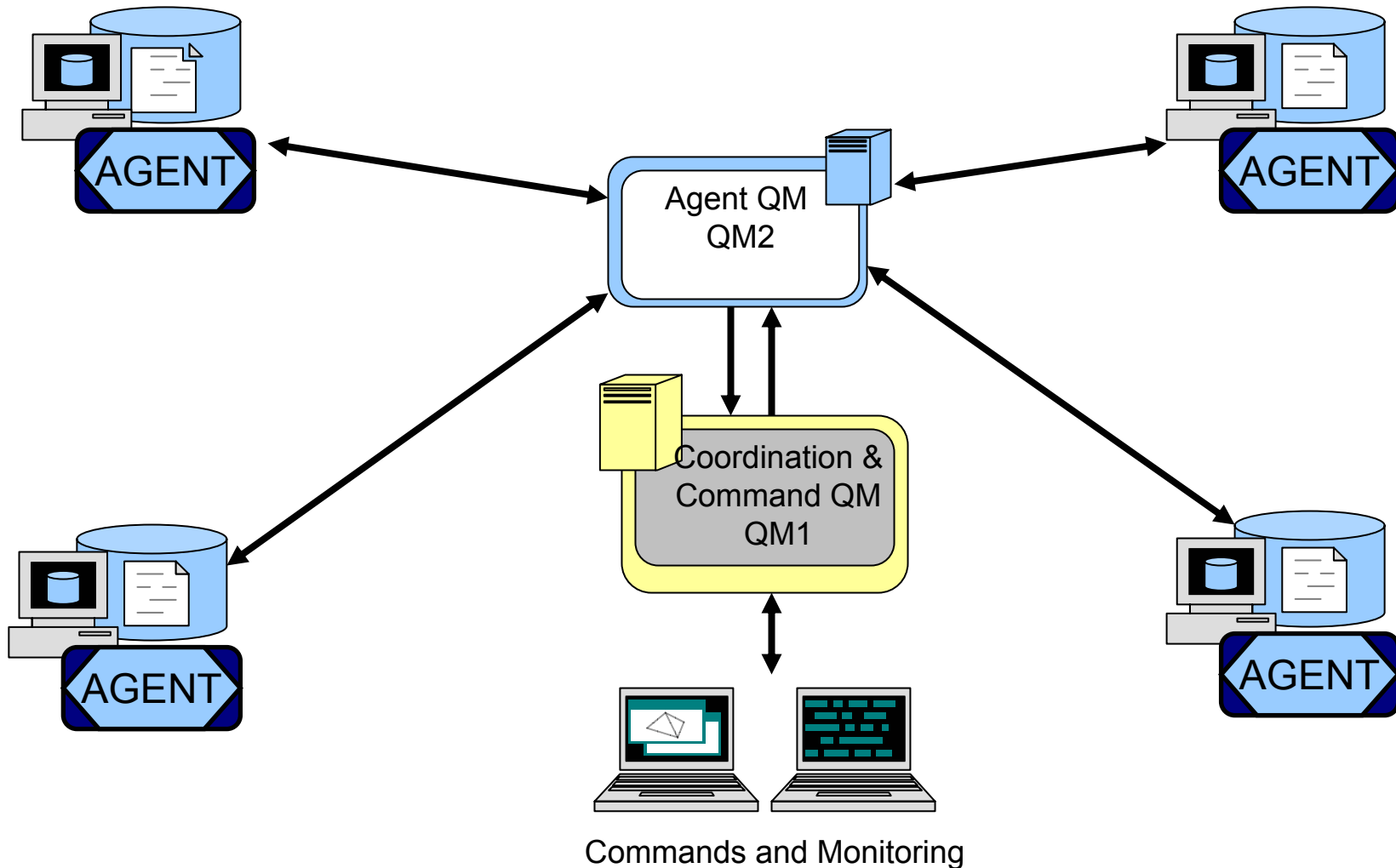
- Hub and Spoke
 - Single agent queue manager
 - Multiple agent queue managers
- Distributed
 - Distributed WMQFTE network
 - Connecting multiple WMQFTE networks



