



WebSphere MQ Express Center

Contents

WebSphere MQ Express Center 1

Using the HTML Help version	1
Using the PDF version	2
Accessibility	2
Trademarks.	2
Notices	3
Product overview.	4
Usage restrictions.	4
Frequently asked questions	5
Product tour	6
Business needs.	6
Introduction to messaging and queuing	8
WebSphere MQ messaging fundamentals	12
Interacting with WebSphere MQ	16
Express File Transfer	19

Preparing WebSphere MQ for use with Express	
File Transfer	20
Using the graphical interface	21
Using the command line interface	26
How Express File Transfer works	34
Understanding return codes	34
Frequently asked questions (FAQ).	36
Tutorials	37
Sending a message to a local queue	37
Sending a message to a remote queue	40
Sending a message on a client-server	
configuration	46
Exchanging files using Express File Transfer	49

Index 53

WebSphere MQ Express Center

Welcome to the WebSphere MQ Express Center:

- *Product overview* describes the product package and prerequisites.
- *Product tour* provides the text from each part of the interactive Product Tour along with a description of the animation. This is included as an accessibility feature for the visually impaired (a screen reader is required).
- *Express File Transfer* contains help for this application which allows users to transfer files in the form of MQ messages.
- *Tutorials* provides training on basic MQ tasks such as setting up queue managers and queues, sending messages, and using the Express File Transfer.

The WebSphere MQ Express Center is provided for Windows users as a compiled HTML Help system (filename amqiceaw.chm), and for Linux users as a PDF document (filename amqiceaw.pdf).

For additional information about WebSphere MQ not covered in this WebSphere MQ Express Center, go to the *WebSphere MQ Help Center*, which contains the base WebSphere MQ product documentation.

Also refer to the text file readme, which you will find in the root directory on the product installation CD-ROM, for additional information about WebSphere MQ Express v5.3.

Please send your comments on the WebSphere MQ Express Center to idrcef@hursley.ibm.com

Using the HTML Help version

To view the HTML Help version, you must be running Windows. The information in the HTML Help version is presented as a hierarchy of *topics*, complete with navigation tree and links. To work with the HTML Help version, use the **Contents** and **Search** panes on the left. To access individual topics, click the topic title in the navigation tree.

To view the Windows HTML Help version of the Express Center comfortably, size the window so that it nearly fills the screen, and drag the left border of this topic to make the navigation pane on the left bigger. The pane retains the new size as its default the next time you open the Express Center. The following options are available while viewing the Express Centre.

- **Show/Hide.** Show or hide the navigation pane of the Express Center. If the navigation pane is currently hidden and you want to show it, click **Show**. If the navigation pane is currently shown and you want to hide it, click **Hide**.
- **Back.** Back out of the current topic to the previous topic.
- **Print.** Print a topic or a group of topics. When you click this option, a dialog allows you to specify whether you want to print the current topic, or the current topic plus all its subtopics.
- **Options.** Display the **Options** menu:
 - **Hide tabs.** Hide the **Contents** and **Search** tabs in the navigation pane.
 - **Back.** Back out from current topic to the previous topic.
 - **Forward.** Go forward to the next topic.
 - **Home.** Go to the home topic (the first topic in the Express Center).
 - **Refresh.** Refresh the screen.
 - **Internet options.** Go to the Internet options dialog, where you can alter the settings.
 - **Print.** Print the current page
 - **Search highlight on/off.** Switches word highlighting on or off when a word is located by a search.

- **Contents.** Display the contents navigation tree containing all topic titles. To view all the subtopics for a parent topic, click the + symbol to the left of the parent topic.
- **Search.** Search on a keyword. Click the **Search** tab, type the keyword into the text field, click the **List topics** button, and select the topic you want to view, from the list of topics that the search returns.

Right click or press SHIFT and F10 in the navigation tree left pane, to display the navigation context menu (**Open all, Close all, Print, View entry, Jump to URL**).

Right click or press SHIFT and F10 in the topic right pane, to display the topic context menu (**Back, Forward, Select all, View source, Print, Refresh, Properties**).

Using the PDF version

To view the PDF version on Windows or Linux, you need Adobe Acrobat 4.0. To download Acrobat, visit the Adobe website at: www.adobe.com/products/acrobat. The information in the PDF version is presented as a formatted online book, complete with table of contents and chapters. To work with the PDF version, open the PDF file and start at the table of contents.

To view the PDF version of the Express Center comfortably on Windows® or Linux, open the PDF file, and size the window so that it fills the screen area.

- Click **Help > Reader Guide** to get help on working with the PDF.
- Press **Page Down** to go to the next page. Press **Page Up** to go to the previous page. Alternatively, select these options on the **Document** menu, or click the arrow icons on the toolbar.
- Select **Zoom In** or **Zoom Out** on the **View** menu to enlarge or reduce the apparent size of the current page.
- To search for a word or phrase, click the toolbar **Find** icon (a pair of binoculars).

Accessibility

The Express Center provides the normal Windows keyboard accelerators and shortcuts. Access these keys in the usual way. See also the Microsoft Windows help for more information (look in the Windows help index for keyboard; for accessibility features look for Accessibility).

A text-based version of the Express Product Tour is included in this Express Center to improve accessibility. The text version of the tour provides the same information as the Macromedia Shockwave Flash interactive version of the tour, but without the animated graphics.

The Express File Transfer graphical interface has the following known accessibility issues:

Send/Receive panels

Panel headings (e.g. "Send", "Receive") are not revealed to screen readers.

Setup Panel

List item locations (for example, "item 5 of 20") are not revealed to screen readers.

List item checkbox status is not revealed to screen readers.

Users who require a fully accessible interface for this application are recommended to use the command line interface provided.

Trademarks

The following are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM iSeries Java Lotus Lotus Notes Microsoft SupportPac UNIX WebSphere Windows z/OS

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Lotus, Lotus Notes and LotusScript are trademarks of Lotus Development Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows 2000, Windows XP, Windows 2003, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries. Other company, product, and service names may be trademarks or service marks of others.

Adobe and Acrobat are trademarks of Adobe Systems Incorporated.

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

Notices

This information was developed for products and services offered in the United States. IBM may not offer the products, services, or features discussed in this information in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

- IBM World Trade Asia Corporation
- Licensing
- 2-31 Roppongi 3-chome, Minato-ku
- Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

- IBM United Kingdom Laboratories,
- Mail Point 151,
- Hursley Park,
- Winchester,
- Hampshire,
- England
- SO21 2JN.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Product overview

The WebSphere MQ Express v5.3 product package includes:

- *a server component* - a program for running queue managers and channels on your computer and for connecting to other computers over a network. Java support is included.
- *a client component* - a program that is a small subset of WebSphere MQ Express v5.3 without a queue manager. The client uses the queue manager and queues on other server computers. It must be used when the computer on which it is installed is connected to another computer running the WebSphere MQ Express v5.3 server program (the client and server programs can be on the same computer if required).
- *Express File Transfer* - a simple application for sending and receiving files over an MQ network. As well as demonstrating the practical use of WebSphere MQ, it should also prove useful as a day-to-day business tool.
- a set of *sample programs* - including amqsput for putting messages on queues, and amqsget for getting messages from queues.
- the *Express Product Tour* - a Shockwave Flash interactive product presentation.

