All About WebSphere MQ File Transfer Edition

Lyn Elkins – elkinsc@us.ibm.com
IBM ATS

Agenda

- **What is Managed File Transfer?**
- **Introducing WebSphere MQ File Transfer Edition**
- **Key Components**
- **How File Transfer Edition uses MQ**
  - Different roles for queue managers
  - Key message exchanges
- **Walkthrough of key File Transfer Edition function**
  - Ways to initiate file transfers
  - Integrating with existing systems
- **Latest News**

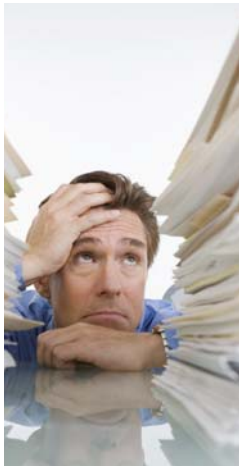
How do most organizations move files today?

- **Most organizations rely on a mix of homegrown code, several legacy products and different technologies … and even people!**
- **FTP**
  - Typically File Transfer Protocol (FTP) is combined with writing and maintaining homegrown code to address its limitations
- **Why is FTP use so widespread?**
  - FTP is widely available – Lowest common denominator
  - Promises a quick fix – repent at leisure
  - Simple concepts – low technical skills needed to get started
  - FTP products seem "free", simple, intuitive and ubiquitous

3

How do most organizations move files today?

- **Legacy File Transfer products**
  - A combination of products often used to provide silo solutions
  - Often based on proprietary versions of FTP protocol
  - Can't transport other forms of data besides files
  - Usually well integrated with B2B but rarely able to work with the rest of the IT infrastructure – especially with SOA
- **People**
  - From IT Staff to Business staff and even Security Personnel
  - Using a combination of email, fax, phone, mail, memory keys…



4

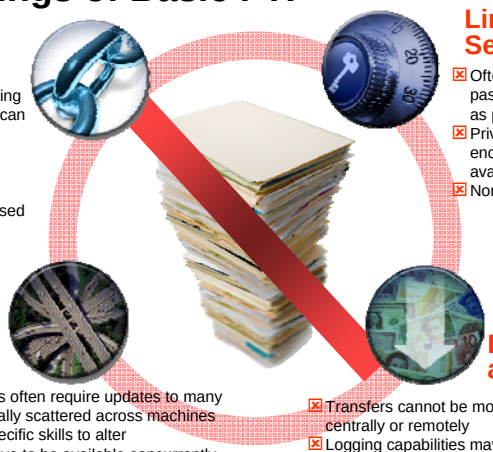
Shortcomings of Basic FTP

Limited Reliability

- Unreliable delivery – Lacking checkpoint restart – Files can be lost
- Transfers can terminate without notification or any record – corrupt or partial files can be accidentally used
- File data can be unusable after transfer – lack of Character Set conversion

Limited Security

- Often usernames and passwords are sent with file – as plain text!
- Privacy, authentication and encryption often not be available
- Non-repudiation often lacking



Limited Flexibility

- Changes to file transfers often require updates to many ftp scripts that are typically scattered across machines and require platform-specific skills to alter
- All resources usually have to be available concurrently
- Often only one ftp transfer can run at a time
- Typically transfers cannot be prioritized

Limited visibility and traceability

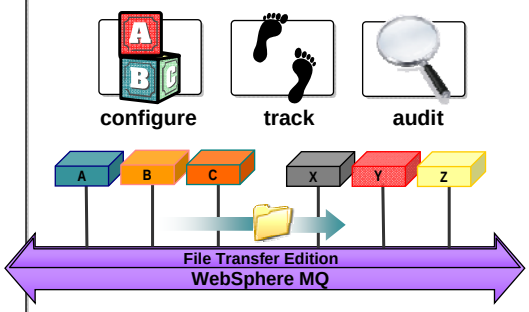
- Transfers cannot be monitored and managed centrally or remotely
- Logging capabilities may be limited and may only record transfers between directly connected systems
- Cannot track the entire journey of files – not just from one machine to the next but from the start of its journey to its final destination

SHARE in Anaheim 2011

Introducing WebSphere MQ File Transfer Edition V7

- **Adds file transfer to WebSphere MQ to enable movement of files – regardless of size – in a managed way (reliable, auditable, secure)**
- **Multi-purpose infrastructure – for both files and messages**

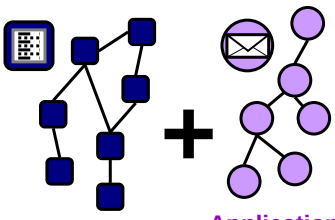
- Flexible backbone for transfers – move files from anywhere to anywhere in your network
- Multi-purpose use – for both files and messages
- Auditable with logging subsystem that tracks transfer at source and at destination for audit purposes
- Centralized control and configuration
- Integration with MQ-enabled apps and ESBs
- No need to program – no need to use APIs
- Automatic file conversion and compression
- Security – of file payload using SSL



SHARE in Anaheim 2011

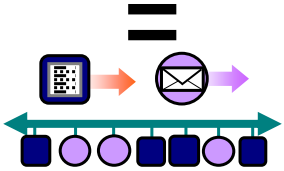
A Consolidated Transport for Files and Messages

- **Traditional approaches to file transfer result in parallel infrastructures**
  - One for files – typically built on FTP
  - One for application messaging – based on WebSphere MQ, or similar
- **High degree of duplication in creating and maintaining the two infrastructures**
- **Consolidating messaging and file transports yields:**
  - Operational savings and simplification
  - Reduced administration effort
  - Reduced skills requirements and maintenance



File Transfers

Application Messaging

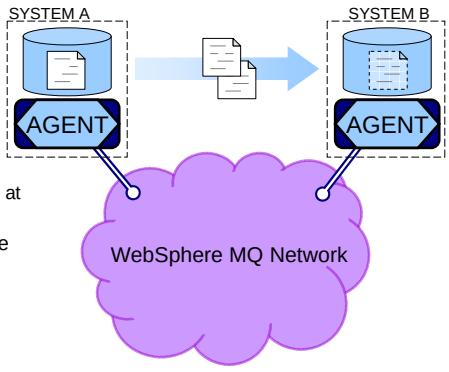


Single Transport for Messages & Files

7

3 Key Components of FTE - #1 Agents

- **Agents**
  - Long running MQ application
  - Transfers files using MQ
  - Run on system where files are to be transferred from / to
  - Multi-threaded file transfers
    - Can both send and receive multiple files at the same time
  - Agent always associated with exactly one queue manager
    - … but one queue manager can host several agents
  - Each agent monitors its own command queue for work



8

NOTES

3 Key Components of FTE - #1 Agents

The first of the three key components of File Transfer Edition (FTE) is the 'agent'

- **FTE agent processes define the end-points for file transfer. That is to say that if you want to move files off a machine, or onto a machine – that machine would typically need to be running an agent process**