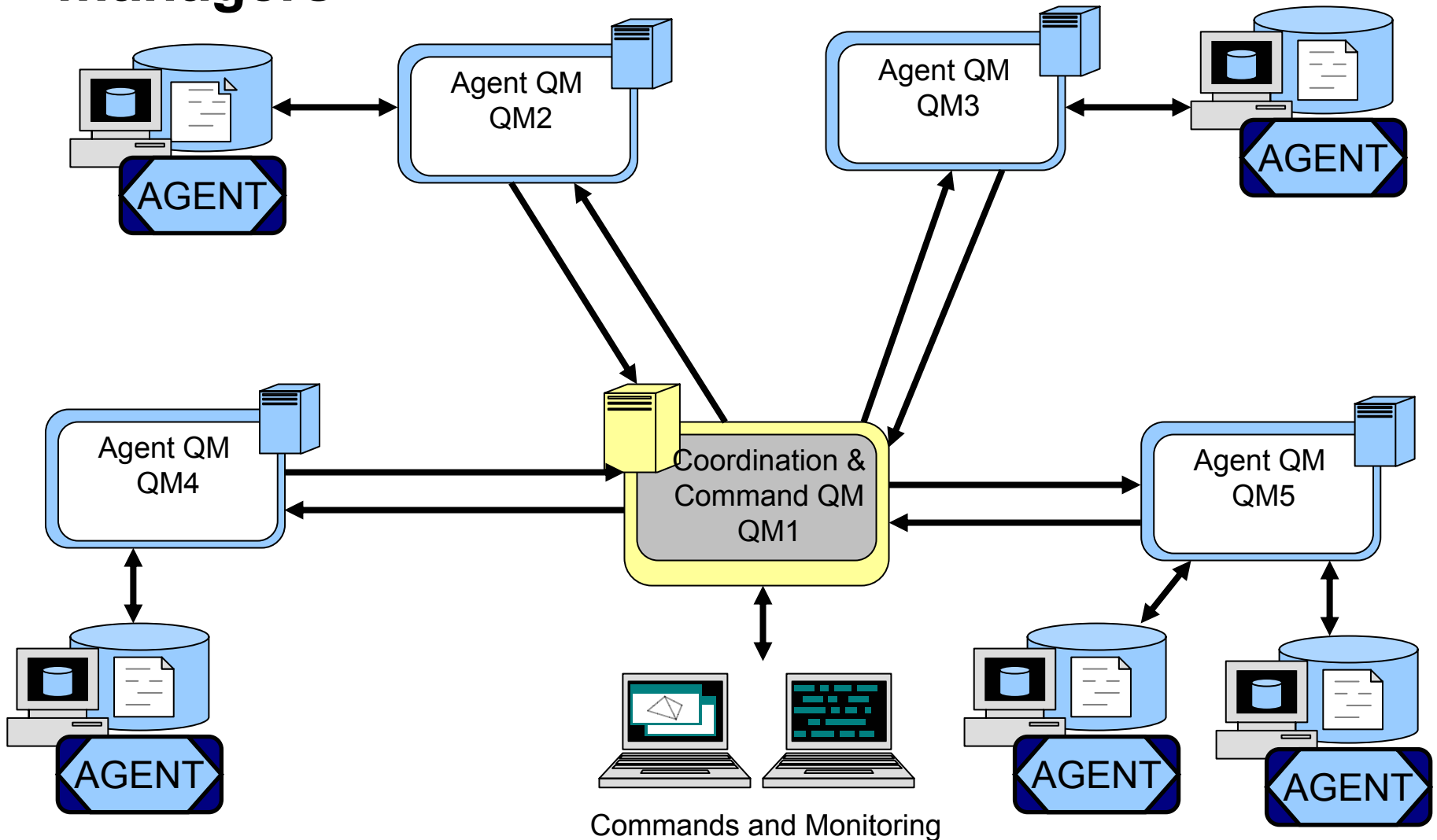
Example Topologies

- Hub and Spoke - Single agent queue manager
 - This is typically seen for a smaller companies with one central office and agents in each of its stores.
- Hub and Spoke - Multiple agent queue managers
 - Typically used for larger companies where files need to be distributed within locations, as well as to head office
- Distributed WMQFTE network
 - This takes a lot of load away from the coordination queue manager. Typically it could be used where the business runs across multiple disparate geographies with a command queue manager for each.
- Connecting multiple WMQFTE networks