Usage restrictions

Feature	Maximum	Comments
Channel connections	10	A single channel pair between two queue managers counts as 2 channel connections. Although the restriction of 10 applies, this is a WebSphere MQ Express v5.3 licensing issue (there is no theoretical limit on the maximum number of channel definitions that can be created).

Feature	Maximum	Comments
Client channel connections	10	Although the restriction of 10 applies, this is also a WebSphere MQ Express v5.3 licensing issue (there is no theoretical limit on the maximum number of client channel connections that can be created).
Message size	4 Mb	This is the maximum size of any given <i>message</i> that can be sent, not the maximum size of a file that can be sent (files of any size can be sent; they are chunked and sent as multiple MQ messages which are then 'reassembled' into files at the receiving end).

Frequently asked questions

What makes WebSphere MQ Express v5.3 different from the WebSphere MQ 5.3 product?	WebSphere MQ Express v5.3 is designed specifically for small to medium sized businesses (SMBs). Certain restrictions exist on the number of channels that can be run concurrently, and the connection of transactional clients to an Express server is not possible. Additional usability features and a sample application for sending and receiving files are also included.
What new features does WebSphere MQ Express v5.3 include?	<p>As well as providing the standard WebSphere MQ 5.3 functionality for creating and managing queue managers, queues and channels, and for sending and receiving messages, the product also includes the following additional features:</p> <ul style="list-style-type: none"> • <i>Product Tour</i> - an interactive Macromedia Flash Shockwave presentation that describes the product. • <i>File Transfer application</i> - a simple messaging application for sending and receiving files (this application includes a graphical user interface as well as a command line interface). • <i>Express Center</i> - a comprehensive information resource provided in compiled HTML Help format for Windows, and PDF format for both Windows and Linux.
Why was WebSphere MQ Express v5.3 developed?	To bring WebSphere MQ messaging capabilities within the reach of small to medium sized businesses.
Where can I get additional information and support?	<p>You can find more information on the WebSphere MQ Web site at: www.ibm.com/software/integration/mqfamily/</p> <p>The SupportPac Web page is at: www.ibm.com/software/integration/support/supportpacs/</p> <p>For current information on known problems and available fixes, see the Support page of the WebSphere MQ Web site at: www.ibm.com/software/integration/mqfamily/support/</p> <p>The latest updates to the Web-based WebSphere MQ documentation are now available from the WebSphere MQ Web site at: www.ibm.com/software/integration/mqfamily/library/manualsa/</p> <p>An online version of this readme file is maintained on the Web, and might contain additions made since this file was frozen on this media. The site is at: www.ibm.com/software/integration/mqfamily/support/readme/</p>

Product tour

This section contains a text version of the interactive Express Product Tour. It is included here as an accessibility feature for visually impaired users who have access to a screen reader. Information about each topic covered in the tour is provided in a two-column table, with the tour text in the left column, and a description of the animation in the right column.

The tour aims to provide an overview of the key concepts and interfaces related to WebSphere MQ messaging.

- *Business Needs* - introduces the ways in which this product can work for your business. The topics covered are integration, asynchrony, assured delivery, and scalability.
- *Introduction to Messaging and Queuing* - introduces the concepts of messages and queues, shows how they are used together, and includes a simple messaging scenario.
- *WebSphere MQ Fundamentals* - introduces the concepts that underpin WebSphere MQ messaging environments. These include queue managers, MQ clients, messaging communications, and queue manager clustering.
- *Interacting with WebSphere MQ* - introduces the WebSphere MQ administration interfaces, the messaging interfaces available for programs to use, and a simple file transfer tool.

The interactive version of the tour is provided in two different formats: as a Shockwave Flash file (.swf), and as a self-contained executable file (.exe).

- The Shockwave flash file requires Macromedia Flash Player 6 to be installed on the system where the file will run. The player can be freely downloaded from the Macromedia website as a plug-in for an internet browser, and is therefore not included with WebSphere MQ Express v5.3. For Windows, player plug-ins are available for Internet Explorer 4.0 or later, Netscape Navigator 4 or later, and Netscape 6.3 or later. For Linux, player plug-ins are available for Mozilla version 1.1.
- The self-contained executable file contains all the code required to launch a window and run the Product Tour in the window. No prerequisites are required for running the executable file, but you can only run it on Windows.

To launch the tour, open the compiled HTML Help version of the WebSphere MQ Express Center, and click the button at the bottom of this topic (the button is not visible if you are viewing the Express Center as a PDF document). To launch the tour on Linux, locate the Shockwave Flash file in the /opt/mqm/express directory, and open it in your Mozilla 1.1 (or later) browser.

Business needs

This part of the tour shows how the product can work for your business. The topics covered here are integration, asynchrony, assured delivery, and scalability. WebSphere MQ meets these business needs, and provides an ideal business information messaging and queuing environment.

Integration

Product tour text	Product tour animation
<p>Typical business applications consist of groups of intercommunicating programs. Communication is harder to achieve when these programs:</p> <ul style="list-style-type: none"> • span business boundaries • are written in different languages • are hosted on different operating systems or processors • use different communications protocols <p>There is a need for a common method of communication, unaffected by languages, operating systems, and protocols.</p>	<p>This part of the animation shows how WebSphere MQ allows the integration of distributed business applications.</p> <p>Two programs appear, and a connection is created between them. A unit of information (a message) is passed over the connection from the first program to the second one.</p> <p>The second program is now enclosed by a business boundary.</p> <p>The names of the languages that the programs might be written in appear: C++, COBOL, Visual Basic, Java, PL/1, C#.</p> <p>The names of the operating systems on which the programs might be running appear: Windows, Linux, z/OS, iSeries.</p> <p>The names of the communications protocols that might be used appear: HTTP, TCP/IP (SSL), LU6.2.</p> <p>A summary animation shows a message being sent from the first program to the second, across a business boundary, regardless of programming language, operating system or communications protocol.</p>

Asynchrony

Product tour text	Product tour animation
<p>Synchronous communication requires programs to be available at the same time, otherwise some programs are prevented from doing work until others become available.</p> <p>There is a need for programs to communicate independently of each other (that is, to use asynchronous communication).</p>	<p>This part of the animation shows how WebSphere MQ supports synchronous communication between programs.</p> <p>Messages are passed from the sending program to the receiving program, while the receiving program is available.</p> <p>The receiving program suddenly becomes unavailable.</p> <p>Messages now sent to the receiving program are stored on the receiving machine (outside the program).</p> <p>The receiving program becomes available again.</p> <p>The messages held on the receiving machine are transferred to the program.</p> <p>Note that messages were still be sent, even while the receiving program was unavailable.</p>