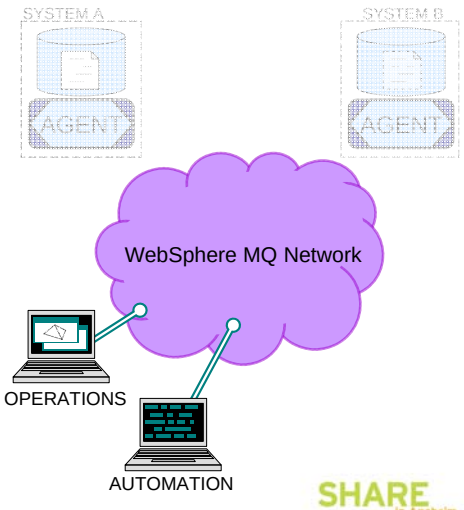
Agent processes are long running MQ applications that oversee the process of moving file data in a managed way. Each agent monitors a 'command' queue waiting for messages which instruct it to carry out work, for example file transfers

- **The FTE agent process needs connectivity to an MQ queue manager to do useful work. It can connect either directly to a queue manager running on the same system, or as an MQ client using an embedded version of the MQ client library (which is kept completely separate to any other MQ client libraries that may or may not already have been installed onto the system)**
  - Each agent requires its own set of MQ queues – which means that an agent is tied to the queue manager where these queues are defined
  - However – one queue manager can support multiple agents

9

3 Key Components of FTE - #2 Commands

1. **Agents**
2. Commands
   - Instruct agents. For example, to start or cancel a transfer
   - Use MQ messaging to relay the instructions
   - A queue manager that the commands connect to is playing the *command queue manager* role
   - GUI or command line or program



10

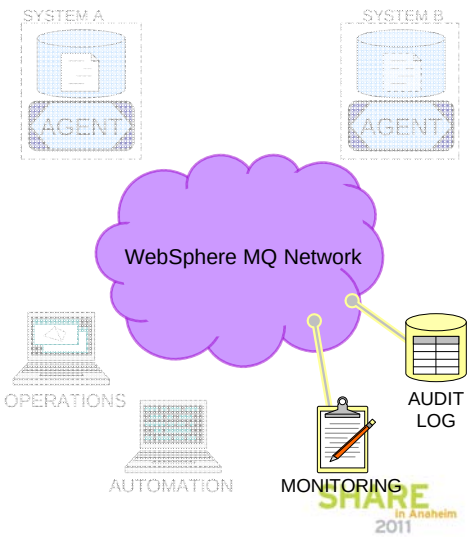
Key Components of FTE - #2 Commands

NOTES

- The second of the three key FTE components are ‘commands’.
- Here we are using the term ‘commands’ to describe anything which issues instructions to an agent process. Examples of ‘commands’ include the command-line, GUI and scripting interfaces that are provided as part of the FTE product.
- The ‘commands’ interact with FTE agents by sending them MQ messages containing instructions to carry out. The ‘commands’ do not need to be directly connected to the same MQ queue manager as the FTE agent as the messages that they send can be routed through the MQ network.

3 Key Components of FTE - #3 Logging/Monitoring

- Agents
- Commands
- Logging / Monitoring
  - Acts as a collection point for monitoring and logging data
    - How are my transfers progressing?
    - What files were transferred?
  - The queue manager that the logging / monitoring tools connect to is performing the coordination queue manager role
  - Coordination qmgr must be MQ V7
    - Uses publish-subscribe capabilities
- File data does not flow through coordination queue manager!



Key Components of FTE - #3 Monitoring

NOTES

- The third key component of FTE are the tools used for logging and monitoring.
- Examples of the logging and monitoring tools, that form part of the FTE product, include the WebSphere MQ Explorer based GUI and also the database logger which can be used to archive audit information into a database.
- Agents produce audit and monitoring data and forward this as messages to a particular queue manager which is performing the 'coordination queue manager' role. This queue manager then uses MQ publish/subscribe (so it must be an MQ v7 queue manager) to distribute this information to any subscribing monitoring / auditing programs.

Monitoring Messages

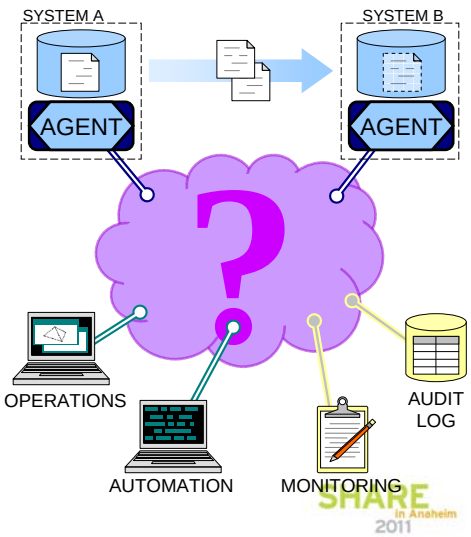
- Monitoring data published using topic string:
  - SYSTEM.FTE/Transfers/<*agent*>/<*transfer_id*>
- Messages are produced periodically during a file transfer
- Information gathered includes:
  - Bytes transferred / total bytes
  - Elapsed time
  - Current file / total files in transfer

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**

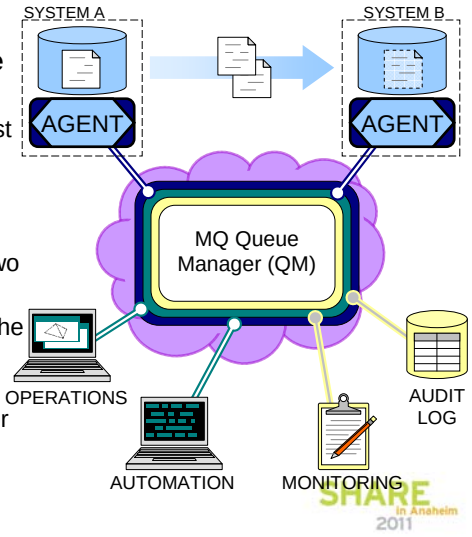
Integration with MQ Networks

- **How do I integrate this with my existing queue manager network?**
  - Let's look at some examples…



MQ Network Integration: Single Queue Manager

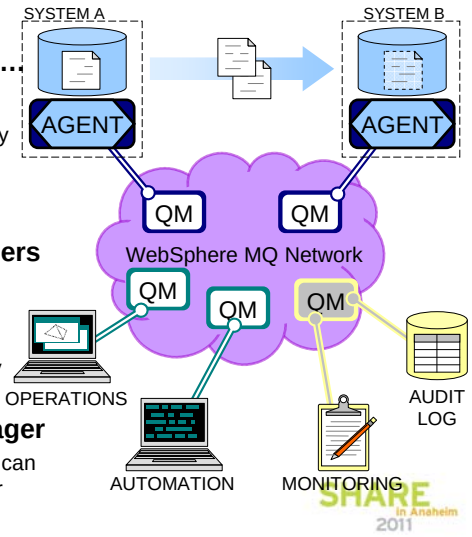
- **At one extreme, you can connect everything to a single queue manager…**
  - Most useful for prototyping or test systems
- **Here one queue manager is playing the following roles:**
  - Agent queue manager (for the two agents)
  - Command queue manager (for the operation and automation commands)
  - Coordination queue manager (for the audit and monitoring processes)