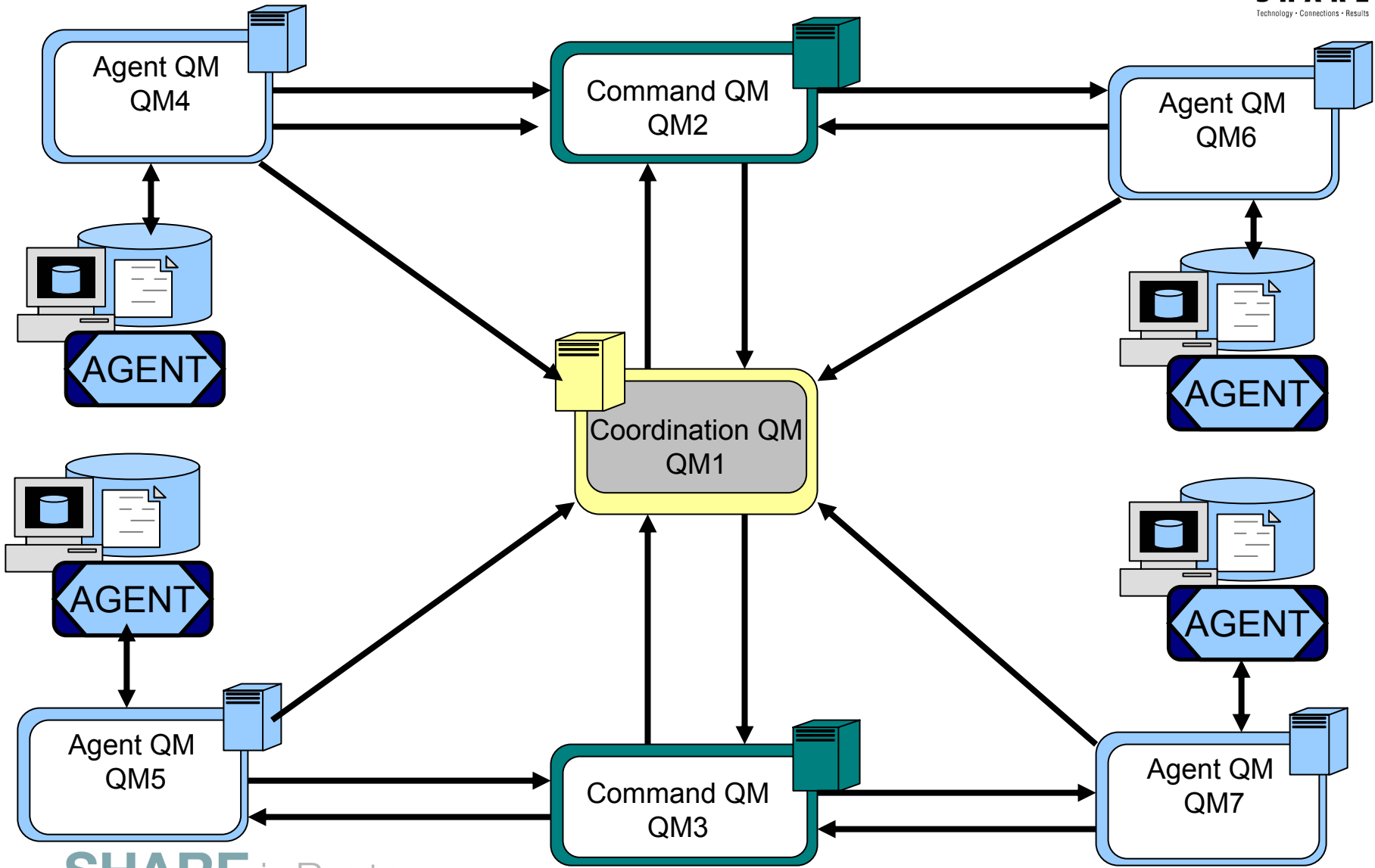
Hub and Spoke – Single agent queue manager



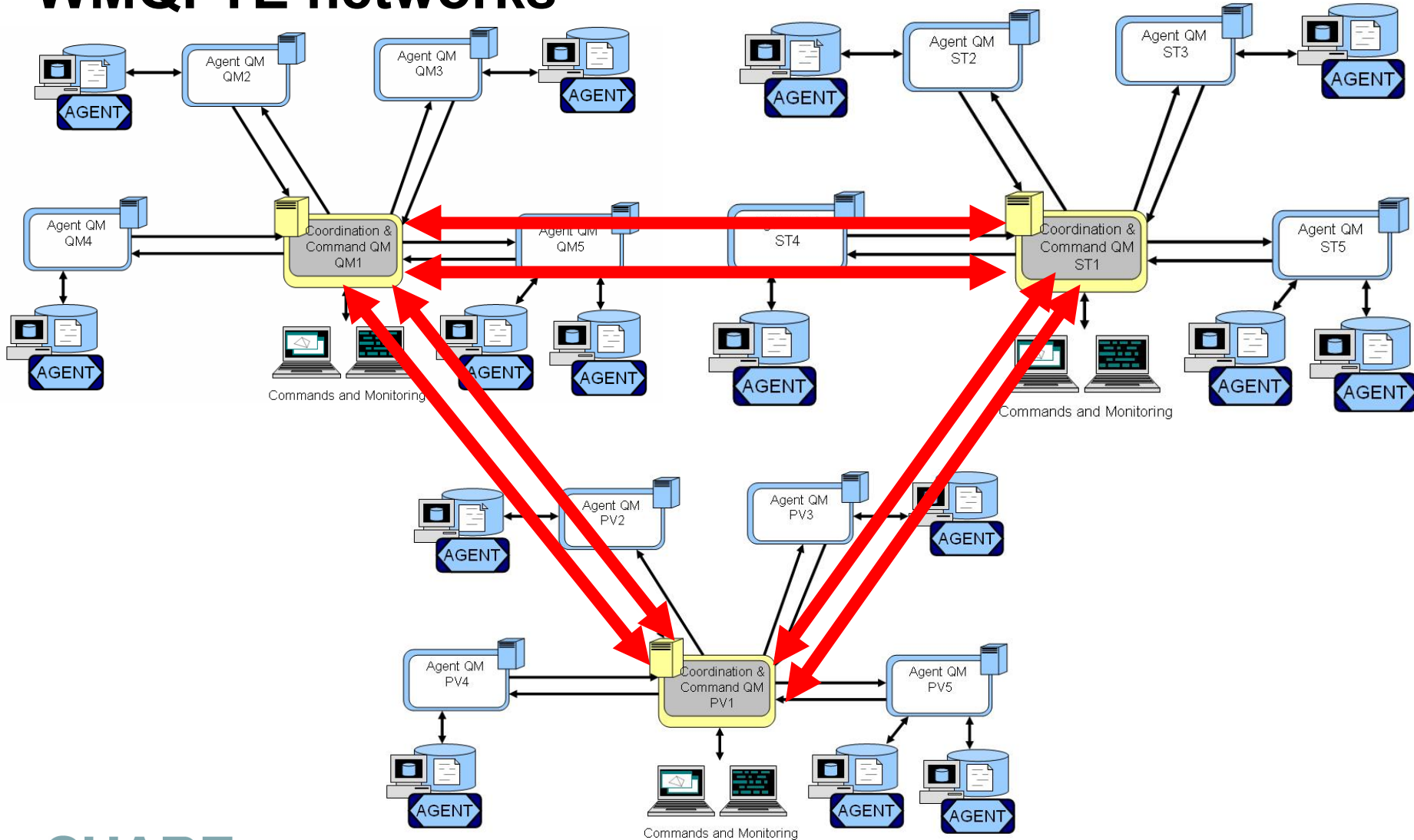
Hub and Spoke – Multiple agent queue managers