Assured delivery

Product tour text	Product tour animation
<p>When all or part of a business system fails, the integrity of the information in transit can be compromised:</p> <ul style="list-style-type: none">• information can be lost if it is not safely stored while enroute• information can be duplicated if resent unnecessarily <p>There is a need to assure that information is not lost in transit or duplicated.</p>	<p>This part of the animation shows how WebSphere MQ provides assured delivery.</p> <p>Messages are sent to the receiving program without using assured delivery. The receiving program becomes unavailable then available again, but the sent messages are lost.</p> <p>Messages are sent to the receiving program without using assured delivery. The sent messages are stored on the receiver. The receiving program becomes unavailable, then available again. The sending program resends the same messages which are now inadvertently duplicated.</p> <p>Messages are now sent using assured delivery. Messages are stored on a queue at the receiver and are therefore not lost. Messages are sent once and once only and are therefore not duplicated.</p>

Scalability

Product tour text	Product tour animation
<p>As businesses expand their operations (for example, when customer demand grows) they may require additional systems. However, the time and effort required to connect a new system into an existing network may be significant, and may require substantial downtime, interrupting customer service.</p> <p>There is a need for a way to integrate new systems with minimal interruption to service.</p>	<p>This part of the tour shows how WebSphere MQ provides a scalable messaging environment.</p> <p>Three external programs are connected to an existing business program, across the business boundary. The three external programs represent an increase in business.</p> <p>A second business program is added alongside the existing one, inside the business boundary.</p> <p>Messages are passed from the three external programs, to the first business program. Additional messages representing increased business are passed to the second business program.</p>

Introduction to messaging and queuing

This part of the tour introduces the concepts of messages and queues, shows how they are used together, and includes a simple messaging scenario.

Message

Product tour text	Product tour animation
<p>Messages are used by communicating programs to exchange data.</p> <p>The sending program constructs a message containing:</p> <ul style="list-style-type: none">• the data to send• a message header (control information, for example message ID and return address) <p>When the message has been constructed by the program, it is then ready to send.</p>	<p>This part of the animation shows how WebSphere MQ messages are constructed by the programs that send them.</p> <p>The animation shows two programs connected by a message channel. One program needs to send a message to the other program.</p> <p>The sending program creates a message wrapper for holding the message header and data.</p> <p>The sending program places a header containing the message ID, expiration and return address into the message wrapper.</p> <p>The sending program places the data to be sent into the wrapper.</p> <p>The completed message is now ready for sending.</p>

Queue

Product tour text	Product tour animation
<p>A queue is where messages are stored until a program receives them.</p> <p>The sending program puts messages onto the appropriate queue.</p> <p>When the receiving program is ready, it gets its messages from the queue.</p>	<p>This part of the animation shows how WebSphere MQ uses queues to safely hold messages in transit.</p> <p>The animation shows a queue on the receiving program.</p> <p>The sending program sends a message to the receiving program.</p> <p>The message is transferred over the connection, and is placed on the queue.</p> <p>The sending program sends a second message to the receiving program.</p> <p>The second message is transferred over the connection, and is placed on the queue next to the first message.</p> <p>The receiving program retrieves the first message from the queue.</p> <p>The receiving program retrieves the second message from the queue.</p>

Messaging topologies

Product tour text	Product tour animation
<p>Messaging topologies can contain different combinations of networked machines and programs. A queue is used for holding messages received from programs on the same machine, or other machines.</p> <p>Messages on the queue can also be accessed by programs on the same machine, or other machines.</p>	<p>This part of the animation shows how WebSphere MQ messaging works on different messaging networks.</p> <p>The animation shows two machines (a sender and a receiver), with a program on each. The machines are connected, and a queue is created on the receiver.</p> <p>A second program is added to the receiver machine.</p> <p>A second machine with a program is now added to the sender side, and a connection to the receiver machine is established.</p> <p>A program on the receiver places a message onto its own queue.</p> <p>A program on a sender places a message on the receiver queue.</p> <p>Two more machines with a program on each are created, and connections established to the receiver machine.</p> <p>A program on the receiver machine retrieves a message from its own queue.</p> <p>A program on one of the new machines retrieves a message from the queue.</p>

A simple messaging scenario

Product tour text	Product tour animation
<p>The flexibility of WebSphere MQ allows it to provide messaging throughout the business.</p> <p>For example, shops can use messaging to send orders to headquarters.</p> <p>Headquarters can use messaging to send delivery requests to depots.</p> <p>Depots can use messaging to confirm delivery dates with headquarters (which then sends confirmations to shops).</p>	<p>This part of the animation shows how WebSphere MQ allows programs running in different parts of the business to communicate.</p> <p>The animation shows how messaging might be applied in a simple business scenario.</p> <p>Headquarters is represented by a single machine, with two programs on it.</p> <p>Shops are represented by two machines, with a program on each.</p> <p>Depots are represented by two machines, with a program on each.</p> <p>The shop and depot machines each have their own connection to headquarters.</p> <p>Messages (for example orders) are placed by the programs on the shop machines, onto the queue on the headquarters machine.</p> <p>The programs on the headquarters machine retrieve these messages from the queue.</p> <p>Queues are created on the depot machines.</p> <p>Messages (for example stock queries) are placed by the programs on the headquarters machines, onto the queues on the depot machines.</p> <p>The programs on the depot machines receive these messages from their respective queues.</p>

Communicating with the wider world

Product tour text	Product tour animation
<p>The flexibility of WebSphere MQ also allows messaging between programs that are internal and external to the business.</p> <p>For example, a business transaction may involve online orders, banks, and suppliers.</p> <p>Messaging provides a common method of communication for these distributed programs.</p>	<p>This part of the animation shows how WebSphere MQ allows messaging communications between a business, and its external service providers (for example banks, online ordering web sites and suppliers).</p> <p>Two bank machines with a program and a queue on each are added.</p> <p>Three supplier machines with a program and a queue on each are added.</p> <p>One online orders (website) machine with a program and a queue on it is added.</p> <p>Each of these machines has a connection to headquarters.</p> <p>Messages are now passed between the programs on this topology, to support the following business operations:</p> <ul style="list-style-type: none">• Receive order• Check credit• Check stock level• Confirm order• Order supplies

WebSphere MQ messaging fundamentals

This part of the tour introduces the concepts central to WebSphere MQ messaging environments. These include queue managers, MQ clients, messaging communications, and queue manager clustering.