17

MQ Network Integration: Many Queue Managers

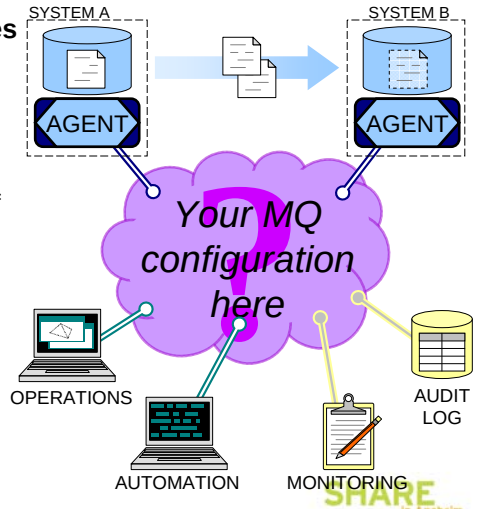
- **Or you can have one or more queue managers for each role…**
- **Many agent queue managers**
  - Each agent is associated with exactly one queue manager
  - One queue manager can host many agents
- **Many command queue managers**
  - Each instance of the commands is associated with exactly one queue manager
  - One queue manager can be used by many instances of the commands
- **One coordination queue manager**
  - Many monitoring / audit applications can use the coordination queue manager



18

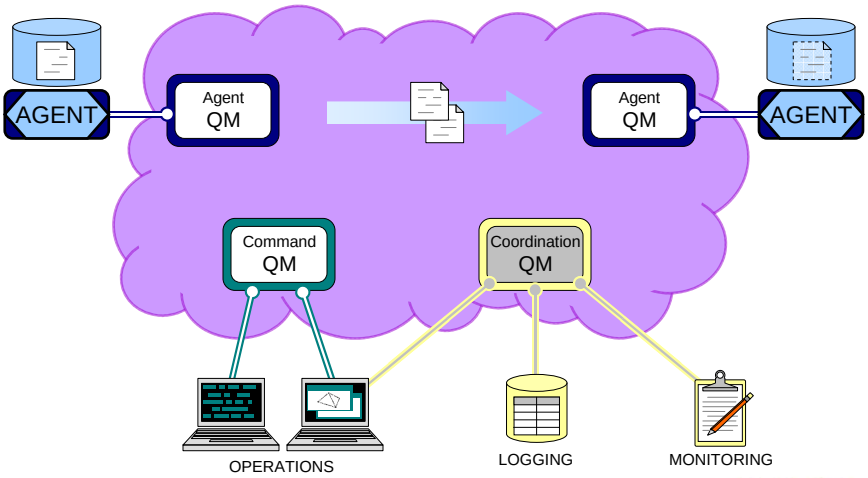
MQ Network Integration: And My Current Network?

- **File Transfer Edition integrates with existing MQ networks**
- **May want to add a coordination queue manager**
  - MQ V7 license comes as part of distributed File Transfer Edition Server product
- **Protocols designed to minimize impact on existing messaging networks**



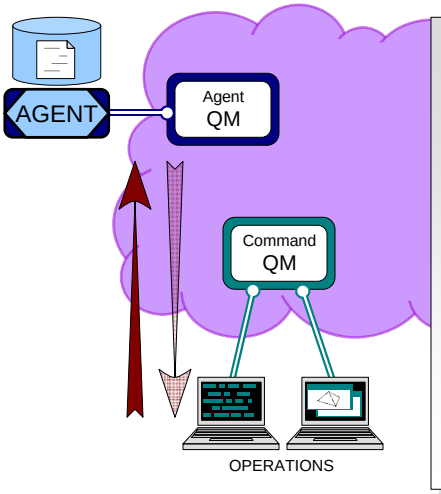
19

Message Flows



20

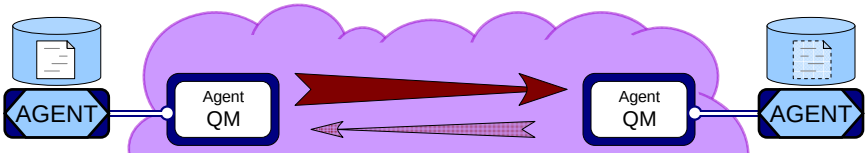
Message Flows: Initiating a Transfer



- **A new transfer is started by sending an MQ message to an agent**
  - The message may be routed via a *command queue manager*
- **The MQ message:**
  - Describes which files to transfer
  - Specifies the agent to which the files will be transferred
- **The agent responds by starting to transfer files, as instructed in the MQ message**
- **The agent can, optionally, reply**

21

Message Flows: Transferring File Data



- **Agents transfer file data as MQ messages**
  - File data sent as MQ non-persistent messages
  - Allows prioritization with existing messaging workloads
  - Protocol used accounts for non-delivery and re-ordering
- **Transfers are paced**
  - This avoids a backlog of messages building up
- **Transfers automatically check-point:**
  - If any part of the infrastructure suffers an outage, transfers automatically re-start from the last check-point

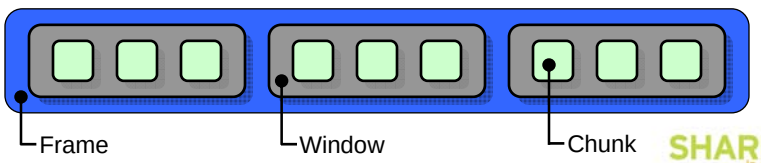
22

Concurrency and transfers

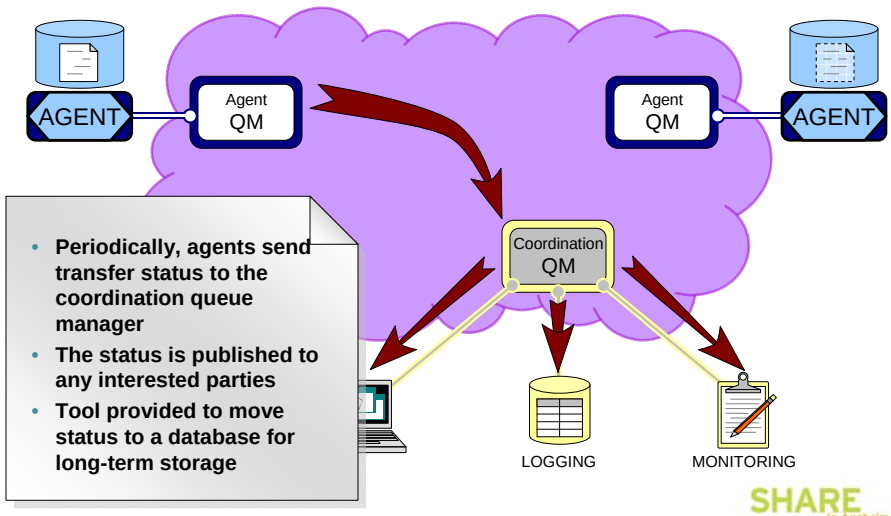
- **Multiple transfer requests are processed in parallel**
  - Up to a configurable maximum
- **Files specified in a transfer request are transferred in sequence**
- **However, be aware that:**
  - Too many transfers happening in parallel can reduce performance
  - Under load some sequential behaviour 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
    - Though this is very unlikely!