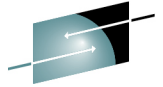


Distributed – Distributed WMQFTE network



Distributed – Connecting multiple WMQFTE networks





SHARE
Technology · Connections · Results

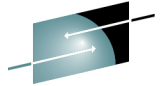
Demo: FO02 – Web Gateway Support Pac



SupportPac FO02 Overview

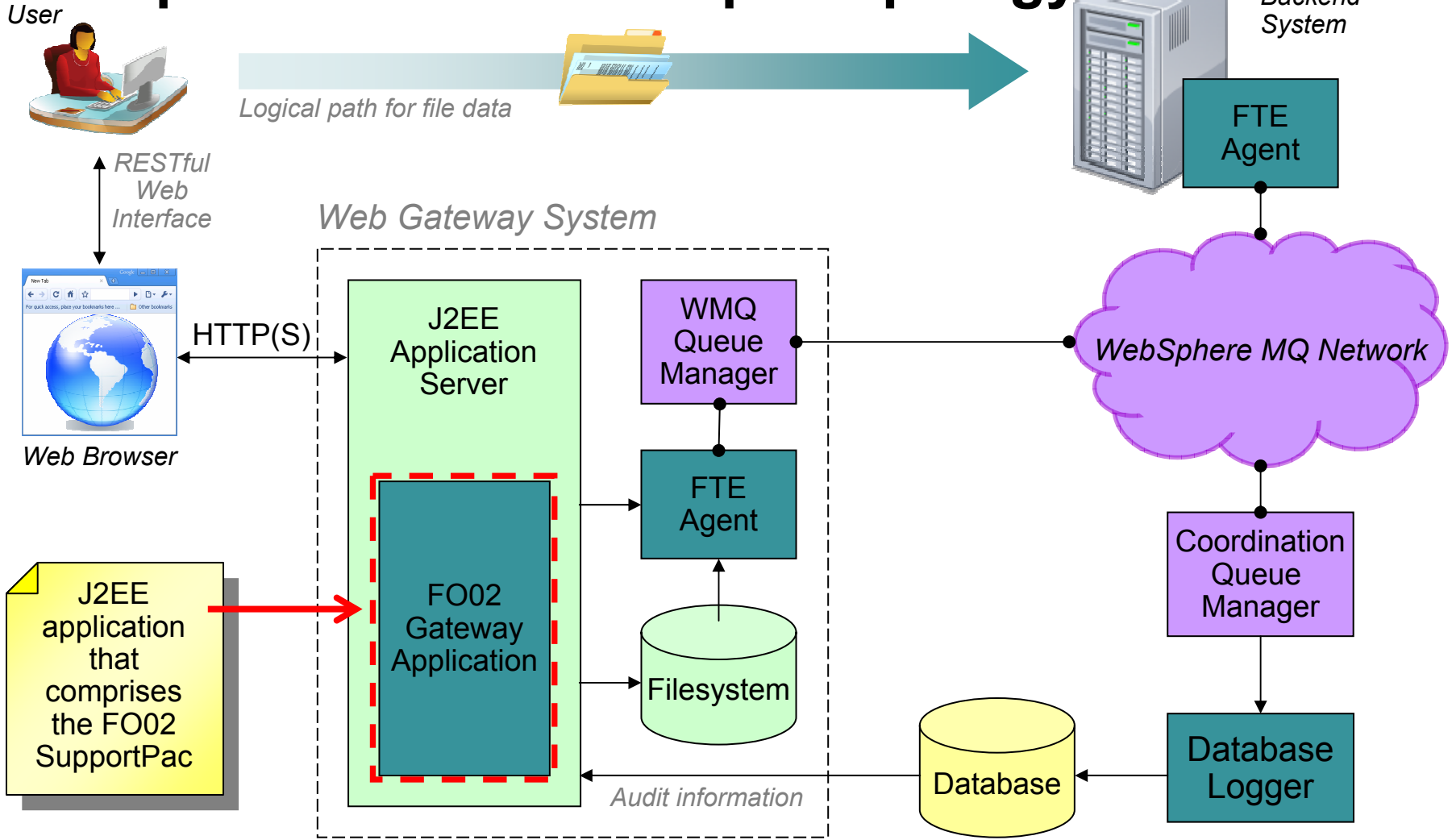


- FO02 is a Category 2 (unsupported) SupportPac which provides a RESTful API to WebSphere MQ File Transfer Edition
 - ***The focus of FO02 is to provide a RESTful API that can be used to build end user Web solutions that involve transferring file data or viewing the outcome of file transfers***
- To download SupportPac FO02:
 - http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24026419&loc=en_US&cs=utf-8&lang=en
- Capabilities:
 - Sending a file from an HTTP client (for example a web browser) to an File Transfer Edition agent
 - Setting typical transfer options (E.g. transfer priority, text/binary transfer type, etc)
 - Receiving a unique ID relating to the transfer which can be used to track the transfer through the FTE system, including querying the status of the transfer (see below)
 - Retrieving audit information about a file transfer operation. Details include:
 - The success of individual files in the transfer
 - The amount of data transferred, the time the transfer occurred
 - etc.



SHARE
Technology · Connections · Results

Components of an example topology





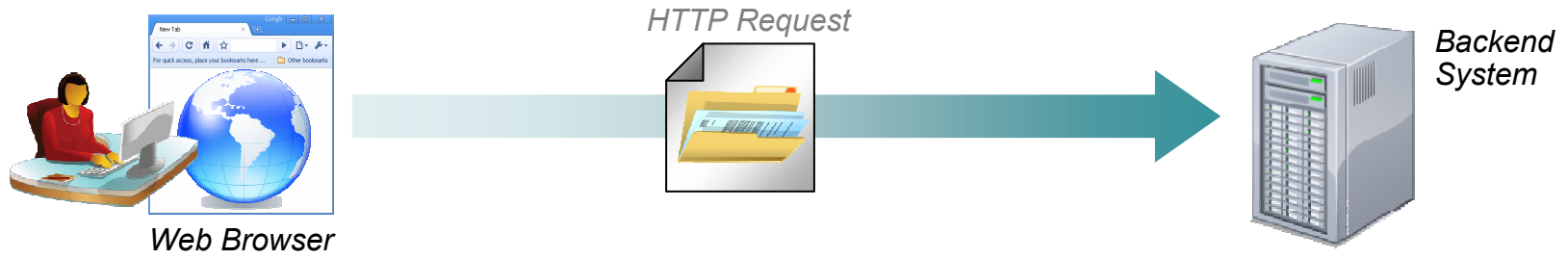
Example interactions on upload of a file:

1. The end user interacts, using a Web browser, with a Web application that has been designed to use the RESTful interfaces provided by the FO02 SupportPac
2. The HTTP POST request, corresponding to the RESTful interaction, is interpreted by the Web Gateway application (running in the J2EE application server) and is used to:
 - A. Write the file data to the local file system of the application server
 - B. Sends a file transfer request to the FTE agent running on the same system as the application server. This request instructs the agent to move the file data (written in step 2A)
3. The FTE agent running on the same system as the application server transfers the file (or files) to the destination agent (as specified in the HTTP POST request – at step 1)
 1. The FTE agent running on the same system as the application server acknowledges the receipt of the file transfer request. This information is returned to the user as a response to the HTTP POST request (and can be formatted for display by the Web application)
 2. As the file transfer progresses audit data is generated, which is flowed to the coordination queue manager. This audit information is recorded in the database, by the Database Logger

Example interactions when querying status of a file transfer:

1. User generates a HTTP GET request (again using a Web browser to interact with a Web application written to use the RESTful interfaces provided by the FO02 SupportPac)
2. The FO02 Web Gateway application interprets the request and queries the database to retrieve the relevant information
3. The information is returned to the user as the response to the HTTP GET request (and can be formatted for display by the Web application, written to use the FO02 RESTful interfaces)

Example: sending a file from an HTTP client to a backend FTE agent (1/2)



Folder to write file(s) to, relative to transfer root of destination agent

Destination FTE agent

Method & URI

```
POST HTTP/1.1 /wmq/fte/file/agent/CLAIMS.AGENT/claims/forms/
```