Queue managers

Product tour text	Product tour animation
<p>Each computer that hosts queues requires a queue manager. Each queue manager has a unique name, and administers the queues that have been created on it (these are known as local queues).</p> <p>Each local queue also has a name which, along with the name of its queue manager, provides a unique address where messages can be sent.</p>	<p>This part of the animation shows how WebSphere MQ queue managers support a typical business scenario.</p> <p>Three machines have a program on each.</p> <p>Three more machines each have a program, a queue manager, and one or more queues. The queue managers and queues on these machines are:</p> <ul style="list-style-type: none">• A queue manager QM_A, and two queues Q1 and Q2• A queue manager QM_B, and one queue Q3• A queue manager QM_C, and one queue, Q4

Communicating with local queue managers

Product tour text	Product tour animation
<p>A program communicates with a local queue manager using a supported application programming interface (API) such as the WebSphere MQ Message Queue Interface (MQI), or the Java Message Service (JMS).</p> <p>Messages are constructed by the program and sent to its queue manager using API calls.</p> <p>If the target queue is local, the queue manager can put messages onto it directly.</p> <p>API calls are also used by programs to retrieve messages from their local queues.</p>	<p>This part of the animation shows how a WebSphere MQ API provides the interface between a program and the queue manager on the same machine.</p> <p>A program on a machine that has a queue manager places a message directly onto a queue on that machine, using a WebSphere MQ MQPUT call.</p> <p>Another program on the same machine retrieves the message from the queue, using a WebSphere MQ MQGET call.</p>

Communicating using WebSphere MQ clients

Product tour text	Product tour animation
<p>A program that does not have a queue manager on the same machine uses API calls via a local WebSphere MQ Client instead.</p> <p>The client communicates with the remote queue manager over a client connection. This allows the program to interact with the remote queue manager as though it were local.</p>	<p>This part of the animation shows how a business might use WebSphere MQ clients.</p> <p>Three machines, each with a program, an API, and a WebSphere MQ Client (but no queue manager or queues) are the client machines.</p> <p>A machine with two programs, an API, a queue manager, and two queues is the server machine.</p> <p>A client connection is created between each of the client machines and the server machine.</p> <p>A program on one of the client machines sends a message to the server machine. The message is transferred over the client connection to a queue on the server machine.</p> <p>One of the programs on the server machine retrieves the message from the queue.</p> <p>The same program on the server machine places a message on the same queue.</p> <p>The message is transferred over the client connection to the client machine.</p>

Sending messages to remote queues

Product tour text	Product tour animation
<p>For a program to send messages to a remote queue (a queue on a different queue manager), the following must be available:</p> <ul style="list-style-type: none">• A message channel (connects the local and remote queue managers)• A transmission queue (used to store messages if the message channel is unavailable)	<p>This part of the animation shows how a program puts WebSphere MQ messages onto a remote queue.</p> <p>A message channel is created between a queue manager on one machine (the sending machine), and a queue manager on another machine (the receiving machine).</p> <p>A transmission queue is created on the queue manager on the sending machine.</p> <p>A program on the sending machine wants to send a message to the receiving machine.</p> <p>The sending program places a message on the transmission queue.</p> <p>The message is sent over the channel to the local queue on the receiving machine.</p>

Making queue locations transparent

Product tour text	Product tour animation
<p>A program can send messages to a remote queue without needing to know its location, if a remote queue definition has been defined locally.</p> <p>The remote queue definition holds the the target queue and transmission queue names. The local queue manager uses this information to determine where to send the message.</p> <p>By using remote queue definitions, message channels and transmission queues, any program can send messages to any queue without needing to know where that queue is located.</p>	<p>This part of the animation shows how a WebSphere MQ queue manager uses remote queue definitions to obtain the location of queues before sending messages.</p> <p>A remote queue definition is created on the sender machine. The definition points to a local queue on the receiving machine.</p> <p>A program on the sending machine wants to send a message to a program on the receiving machine.</p> <p>The queue manager on the sender machine obtains the name of the transmission queue for the required destination, from its remote queue definition.</p> <p>The message is placed on the correct transmission queue, and then transferred over the message channel, to the local queue on the receiving machine.</p> <p>The program on the receiving machine retrieves the message from its local queue.</p> <p>The animation now shows how messages might be sent to other remote queues on the topology, without the sending queue managers having to know explicitly, where the remote queues are physically located.</p>

Simplifying queue administration

Product tour text	Product tour animation
<p>Administration of complex messaging networks can become demanding, particularly when queues are regularly being added or removed. This administrative burden can be reduced by using queue manager clusters.</p> <p>Queue managers in the same cluster can exchange messages without requiring remote queue definitions, transmission queues, or message channels.</p> <p>Information about all cluster queues on the cluster are held in a cluster full repository.</p> <p>When a queue manager in the cluster wants to send a message to a cluster queue, it obtains the queue location from the cluster full repository. A cluster channel can then be automatically established.</p> <p>Whenever a cluster queue is added to, or removed from a queue manager in the cluster, the cluster full repository is automatically updated.</p>	<p>This part of the animation shows how a WebSphere MQ queue manager cluster reduces the administration overhead on a messaging network.</p> <p>Three machines each hold a queue manager, multiple transmission queues, and multiple remote queue definitions. The queue managers on this network are connected by multiple message channels. The queue managers on this complex network are not currently clustered.</p> <p>The queue managers are now clustered, and all the transmission queues, remote queue definitions, and message channels that were previously required are removed, leaving only one or two local queues on each queue manager. The local queues are known as cluster queues, and the messaging network has already become much less complex.</p> <p>A cluster full repository containing information about the location of each queue on the cluster is created on one of the queue managers. The entries in the repository map each cluster queue to its owning queue manager, for example Q1 is on QM_A.</p> <p>A program within the cluster now wants to send a message to another program within the same cluster.</p> <p>The queue manager on the sending machine obtains the name of the queue manager where the target queue is located, from the full cluster repository.</p> <p>A cluster channel is automatically created between the sending queue manager and the receiving queue manager, and the message is sent over the cluster channel, to the cluster queue on receiving queue manager.</p> <p>The message is now retrieved from the local cluster queue by the receiving program.</p> <p>Another queue is added to one of the queue managers within the cluster.</p> <p>The full cluster repository is automatically updated with the queue name and the owning queue manager name.</p> <p>The new queue has now become a cluster queue and can easily be located by other queue managers in the cluster.</p>

Workload balancing between programs

Product tour text	Product tour animation
<p>If the volume of messages being sent to a cluster queue exceeds the processing capabilities of the receiving program(s), the workload can be balanced by defining cluster queues with the same name, on other queue managers in the same cluster.</p> <p>When messages are sent to a cluster queue that has been defined on multiple queue managers, WebSphere MQ automatically balances the workload by distributing the messages equally.</p>	<p>This part of the animation shows how WebSphere MQ allows an increasing amount of messaging traffic to be balanced between the cluster queues on a queue manager cluster.</p> <p>A queue named Q5 is defined on queue manager QM_B and queue manager QM_C, within the same queue manager cluster.</p> <p>The full cluster repository is automatically updated with the names of the cluster queues and the names of their queue managers.</p> <p>These cluster queues are now available to any queue manager within the cluster.</p> <p>Programs on machines outside the cluster now start sending multiple messages destined for queue Q5, to one of the machines within the cluster.</p> <p>As the first message arrives, the cluster queue manager on that machine obtains the locations of the two queues named Q5, from the full cluster repository.</p> <p>Subsequent messages arriving for queue Q5 can now be sent alternately to each Q5 queue, thus balancing the workload.</p>

Interacting with WebSphere MQ

This part of the tour introduces the WebSphere MQ administration interfaces, the messaging interfaces available for programs to use, and a simple file transfer tool.