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



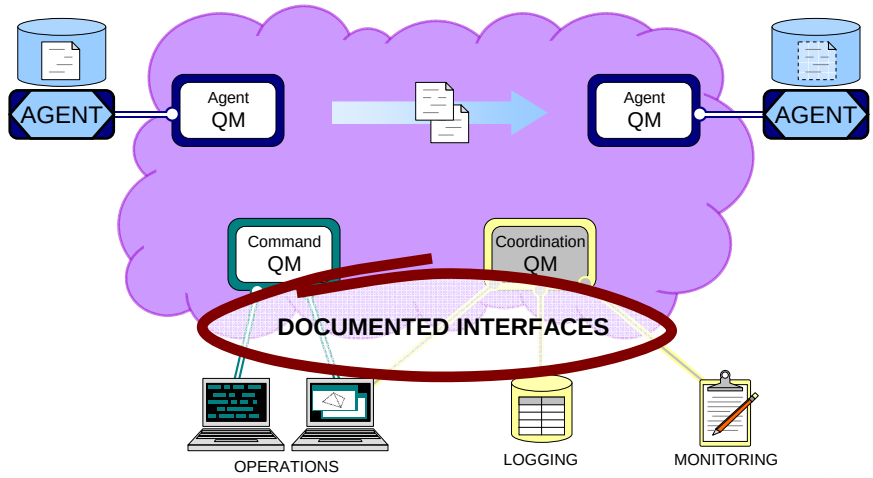
Message Flows: Log and Progress Data



- **Periodically, agents send transfer status to the coordination queue manager**
- **The status is published to any interested parties**
- **Tool provided to move status to a database for long-term storage**

25

Message Flows: Documented Interfaces



26

Ways to configure and control managed file transfers

Graphical Eclipse-based MQ Explorer

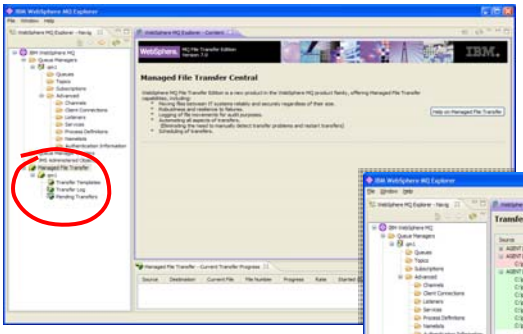
Command Line Interface

Job Control Language (JCL)

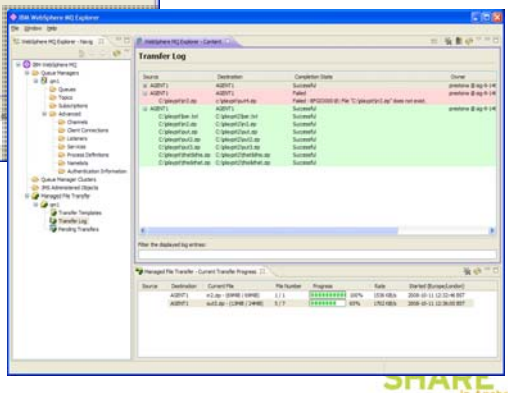
XML Scripts using Apache Ant

27

Centralized config using MQ Explorer



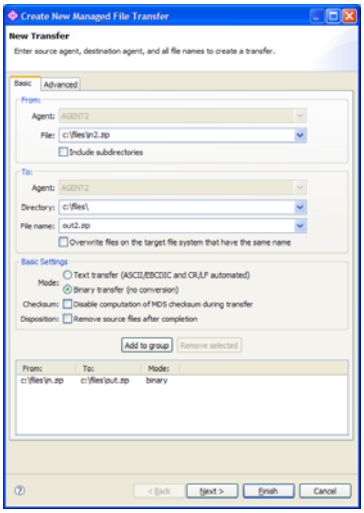
Eclipse GUI integrated into WebSphere MQ Explorer



- **Shows progress of current transfers and outcome of recent transfers**
- **3rd party and bespoke applications can also subscribe to these events**

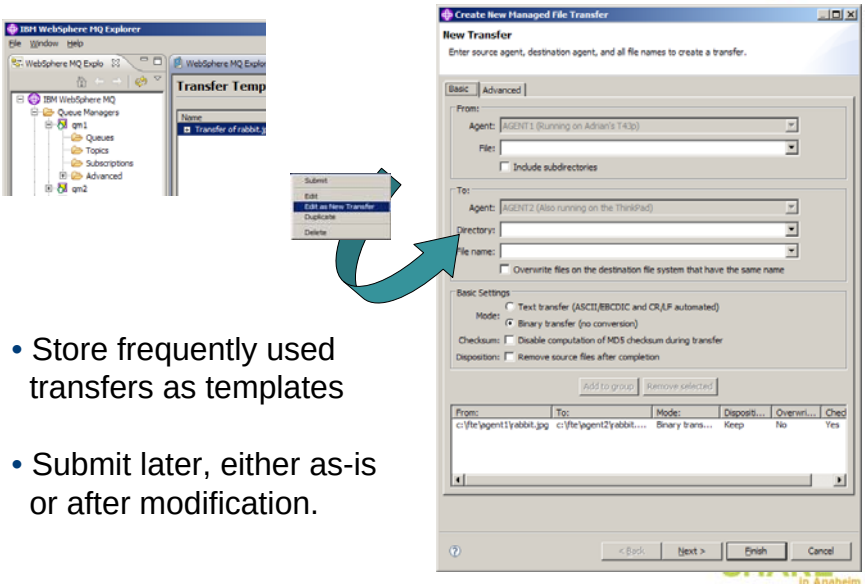
28

Managed file transfers using MQ Explorer



- **Graphical user interface that allows:**
  - Operators to create ad-hoc file transfers to satisfy a particular business need
- **File transfers can:**
  - Specify code page conversion
  - Group together multiple files into a single transfer
  - Recursively move directory structures

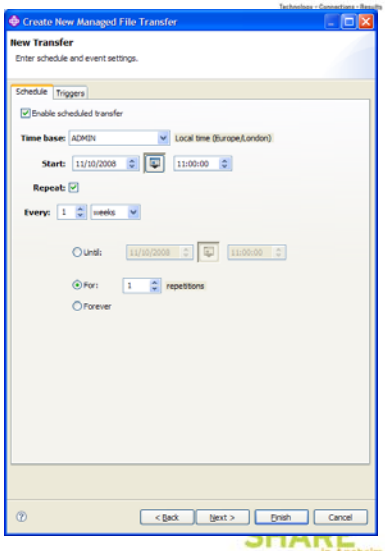
File transfers based on templates



- Store frequently used transfers as templates
- Submit later, either as-is or after modification.