Headers

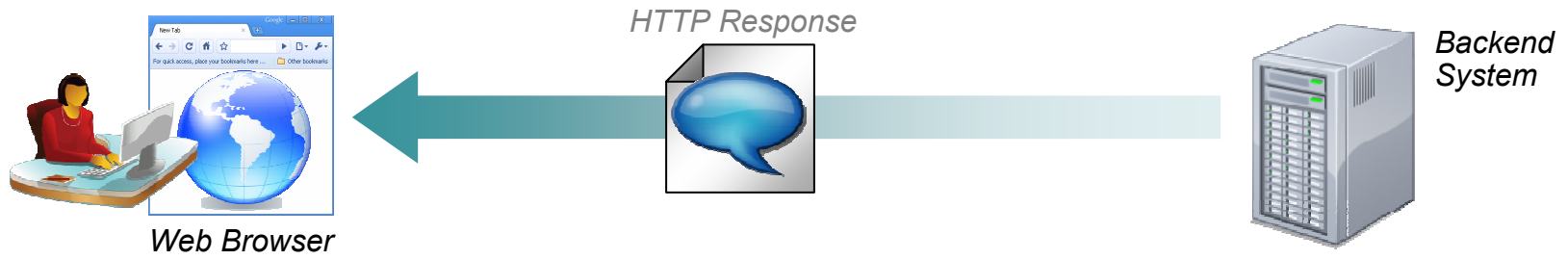
```
x-fte-priority: 4  
x-fte-type: binary  
Content-type: multi-part/form-data; boundary=--Aa6b74
```

Body

```
--Aa6b74  
Content-Disposition: form-data; name="files"; filename="claim29-04-2010.txt"  
Content-Type: text/plain  
  
<contents of claim29-04-2010.txt goes here in the HTTP request>  
--Aa6b74--
```

Filename to write file to at destination agent

Example: sending a file from an HTTP client to a backend FTE agent (2/2)



Operation was successful

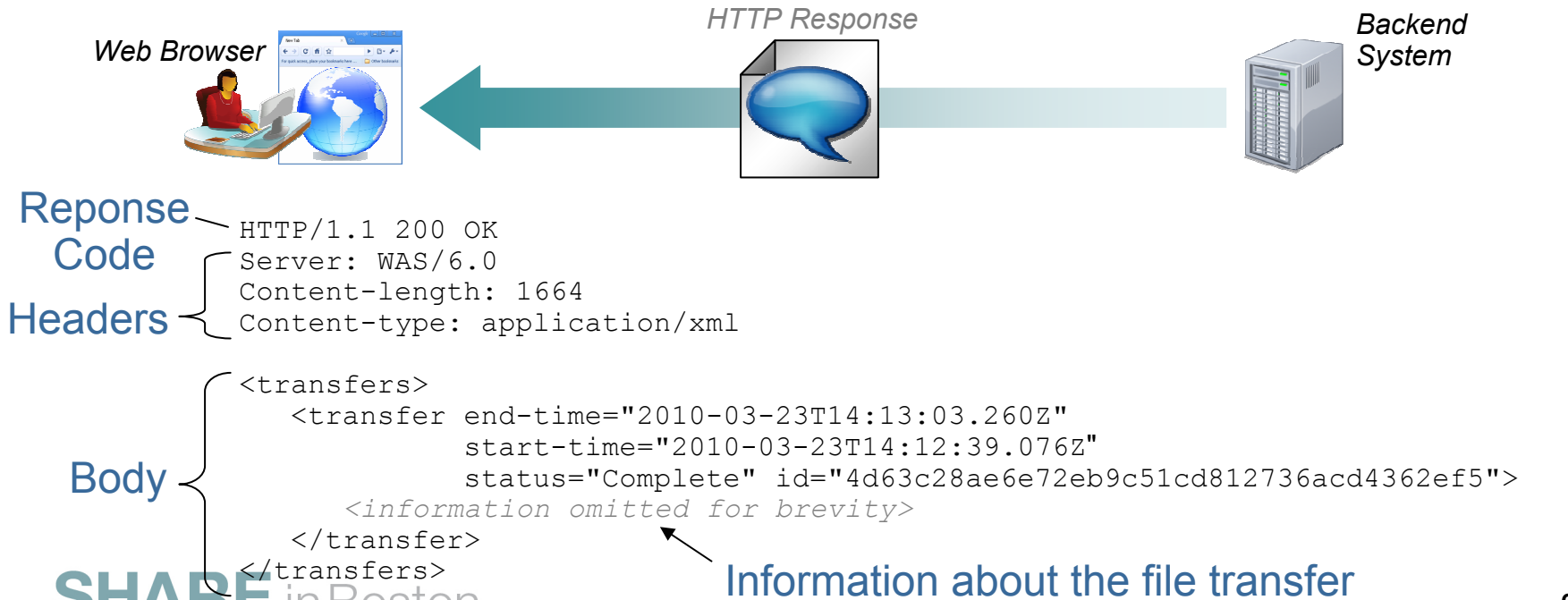
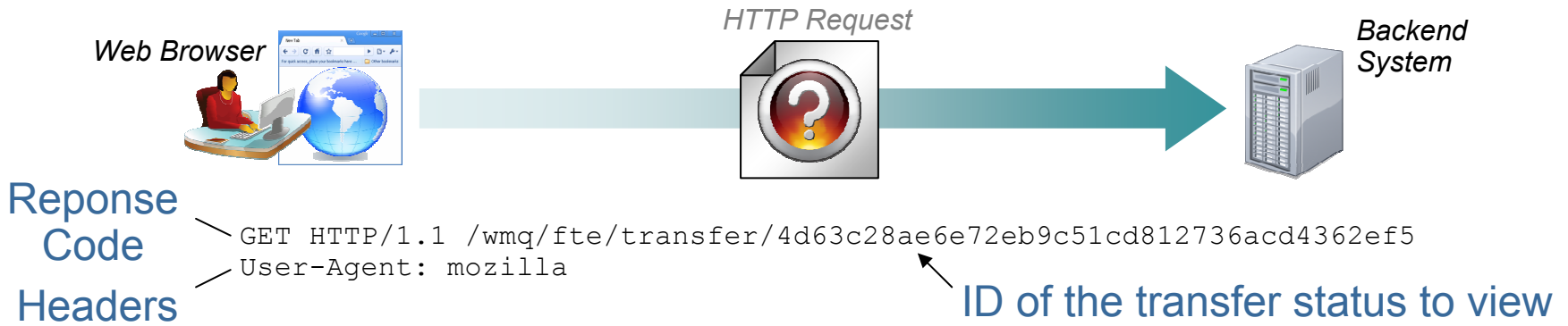
Reponse Code: HTTP/1.1 200 OK