Creating and administering WebSphere MQ objects

Product tour text	Product tour animation
<p>WebSphere MQ objects such as queues and channels can be administered by using:</p> <p>WebSphere MQ Explorer</p> <p>WebSphere MQ Script Commands (MQSC)</p> <p>The WebSphere MQ Explorer GUI runs on Windows platforms, and provides an easy-to-use administration interface.</p> <p>WebSphere MQ Explorer can administer remote queue managers running on Windows, Unix, Linux, OS/400, and other platforms.</p> <p>MQSC is a command line based interface which can be used to issue commands interactively or from scripts.</p> <p>For example, the MQSC command DEFINE (used to define objects such as queues), DISPLAY (used to display object attributes such as the number of messages on a queue) and CLEAR (used to remove messages from a queue).</p>	<p>This part of the animation shows how WebSphere MQ objects such as queues and channels are created and managed.</p> <p>The animation uses a single machine with two programs, two APIs, a queue manager QM_A, and a single queue Q1.</p> <p>The MQ Explorer dialog opens.</p> <p>The animation shows how these objects are represented in the MQ Explorer (the queue manager is highlighted in the navigation tree in the left pane, and the queue is highlighted in the pane on the right).</p> <p>Another machine with a single program and an API appears in the topology. This machine has a queue manager QM_C and two queues, Q4 and Q5. The animation shows how these objects are represented in the MQ Explorer.</p> <p>A command prompt window opens, the command runmqsc QM_A is issued, and the status message Starting WebSphere MQ script commands appears.</p> <p>A sequence of operations in the command prompt window now shows how MQSC commands can be used to define a queue, check the queue current depth (number of messages are on the queue), put a message onto the queue, check the queue current depth, clear the queue, check the queue current depth:</p> <p>The command DEFINE QLOCAL(Q2) CLUSTER(QMCLUS1) (create new queue) appears in the command prompt window. The new queue Q2 appears on the cluster. The status message AMQ8006 WebSphere MQ queue created (queue created) appears in the command prompt window.</p> <p>The command DISPLAY QLOCAL(Q2) CURDEPTH (display current queue depth) appears in the command prompt window. The status message AMQ8409 Display queue details appears in the command prompt window, followed by the message QUEUE(Q2) CURDEPTH(0) (zero messages on queue).</p> <p>A message is now put onto queue Q1.</p> <p>The command DISPLAY QLOCAL(Q2) CURDEPTH (display current queue depth) appears in the command prompt window. The status message AMQ8409 Display queue details appears in the command prompt window, followed by the message QUEUE(Q2) CURDEPTH(1) (1 message on queue).</p> <p>The command CLEAR QLOCAL(Q2) (clear queue) command, followed by the status message AMQ8022 WebSphere MQ queue cleared appear in the command prompt window.</p> <p>The command DISPLAY QLOCAL(Q2) CURDEPTH (display current queue depth) appears in the command prompt window. The status message AMQ8409 Display queue details appears in the command prompt window, followed by the message QUEUE(Q2) CURDEPTH(0) (zero messages on queue).</p>

WebSphere MQ application development

Product tour text	Product tour animation
<p>Programs use WebSphere MQ application programming interfaces (APIs) to communicate with queue managers.</p> <ul style="list-style-type: none"> • Procedural languages such as C use the Message Queue Interface (MQI). • Object oriented languages such as Java use WebSphere MQ classes. <p>MQI commands such as MQCONN (to connect with queue managers), MQOPEN (to open queues for messaging), and MQPUT (to send messages), are provided by the MQI. MQI commands are standard across all supported procedural languages, including C, COBOL and Visual Basic.</p> <p>Java applications connect to a queue manager by defining an object of class MQQueueManager, open a queue for messaging by calling the accessQueue() method, and receive messages by calling the get() method.</p> <p>C++ applications use similar objects and methods.</p>	<p>This part of the animation shows how applications interact with WebSphere MQ.</p> <p>The animation uses a single machine with two programs, two APIs, a queue manager QM_A, and two queues Q1 and Q2.</p> <p>A programming editor window containing a sample of C code opens, and the sample code in the editor window scrolls to reveal examples of how the MQI commands MQCONN (connect to queue manager), MQOPEN (open queue), and MQPUT (put message onto queue) are used in C application messaging.</p> <p>The programming editor window now shows a sample of Visual Basic code which scrolls to reveal how the MQI MQPUT command is used in Visual Basic application messaging.</p> <p>The programming editor window now shows a sample of COBOL code which scrolls to reveal how the MQI MQPUT command is used in COBOL application messaging.</p> <p>The programming editor window now shows a sample of Java code which scrolls to reveal how the MQI MQPUT command is used in Java application messaging.</p> <p>The programming editor window now shows a sample of C++ code which scrolls to reveal how the MQI MQPUT command is used in C++ application messaging.</p>

WebSphere MQ and the Java Message Service

Product tour text	Product tour animation
<p>Programs that use the Java Message Service (JMS) API do not directly specify queues or queue managers. Instead, they use generic objects such as Queue and QueueConnectionFactory.</p> <p>Mappings to WebSphere MQ queues and queue managers are defined in a Java Naming Directory Interface (JNDI) namespace, using the command line based JMS Admin tool.</p> <p>JMS programs import the mappings from the JNDI at runtime.</p> <p>Any update to the JNDI mappings will be imported by every JMS program that uses them. Tasks such as changing queues or queue managers are therefore greatly simplified.</p>	<p>This part of the animation shows how WebSphere MQ interacts with the Java Messaging Service.</p> <p>The animation uses a single machine with two programs, two APIs, a queue manager QM_A, and two queues Q1 and Q2.</p> <p>A program editor window opens and a Java code example shows how the Queue and QueueConnectionFactory objects are invoked.</p> <p>A command prompt window opens, and the JMSAdmin command starts the JMS Admin tool.</p> <p>A JMS admin DEFINE command is issued to define an entry for the queue connection factory in the JNDI (Java Naming Directory Interface).</p> <p>A JMS admin DEFINE command is issued to define an entry for the queue in the JNDI.</p> <p>The Java code in the program editor window runs, the JMS object definitions are imported from the JNDI and a message is put onto the queue.</p>