Scheduled transfers

- **Transfers can be scheduled to take place at a specific time**
- **Scheduled transfers can be used to initiate a managed file transfer during a processing window**
- **Scheduled transfers can repeat:**
  - Every day, week, year etc.
  - For a specified number of repetitions
  - Until an end date/time

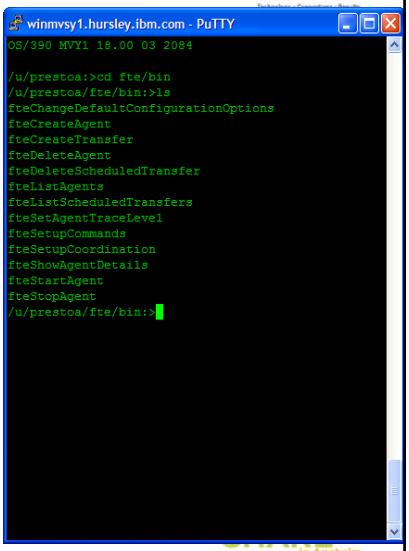


Directory Monitoring

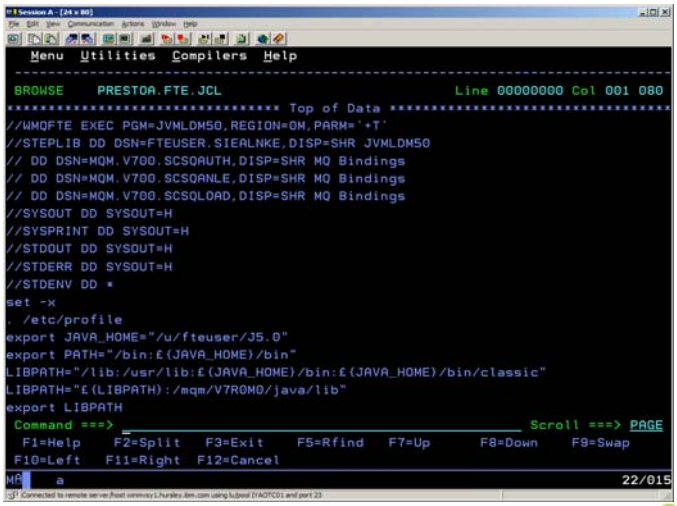
- **Monitor file system for presence of "trigger files" and start a file transfer operation**
- **Directory monitoring can be used to integrate with an application which produce files as its output – without needing changes to the existing application**
- **"Trigger files" can be:**
  - Specified using wildcards
  - Found by recursively searching directory trees
- **Attributes of file transfer can be inferred from trigger file, for example:**
  - Transfer all files from same directory as trigger file
  - Select destination agent name from parent directory of trigger file
  - Name destination file with timestamp of trigger file
  - Etc.
- **Supports extension via user exit routines**

Via the Command Line...

- **Equivalent function to the GUI is also scriptable from the command line**
- **Administrative commands:**
  - Define and delete agents
  - Configure agents
  - List and show details about agents
  - Start and stop agents
  - Create and delete scheduled transfers
  - Create and delete directory monitors
- **Operational commands:**
  - Start transfers
  - Cancel transfers
  - List in-progress transfers

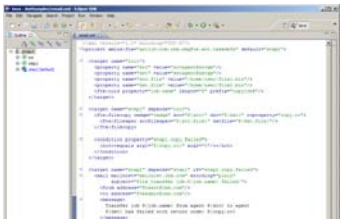


... and also Job Control Language (JCL)



Scripting

- **Script together complex sequences of transfer operations which execute conditionally**
  - E.g. If step1 successful do step2 else email administrator
- **Use scripting to call out to other programs to perform pre/post processing of file data**
  - E.g. Run a program prior to the transfer to generate the file, or after the transfer to process the file
  - The output from any programs run is captured in the transfer audit information



- **Based on the Apache Ant language which is:**
  - Open… (so you can get a book on it…)
  - ..and extensible (so you can use Ant tasks developed elsewhere)

Security Controls

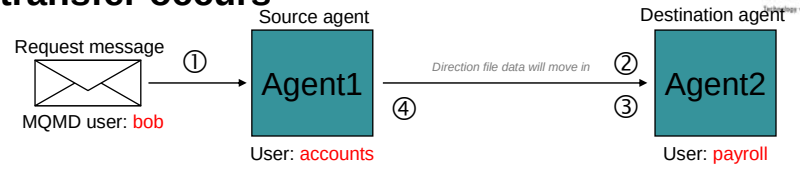
Access Control: Overview

- **WebSphere MQ already provides access control that can be used to prevent unauthorized users from accessing MQ objects (such as queues)**
- **File Transfer Edition extends this to include authorities that relate to file transfer operations (e.g. should this user be able to transfer files from this system?)**

Granular Access Control

- **Access control to agent capabilities can be broken down into steps:**
- **Determine a user's identity**
  - (MQMD user ID of request message)
- **Work out what action is being taken**
  - (Parse payload of request message)
- **Map what they are trying to do to one (or more) FTE authorities**
  - (Simple 'look-up' table in the code)
- **(Optionally) determine the agent's identity**
  - (MQMD user ID of messages sent by the agent)
- **Check to see if the identities have the appropriate authorities**
  - (Map FTE authority to MQ authority and see if the user is authorized)
- **Permit or deny the action**
  - (Either carry on as normal, or fail the request)

Example authority checks before transfer occurs



- **Checks that occur before the transfer starts:**
  - Does 'bob' have 'transfer source' authority?
    - (i.e. can bob move files off agent1?)
  - Does 'accounts' have 'agent source' authority?
    - (i.e. is 'agent2' going to allow 'agent1' to transfer files to it?)
  - Does 'bob' have 'transfer destination' authority?
    - (i.e. can bob move files onto agent2?)
  - Does 'payroll' have 'agent destination' authority?
    - (i.e. is 'agent1' going to allow 'agent2' to receive files from it?)
- **Checks 1+4 happen at the source agent, and 2+3 at destination agent**

Mapping FTE Authorities to MQ Authorities

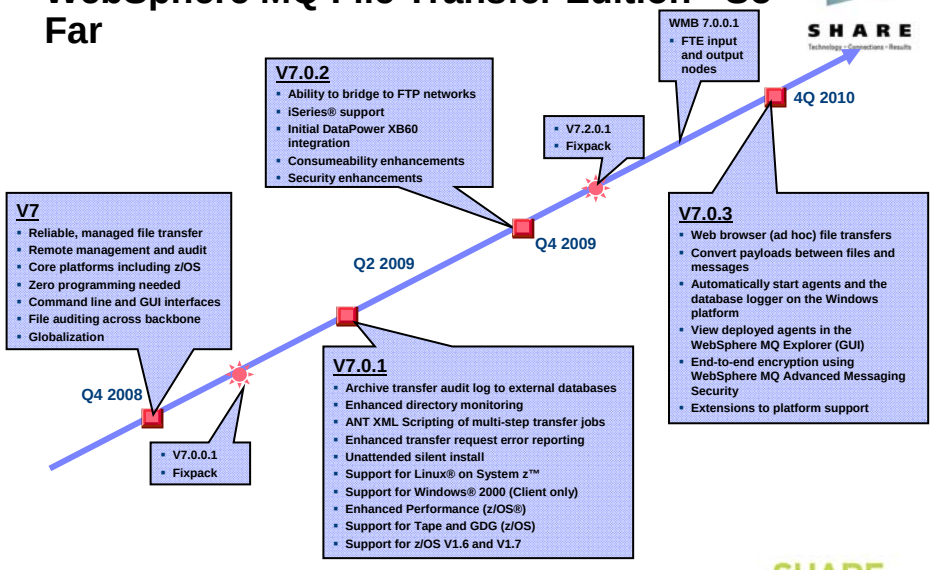
- **We have talked about FTE authorities (like 'transfer source' or 'schedule')**
  - But how does an administrator configure these?
- **FTE authorities are mapped to MQ authorities on specific MQ objects**
  - E.g. the FTE 'administration' authority maps to the MQ 'browse' authority on queue 'SYSTEM.FTE.AUTHADM1.agentname'.
- **The same model used for Distributed platforms (via the OAM) and for z/OS (via SAF)**