Headers: Server: WAS/6.0
Content-length: 93
x-fte-id: 4d63c28ae6e72eb9c51cd812736acd4362ef5

Body: <transfers>
<submission id="4d63c28ae6e72eb9c51cd812736acd4362ef5">
</submission>
</transfers>

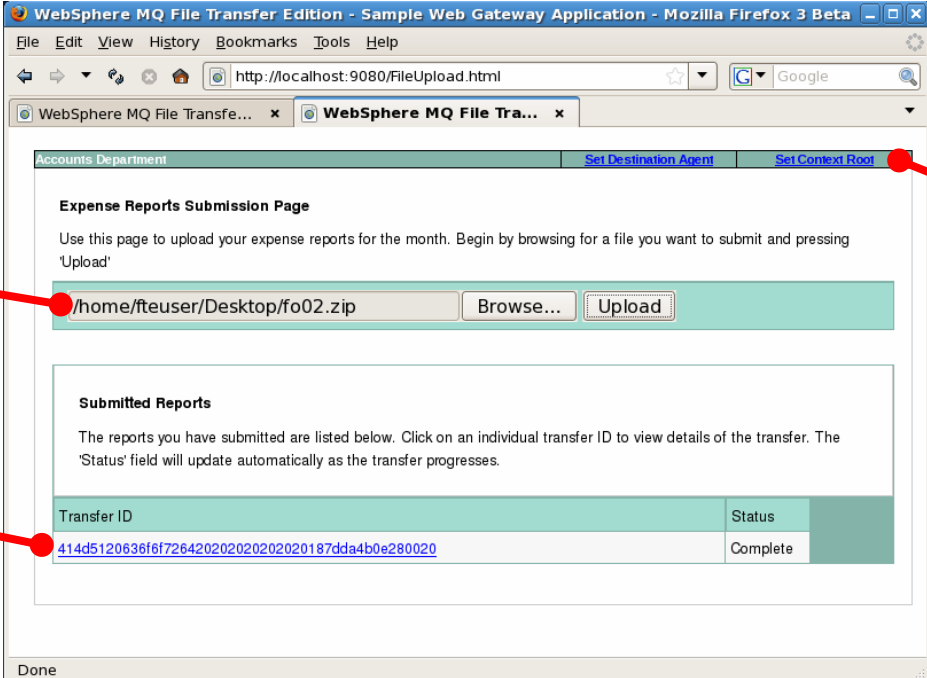
Transfer ID of the transfer

Example: querying information about a file transfer using a HTTP client



Sample application

- The FO02 SupportPac ships with a sample Web application that demonstrates the types of function available from the RESTful API