Simple file transfer using WebSphere MQ

Product tour text	Product tour animation
<p>The Express File Transfer has an easy-to-use GUI that users such as office managers, personal assistants and sales staff can use to send and receive files such as spreadsheets and word processed documents, with all the benefits that WebSphere MQ provides.</p> <p>These users do not need to be aware that the underlying technology is WebSphere MQ messaging and queuing.</p>	<p>This part of the animation shows how the WebSphere MQ Express File Transfer is used for sending and receiving files in the form of MQ messages.</p> <p>The animation uses a single machine with two programs, two APIs, a queue manager QM_A, and two queues Q1 and Q2.</p> <p>The Express File Transfer GUI opens with the Send dialog displayed by default.</p> <p>The animation now shows the sequence of user actions for sending a file to a selected destination (that is, for putting a message onto a specified queue). The user actions are: browse, select file, select destination, click Send.</p> <p>The animation now shows the sequence of user actions for receiving a file (that is, for getting a message from a specified queue). The user actions are: select file, click Receive.</p>

Express File Transfer

Express File Transfer allows you to send and receive ordinary files in the form of MQ messages. You can use Express File Transfer to send and receive any type of file in any format, for example: ASCII Linux format (with line feed characters), ASCII file Windows format (with carriage return/line feed characters), binary (for example, image files, wordprocessor files, spreadsheet files, or zip files), also reports, letters, memos and charts.

Express File Transfer provides two methods of interaction to cater for the requirements of different types of user:

- A *graphical user interface*. This interface allows non-experienced users to send files, receive files, and create a list of sent/received files in an intuitive way. Users of the GUI need no knowledge of how the underlying WebSphere MQ technology works.

For instructions on how to set up and use the Express File Transfer graphical user interface, see “Using the graphical interface” on page 21.

- A *command line interface*. This provides a way for experienced users to send and receive files by issuing commands from the command-line. Additional options are provided for controlling file transfer characteristics such as message size and persistence. Users of the command-line interface need to have an understanding of how WebSphere MQ works.

For instructions on how to use the Express File Transfer command line interface, see “Using the command line interface” on page 26.

Advantages of using Express File Transfer

- *Files of any type can be transferred*. Because Express File Transfer does not distinguish between files of different types, you can send and receive files in any format (for example, spreadsheets, memos, letters). You can even send and receive image and sound files.
- *File transfer is technology independent*. Files can be transferred between dissimilar operating platforms (for example Windows, Unix), over different network communications protocols (for example TCP/IP, LU6.2).
- *Transferred files cannot be accidentally duplicated*. Files are sent once-and-once-only to a specified destination.
- *Files are transferred securely*. High-level data security and integrity is provided if SSL (secure sockets layer) encrypted message channels are used.
- *The sender and receiver run independently*. The sender and receiver do not both have to be running at the same time. If the receiver is currently unavailable or busy, the file is held on a queue. When the receiver becomes available, the file is then automatically transferred. The persistent option assures maximum reliability (non-persistent messages are automatically deleted from the queue on the receiving machine, when the queue manager restarts).
- *The system is scalable*. An administrator can add new sources and destinations to the Express File Transfer user interface, so that they become accessible to users.
- *A GUI for the Client*. Express File Transfer provides a graphical user interface for the Client as well as the Server.

Preparing WebSphere MQ for use with Express File Transfer

Express File Transfer converts files into messages and sends them to specified destinations, from where they are reassembled into files. These destinations are WebSphere MQ queues, and Express File Transfer relies on an appropriately configured messaging environment to route its messages to these queues.

Express File Transfer can be used on computers that have a local queue manager set up on them - this requires a WebSphere MQ Server installation. Express File Transfer can also be used on computers which do not have a local queue manager, but which have the WebSphere MQ Client installed and configured to connect to a WebSphere MQ Server.

Preparing for server to server file transfer

For Express File Transfer to send files from one queue manager to a queue on another queue manager, the following must be set up:

- On the sending queue manager:
 - A remote queue definition identifying the destination queue and queue manager names.

- A transmission queue to hold messages if the destination queue manager is unavailable.
- A sender channel definition to connect to the destination queue manager.
- On the destination queue manager:
 - A receiver channel definition to connect to the sending queue manager.
 - A local queue that is the destination.

For a working example of the above configuration, see “Sending a message to a remote queue” on page 40.

The above set up provides the minimum configuration required for server to server messaging. For further information on the WebSphere MQ objects mentioned above, and other configuration options, refer to the product manuals.

Preparing for client to server file transfer:

For Express File Transfer to send files from a computer with MQ Client installed to a queue on another computer, the following must be set up:

- On the sending computer:
 - The MQ Client must be set up to connect to the destination queue manager. This can be achieved by:
 - Configuring the MQSERVER environment variable with the connection details of the destination queue manager, or
 - Adding the connection details of the destination queue manager to the client channel definition table
- On the destination queue manager:
 - A server-connection channel definition to allow client connection.
 - A local queue that is the destination.

For a working example of the above configuration using the MQSERVER environment variable, see the “Sending a message on a client-server configuration” on page 46.

The above set up provides the minimum configuration required for client to server messaging. For further information on the WebSphere MQ objects mentioned above, and other configuration options, refer to the product manuals.

Using the graphical interface

The Express File Transfer graphical interface is aimed at non-technical users, who want a simple method of transferring files, with all the benefits that WebSphere MQ provides. Users of the graphical interface do not need knowledge of how the underlying WebSphere MQ technology works.

The Express File Transfer graphical interface is available on computers that have had WebSphere MQ Server or Client installed.

Setting up the graphical interface for first use

An IT system administrator should already have created the required WebSphere MQ objects (queue managers, queues, and messaging channels). The administrator should also have installed Express File Transfer onto the user machine, and must be available to do the initial setup described here.

For further information see ‘Preparing WebSphere MQ for use with Express File Transfer’ in the WebSphere MQ Express Center.

The objective of this task is to set up the queues that were previously defined in WebSphere MQ, as destinations and sources for the Express File Transfer GUI. These can then be accessed for sending and receiving files. Setup consists of making the destinations available and ensuring that only their meaningful names appear in the user interface. Note that the first time you start Express File Transfer, a message prompts you to complete the initial setup task.

1. Start the **Express File Transfer** graphical user interface:
 - a. On Windows computers, use the Start Menu (**Start > Programs > IBM WebSphere MQ Express > Express File Transfer**) or open a command prompt window and type:

```
mqftapp
```

- b. On Linux computers, open a command prompt and enter the command:

```
mqftapp
```

2. Click the **OK** button on the Welcome Window.
3. If the computer Express File Transfer is running on has a WebSphere MQ Server installed on it and a local queue manager is to be used, in the **Queue Manager Name** field, enter the queue manager name, then click the **Local** radio button.
4. If the computer Express File Transfer is running on has the WebSphere MQ Client installed on it and a remote queue manager is to be used, in the **Queue Manager Name** field, enter the queue manager name, then click the **Remote** radio button.
5. Click the **OK** button. Express File Transfer connects to the queue manager and displays a list of all available destinations (the **Destinations** tab is selected by default).
6. Select the check box adjacent to each destination whose WebSphere MQ queue name (the alias name in the **Type** column), you want to use.
7. Click the **Sources** tab.
8. Select the check box adjacent to each source whose WebSphere MQ queue name (the alias name in the **Type** column), you want to use.
9. Click the **OK** button.