Sandboxes

- **"Sandboxes" is the terminology used to describe the mechanism that File Transfer Edition uses to restrict what areas of the file system can be accessed**
- **FTE originally allowed all agent file operations to be constrained to be within a 'sandbox' (one or more directories).**
- **But it imposed the following practical restrictions:**
  - It does not discriminate between read + write operations
  - The settings apply to the agent as a whole – there is no support for applying different settings at a per 'user' level.
- **FTE 7.0.2 removed these restrictions**
  - Read and write operations can be specified separately
  - Settings are at the per user (or collection of user) level
- **Defined by an XML document in the agent configuration directory**

- **That was the basic coverage of what FTE does and how it does it**
- **But we also need to go beyond that …**
- **7.0.2 brings more connectivity to file-based systems**
- **7.0.3 brings more connectivity to non-file (message-based) applications**

WebSphere MQ File Transfer Edition - So Far



Options for integration with existing systems

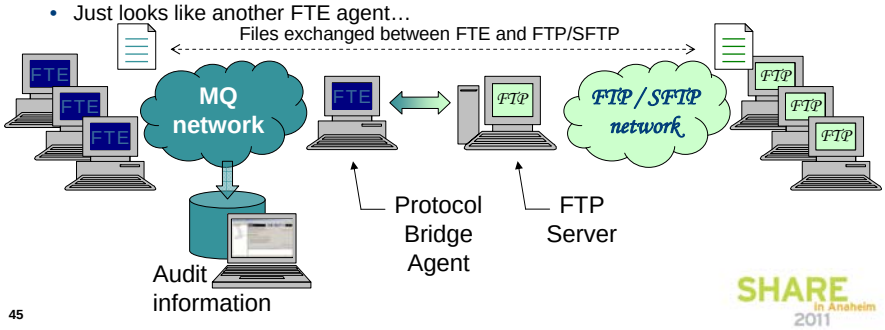
Integrate through the use of directory monitoring and scripting

Connect to existing FTP/SFTP servers using Protocol Bridging

Transfer files outside your enterprise using the DataPower XB60 appliance

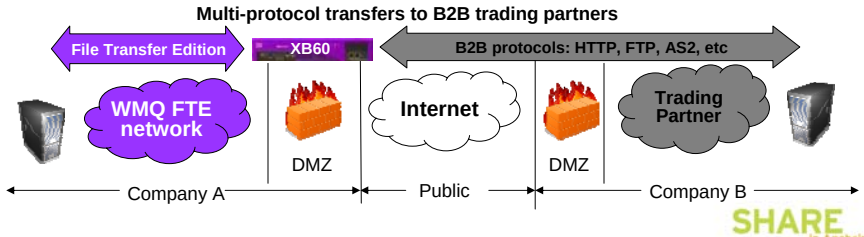
Protocol Bridging Agents

- **Support for transferring files located on FTP and SFTP servers**
  - The source or destination for a transfer can be an FTP or an SFTP server
- **Enables incremental modernization of FTP-based home-grown solutions**
  - Provides auditability of transfers across FTP/SFTP to central audit log
  - Ensures reliability of transfers across FTP/SFTP with checkpoint restart
- **Fully integrated into graphical, command line and XML scripting interfaces**
  - Just looks like another FTE agent…



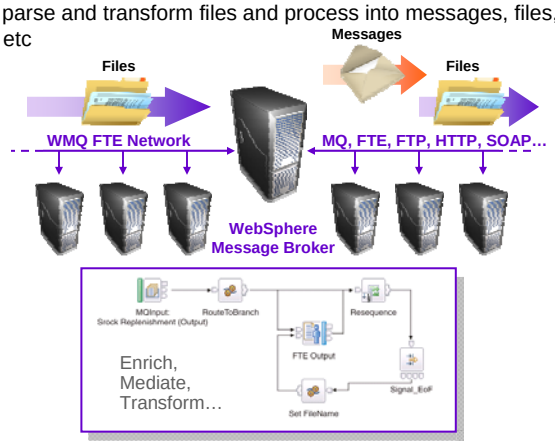
Interoperation with Datapower B2B Appliance XB60

- **Documented and tested configurations for integrating with DataPower Appliances**
  - WebSphere DataPower XB60 B2B Appliance – for B2B connectivity
  - WebSphere DataPower IX50 Integration Appliance – for ESB connectivity
- **Enables sending files to trading partners over a range of protocol transports**
  - via DataPower Appliances acting as B2B gateways

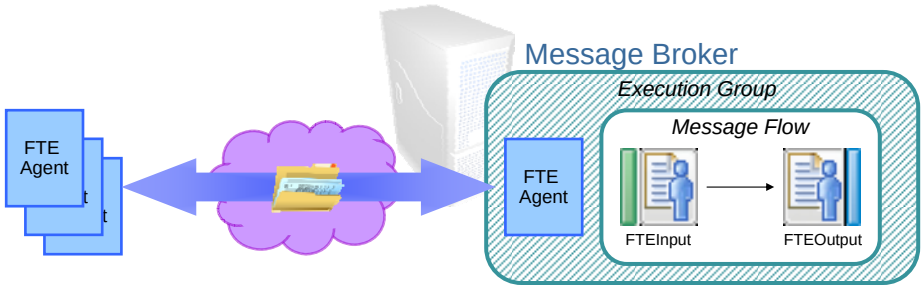


ESB File Processing

- **Integration with WebSphere Message Broker for File Processing**
  - Tight integration between FTE and WebSphere Message Broker
  - Enables ESB capabilities to be applied to file data
  - Ability to parse and transform files and process into messages, files, events, service requests etc



WMB FTEInput and FTEOutput nodes



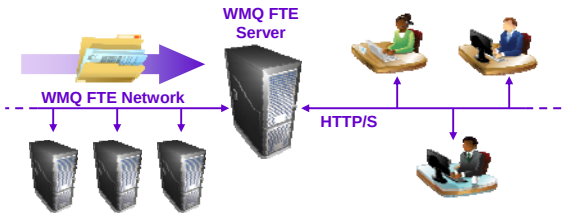
- **FTEInput node**
  - Build flows that accepts file transfers from the WMQ FTE network
- **FTEOutput node**
  - Build flows that are designed to send a file across a WMQ FTE network
- **When WMQ FTE nodes are used in a flow an FTE agent is automatically started in the Message Broker Execution Group**

V7.0.3 at a glance…

- **Key new features in WebSphere MQ File Transfer Edition V7.0.3**
  - A Web Gateway component that provides a web-based interface for users that wish to exchange file data with a WMQ FTE network
  - Convert payloads between file and messages (AKA “file to message” or “message to file”)
  - A Windows Service to automatically start the WMQ FTE agent and WMQ FTE database at system start-up time
  - View deployed WMQFTE Agent in the WebSphere MQ Explorer GUI including their current status.
  - Encrypt file data in the payloads of messages resting on WebSphere MQ queues using WebSphere Advanced Message Security v7.0.1
  - Database logger component can now be deployed and managed as a JEE application
  - Support for new platforms including Solaris on x86 hardware