Select a file and upload it to the backend agent

History of transfers carried out by this user

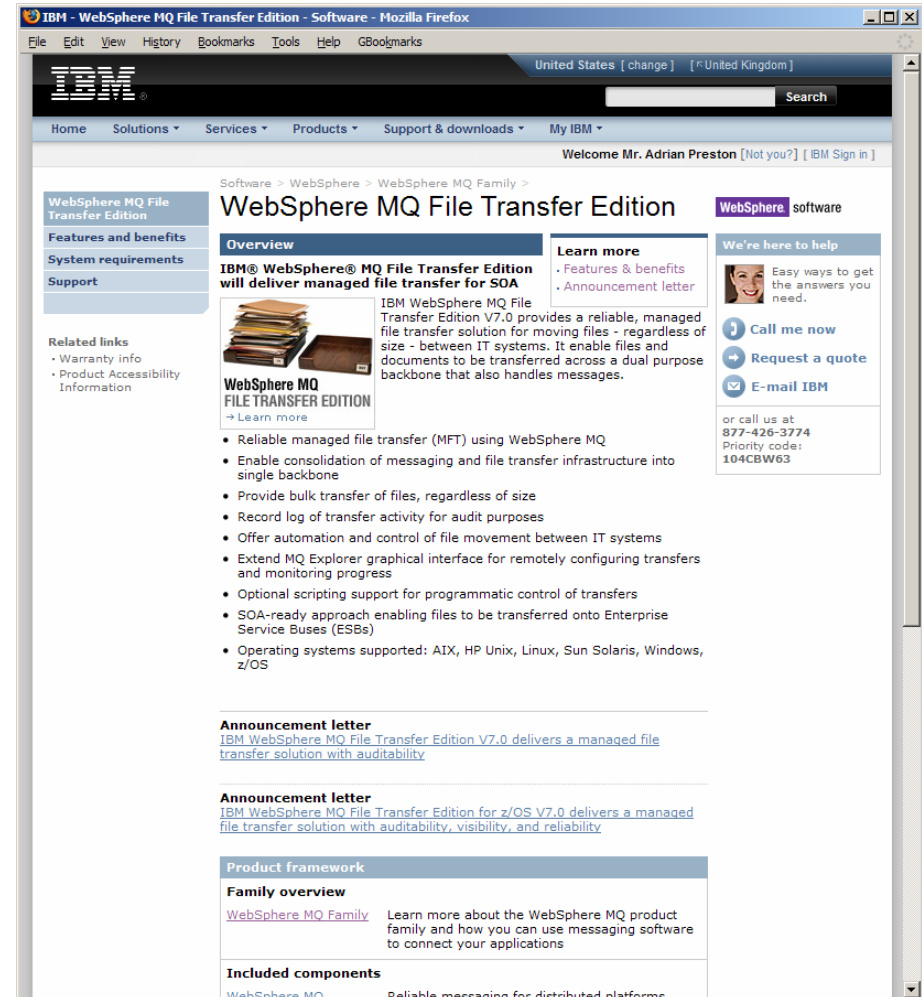
Configure the sample for your environment (typically these would not be seen by the end user)

- Transferred files are placed into the default transfer root of the destination agent inside a directory called 'webuploads'
 - Unless explicitly set using the 'transferRoot' property, this will be the *home* directory for the user ID that runs the destination agent process

Note to obtain the best user experience use Firefox or Chrome. Current versions of the SupportPac have restrictions when using Internet Explorer

Apply to Join the Early Access Program

- **What is it?**
 - Opportunity to get early access to pre-GA code
 - Influence product directions and features
 - Build product knowledge and skills
- **When does it start?**
 - You can apply to join anytime during 2010
- **What is required to join?**
 - A specific Non-Disclosure Agreement (NDA) must be signed
 - Can be signed electronically via the Betaworks Web site
- **What could I receive?**
 - Download regular code drops of the latest pre-GA iterations
 - Opportunity to provide product feedback direct to labs
 - Product roadmap details and materials
- **Who can I contact with more questions?**
 - Contact your local IBM representative



The screenshot shows the IBM WebSphere MQ File Transfer Edition software page. The browser window title is "IBM - WebSphere MQ File Transfer Edition - Software - Mozilla Firefox". The page features the IBM logo, a search bar, and navigation links for Home, Solutions, Services, Products, Support & downloads, and My IBM. The main content area is titled "WebSphere MQ File Transfer Edition" and includes an "Overview" section with a sub-heading "IBM® WebSphere® MQ File Transfer Edition will deliver managed file transfer for SOA". Below this is a list of features and benefits, such as "Reliable managed file transfer (MFT) using WebSphere MQ" and "Enable consolidation of messaging and file transfer infrastructure into single backbone". There are also "Announcement letter" sections with links to news articles. A sidebar on the left contains navigation links for "WebSphere MQ File Transfer Edition", "Features and benefits", "System requirements", and "Support". A sidebar on the right offers "We're here to help" with options like "Call me now", "Request a quote", and "E-mail IBM".

Resources

- Information Center:
 - <http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/index.jsp>
- Redbooks / Redguides / Redpapers:
 - Getting Started with WebSphere MQ File Transfer Edition V7
 - <http://www.redbooks.ibm.com/abstracts/sg247760.html>
 - IBM WebSphere MQ File Transfer Edition Solution Overview
 - <http://www.redbooks.ibm.com/abstracts/redp4532.html>
 - Managed File Transfer for SOA using IBM WebSphere MQ File Transfer Edition
 - <http://www.redbooks.ibm.com/abstracts/redp4533.html>
 - B2B Enabled Managed File Transfer using WebSphere DataPower B2B Appliance XB60 and WebSphere MQ File Transfer Edition
 - <http://www.redbooks.ibm.com/abstracts/redp4603.html>
- Trial Download:
 - <http://www.ibm.com/software/integration/wmq/filetransfer/>