The initial setup task is now complete, and the main Express File Transfer window displays the selected destinations and sources with only their meaningful (alias) queue names. You can now close Express File Transfer by clicking **File > Exit**.

Sending a file

This task shows you how to send a file to another destination using the Express File Transfer GUI. You can also send files from the command line by using the `mqftsnd` command.

1. Start the **Express File Transfer** graphical user interface.
2. Click the **Send** tab.
3. Click the **Browse** button and in the browser dialog, select the file to transfer and click **OK**.
4. In the **Comments** field, add an optional comment, for example: **Hi Frank, here's the spreadsheet.**
5. In the **Destination** pane, click the required destination, for example: **HQ Sales**.
6. Click **Send**. If the time to complete the receive action is more a couple of seconds, a progress bar appears.
7. Look for confirmation that the file was sent in the **Session Log** list.

Express File Transfer sends the specified file to the selected location.

Receiving a file

This task shows how to receive a file from another destination using the Express File Transfer GUI. You can also receive files from the command line by using the `mqftrcv` command.

1. Start the **Express File Transfer** graphical user interface.
2. Click the **Receive** tab.
3. Click the **Files in** drop-down list to display the source where the file is held, for example: Hants regional depot.
4. Select the source to display the files stored there.
5. Select the file to receive, for example: Stocks 14 Aug 03.doc.
6. Click **Receive**. If the time to complete the receive action is more a couple of seconds, a progress bar appears. The **Receive** option removes the file from the source. If you want to receive the file, but keep it on the sending machine, you can use the **File > Extract** option instead of **Receive**.

You can check that the file was received by looking in the **Session log** list.

Listing all sent and received files

This task shows how to create a list of all files sent and received using the Express File Transfer GUI.

1. Start the **Express File Transfer** graphical user interface.
2. Click **History log**. The list shows all files sent, received, extracted or deleted since Express File Transfer was first used, or since the log was last cleared.
3. Click **Save as**.
4. Navigate to a folder on the hard drive on your machine where you want to save the list.
5. In the **File name** field, type a meaningful file name, for example: File log 9 Apr 03.
6. In the **Save as type** field, type .txt (this is currently the only supported filetype for saving files).
7. Click **Save**.
8. Click **Close**.

The text file containing the list of files sent and received is created and saved as a text file.

Further graphical interface options

All options that are available from the Express File Transfer graphical interface are listed here, grouped by menu or panel title.

Send menu

Feature	Characteristic	Purpose
File	Menu command	Select this menu choice to work with the File menu.
View	Menu command	Select this menu choice to work with the View menu.
Help	Menu command	Select this menu choice to work with the Help menu.
Send	Tab	Select this tab to work with the Send dialog, from where you can send files.
Files	Text field	This field contains the name of the file to send. Use the Browse button to locate the file.
Browse...	Button	Click this button to open the file browser where you can navigate to a file to send.
Comments	Text entry field	Enter any additional comments you want to send with the file, for example: Hi Frank, here's the weekly sales report.
Destination	Table element	This pane contains a list representing the currently available destinations where files can be sent. Click the destination where you want to send the file.
Files sent	Table element	This dynamically updated field shows the number of files sent to a given destination.

Feature	Characteristic	Purpose
Send	Button	Click this button to send the currently selected files to the specified destinations
Session Log	Check box	Click this check box to display all the files that have been sent and received since Express File Transfer was started, or since the log was cleared.
Name	Record field	This field shows the name of the file.
Time	Record field	This field shows the time of the action.
To	Record field	This field shows the destination of the file.
Action	Record field	This field shows whether the file was sent, received, deleted or extracted.
From	Record field	This field shows where the file was sent from.
History Log...	Button	Click this button to view the log containing a list of all files sent and received. You can also save this list as a text file.

Receive menu

Feature	Characteristic	Purpose
File	Menu command	Select this menu choice to work with the File menu.
View	Menu command	Select this menu choice to work with the View menu.
Help	Menu command	Select this menu choice to work with the Help menu.
Receive	Tab	Select this tab to work with the Receive dialog, from where you can receive files.
Files in	Drop-down list	This list contains the sources from where files can be received, for example: Fleet Contracts. Click source to display the list of files that can be received from that source. The list of files appears in the pane immediately below the drop-down list of sources.
Name	Record field	This field contains the filename and filetype of a file that can be received from the currently selected source, for example: ABC Leasing.doc.
Comments	Record field	This field contains a comment associated with a file that can be received, for example: Bill, sent you these figures for approval.
Size	Record field	This field shows the file size.
Receive	Button	Click this button to receive the currently selected files.
Session Log	Check box	Click this check box to display all the files that have been sent and received since Express File Transfer was started, or since the log was cleared.
Name	Record field	This field shows the name of the file.
Time	Record field	This field shows the time of the action.
Action	Record field	This field shows whether the file was sent, received, deleted or extracted.
To	Record field	This field shows the where the file has been received.
From	Record field	This field shows the source of the file.
History Log...	Button	Click this button to view the log containing a list of all files sent and received. From the log, you can save the list as a text file.

Identify queue manager panel

Feature	Characteristic	Purpose
Name	Text entry field	Enter the name of a local queue manager, or the name of a remote 'target' queue manager, for example: QM_showroom5. If you do not enter a value, Express File Transfer automatically uses the default queue manager name.
Local	Radio button	Select this button if the value you specified in the Name field is the name of a local queue manager.
Remote	Radio button	Select this button if the value you specified in the Name field is the name of a remote queue manager.
OK	Button	Click this button to confirm the values and settings in this dialog and to exit the dialog.
Cancel	Button	Click this button to cancel any changes in this dialog.

Setup panel

Feature	Characteristic	Purpose
Queue manager	Field	This field displays the name of the queue manager you selected in the Identify queue manager dialog. This is the queue manager to which you are going to make selected sources and destinations available for the user.
Change...	Button	Click this button to modify the name of the current queue manager. When you click this button, the Identify queue manager dialog opens, and you can make the changes.
Destinations	Tab	Select this tab to work with destinations.
Sources	Tab	Select this tab to work with sources.
Name	Record field	This field shows the meaningful queue name of an available queue, for example: HQ.Sales.
Type	Record field	This field shows the queue's type, for example: alias or remote.
Description	Record field	This field contains a reminder of the purpose of the queue, for example: Send your weekly sales report.
Select all	Button	Click this button to select all queues. When you do this, the box to the left of each queue is checked.
Deselect all	Button	Click this button to deselect all queues. When you do this, the box to the left of each queue is blank.
OK	Button	Click this button to confirm the values and settings in this dialog, and to exit the dialog.
Cancel	Button	Click this button to cancel the changes in this dialog.