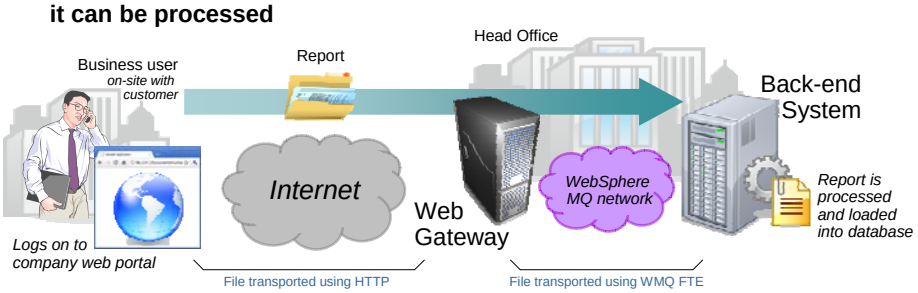
Web based File Transfers using the Web Gateway

- **Web-based File Transfer**
  - A RESTful API for sending files into and receiving files from a WMQ FTE network
  - Reliable and secure file transfer option for Web users
  - Auditable transfer and large file support
  - Zero-footprint file transfer support without the need to provision and install code
  - Interfaces for embedding into third party and custom user applications



Web Gateway Example Scenario (1)

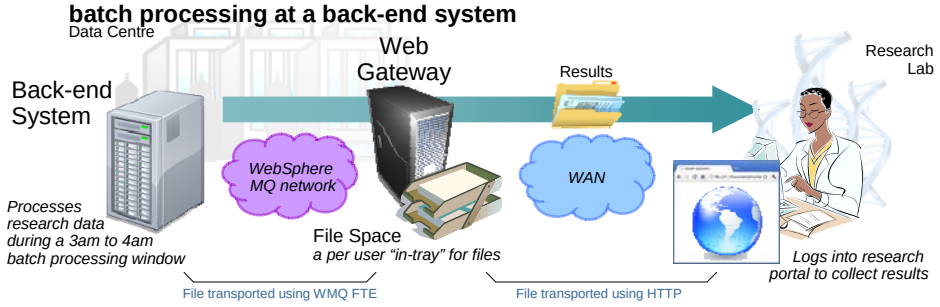
- **In this example usage scenario the Web Gateway allows a business user to upload a file (via the company web portal) to a back-end system where it can be processed**



1. The business user logs onto the company web portal using a web browser and is prompted to select a file to upload
2. The portal uses the RESTful API provided by the Web Gateway to upload the file using HTTP
3. The Web Gateway transfers the file, using WMQ FTE, to a back-end system
4. At the back-end system WMQ FTE starts a program to process the data from the file

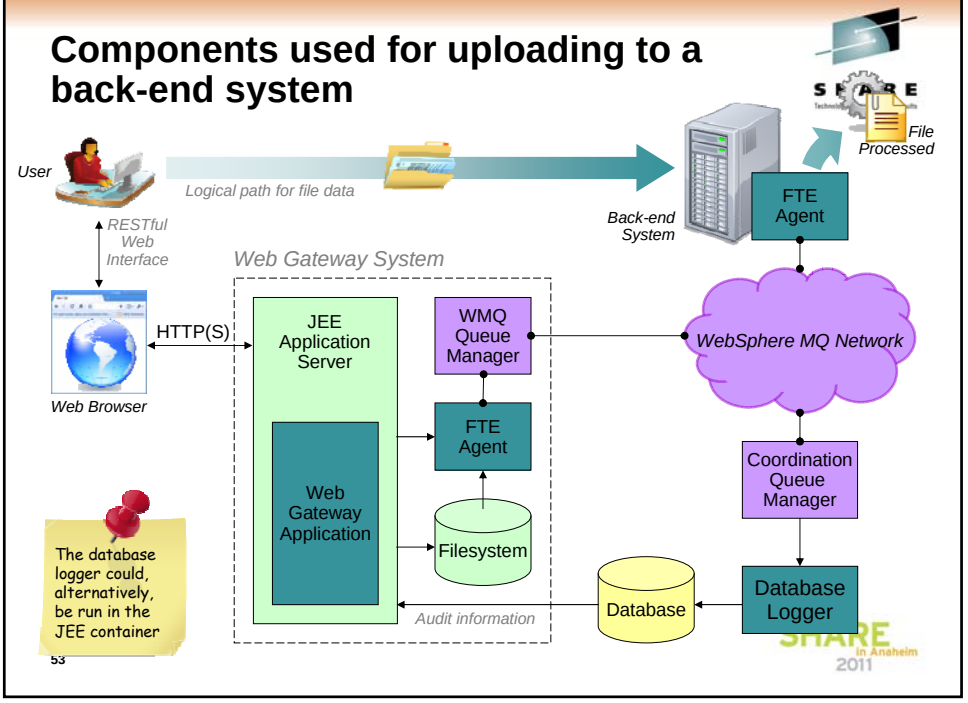
Web Gateway Example Scenario (2)

- **In this example usage scenario the Web Gateway is used to enable a researcher to pick up files that have been produced (hours earlier) by batch processing at a back-end system**

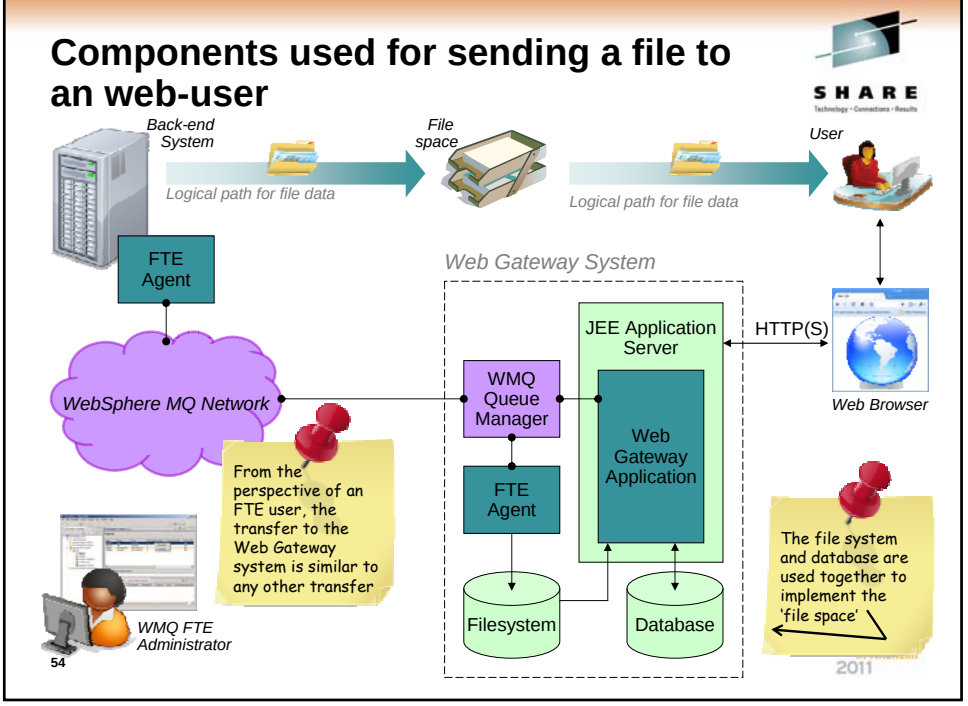


1. A batch process running at the data centre produces a set of results which it sends, using WMQ FTE, to the web gateway
2. The Web Gateway system places the data into a *file space* where it awaits collection by the user
3. The user logs in to the research portal using her web browser and is shown a list of files waiting for her attention
4. The user selects a file to download and the Web Gateway transfers the file to her computer

Components used for uploading to a back-end system



Components used for sending a file to an web-user



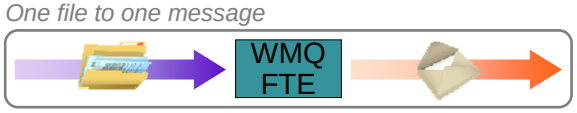
Batch Modernization

- **Convert payloads between files and messages**
  - Enables simple conversion of payload between files and messages
  - Helps modernize batch-oriented architectures into micro-batches and ultimately messaging
  - Readily and rapidly connect file-based and message enabled applications

55

Options to convert data between files and messages

One file to one message



- **One file becomes one message**

One file to a group of messages

- **File can be split based on:**
  - Size
  - Binary delimiter
  - Regular expression

One message to one file

- **One message becomes one file**

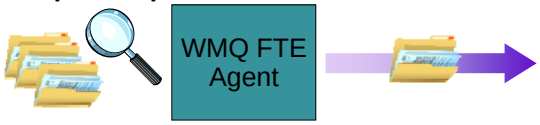
A group of messages (or all messages on the queue) to one file

- **Optionally, a delimiter can be inserted between each message composing a file**

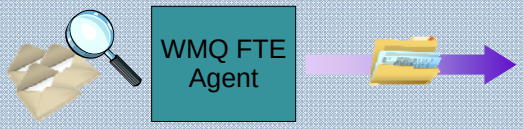
56

Monitoring queues for the arrival of messages

- The WMQ FTE agent can monitor queues for the arrival of messages, then transfer the payload from the messages as a file – as per the previous slide



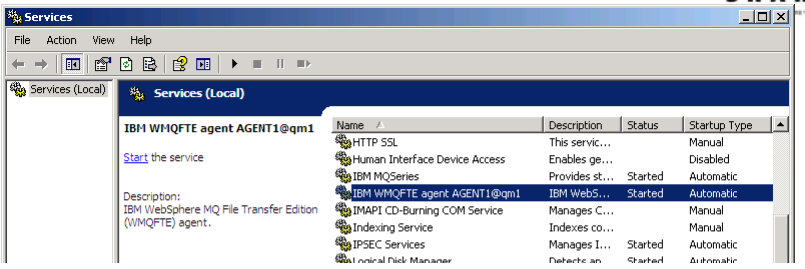
- WMQ FTE already supports monitoring a directory for new files to arrive – then transferring them



- V7.0.3 extends this concept to include monitoring queues for the arrival of messages or groups of messages

57

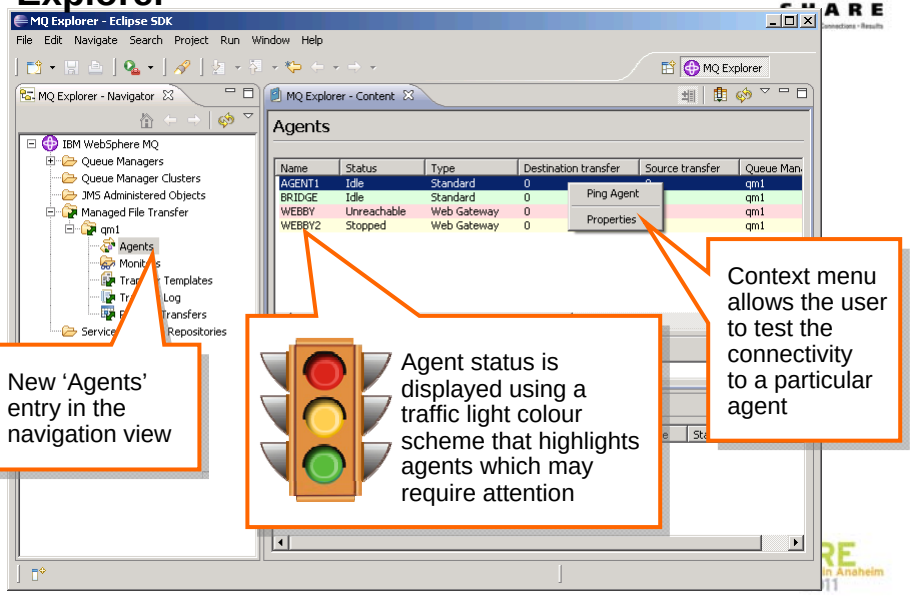
Agents & Database Logger as Windows Service



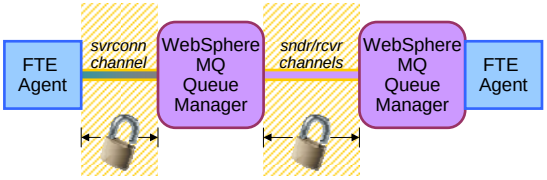
- Increases availability of FTE agent and Database Logger components by:
  - Automatically starting the components when the Windows system starts-up
  - Automatically restarting the component after a failure

58

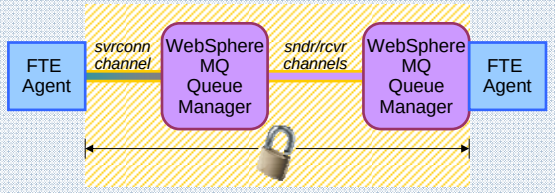
Monitoring deployed agents from the MQ Explorer



End-to-end encryption using WebSphere MQ Advanced Message Security

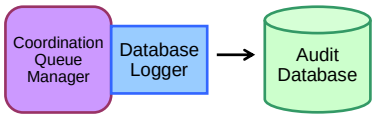


- **WMQ FTE already supports transport level encryption using SSL**
- **Data is encrypted before it is sent over a channel and decrypted when it is received**

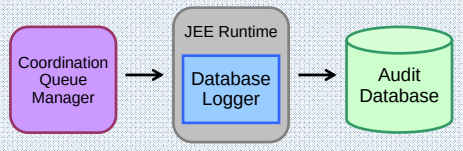


- **V7.0.3 (when combined with WMQ AMS v7.0.1) allows file data to be encrypted at the source system and only decrypted when it reaches the destination system**
  - This helps reduce encryption costs
  - Data is secure even when at rest on a queue

Alternate runtime for Database Logger component



- **WMQ FTE already provides a component for archiving the 'audit' type information about file transfers into a database**
  - This component runs as a stand-alone operating system process



- **V7.0.3 also provides the option of running the database logger component as a JEE application**
  - Simplifies the Web Gateway deployments
  - Re-use of JEE administration skills

61

Thank you

www.ibm.com/webspheremq/filetransfer

62