Features menu

Friendly formatting	Button	Click this button to format source and destination names.
Msg size	Text entry field	This is the size of the message, files larger than this will be sent in more than one message.
Persistent msgs	Drop down list	Persistent messages should be used for file integrity. Select Yes for persistent messages, No for non-persistent messages or as queue definition to use the message persistency as defined on the queue.

Using the command line interface

The Express File Transfer command line interface is aimed at experienced users, who are expert IT users working within a business, and have an understanding of WebSphere MQ. These users want to send and receive files from the command-line, using additional control parameters, or write applications, scripts, or macros which call the Express File Transfer.

Important: Before Express File Transfer can be used, WebSphere MQ must be configured to provide a suitable messaging environment. See “Preparing WebSphere MQ for use with Express File Transfer” on page 20 for further details.

The Express File Transfer command line interface consists of the following main components:

- A *sender*. This is a program that puts a file stored in the local file system onto a queue, as one or more MQ messages.
For details of how to invoke the sender program on WebSphere MQ Servers, see “Send file command (mqftsnd)”.
For details of how to invoke the sender program on WebSphere MQ Clients, see “Send file command (mqftsndc)” on page 30.
- A *receiver*. This is a program that receives files from a queue and stores them in a local file system.
For details of how to invoke the receiver program on WebSphere MQ Servers, see “Receive file command (mqftrcv)” on page 28.
For details of how to invoke the receiver program on WebSphere MQ Clients, see “Receive file command (mqftrvc)” on page 32.

Send file command (mqftsnd)

Purpose

Use this command to put a file onto a specified queue when working from a WebSphere MQ Server. When you use this command, you must specify the QueueName and FileName. Additional options are also available.

Syntax

```
>>-mqftsnd -q QueueName --+-----+-----+-----+----->
                        '- -m QMgrName --'   '- -t TargetQMGrName -'

                                     .- -p yes ---.
>-----+-----+-----+-----+-----+----->
        '- -v -'   '- -l MsgLength -+   +- -p no ----+
                                     '- -p queue -'

>-----+-----+-----+-----+-----+-----><
        '- -s Comments --+
```

Options

-q QueueName

The name of the local queue for putting messages. There is no default. It is recommended that a queue is dedicated for file transfer. This value is mandatory.

-m QMgrName

The name of a local queue manager. If a name is not specified, the application uses the default queue manager.

-t TargetQMGrName

The name of the target queue manager. The target queue manager defaults to the name of the queue manager you are connected to. The Express File Transfer does not attempt to create a channel

between the queue manager and the target queue manager. This field is only used if sending to a different queue manager to the one you are sending from.

-v Verbose option that shows the message correlation ID (CorrelId). The default is not to show this.

-l MessageSize

Restricts the size of message data. Large files are sent as multiple messages. The minimum size is restricted by the need to send some of the data and a small header (approximately 250 bytes) in each message. An error is reported if this is not possible. If the queue manager configuration does not allow messages of MessageSize to be put to the target queue, the request may fail. The default value is 100000 bytes. Setting a very small message size disproportionately increases the number of messages required to send the file, because each message includes the overhead of a small (approximately 250 bytes) message header.

-p Persistence

The persistence for new messages.

- y (or yes): messages are persistent
- n (or no): messages are not persistent.
- q (or queue): message persistence as defined by the queue.

The default is for persistent messages.

-s Comments

Specifies a string of arbitrary user data that is used to distinguish between files. This can be useful for differentiating between two files with the same name on the specified queue, for example: "XYZ Internal sales-figures for 2003".

-f FileName

The name of a file to send. The source file is not deleted. The filename can include a path to the file. A filename with embedded spaces must be quoted, for example: "Long File Names.txt". There is no wildcard expansion, no default, and only one file is sent. This value is mandatory.

Usage

```
mqftsnd -q QueueName [-m QMgrName] [-t TargetQMGrName] [-v]
        [-l MsgLength] [-p yes | no | queue] [-s Comments] -f FileName
```

Examples

This example sends a file to the default queue manager to MY.QUEUE on your default local queue manager:

```
mqftsnd -q MY.QUEUE -f spreadsheet.xls
```

This example sends a file and displays the correlID to standard output for audit purposes:

```
mqftsnd -q MY.QUEUE -v -f spreadsheet.xls
```

Usage notes

- Before using this command, ensure that the required queue managers, queues and channels are available.
- Use this command to send binary files and fixed or variable record length files.
- Options must have a leading dash or a forward slash.
- Options for Windows are not case sensitive.
- Options for UNIX are case sensitive.
- Options cannot be repeated.
- Options and arguments must be separated by one or more spaces.
- If running a WebSphere MQ Express v5.3 Client, use the "Send file command (mqftsndc)" on page 30 to send files.

Purpose

Syntax diagram

Options

The name of the local queue for getting messages. There is no default queue name. The receiving program tries to dead letter queue 'other' messages placed on the receiver queue by other applications. The use of a dedicated queue is recommended. This value is mandatory.

The name of a local queue manager. If a name is not specified, the application uses the default queue manager.

- v Verbose option that shows the message Correlation ID and Message ID for MQ messages that do not consist of files or part files. For messages that consist of files or part files, only the Correlation ID is shown. The default is not to show these attributes. The Message ID is not shown for files because they are uniquely identified by their Correlation ID.
- a Lists all files by status: complete/incomplete/other, and then the queue order. This is the default action and does not have to be specified.
- l Lists complete files with status in queue order.
- i Lists incomplete files with status in queue order. An incomplete file is a file that has not yet been fully received by the queue manager. Rerunning the command with the -l option at a later time should show the file as complete, because the Express File Transfer assures delivery.
- o Lists other messages with status in queue order. In this situation, the receiver has attempted to remove all 'other' messages to the dead letter queue. If this fails, they can be manually deleted. An 'other' message is a bogus message on the specified queue (a message that is not part of any file that was previously sent by the Express File Transfer).
- d Deletes a specified file. There is no 'default' file. If multiple files match the selection, no files are deleted and the application returns a non zero return code.
- g Gets a complete file. The messages associated with the file are removed from the queue. By default use of this option does not replace an existing file. If the file exists, either rename the file when it is received (use the -r FileName option), or modify the behaviour (use the -y option).
- e Extracts (gets) a complete or incomplete file without deleting any messages from the queue. The messages associated with the file are retained on the queue, so that the -g (get) option can be used to retrieve the complete file at a later date. The file is reassembled using data from the beginning of the

file and is truncated at the lowest offset where data is not yet available. Data at a higher offset in the file, is ignored if any data at a lower offset is currently unavailable.

The command fails if the data at offset 0 is unavailable and the file length is non-zero. The return code is non-zero indicating that the command failed. The return code is non-zero if the file is incomplete, in order to distinguish it from a complete file that has been extracted.

By default use of this option does not replace an existing file. If the file exists, either rename the file when it is received (use the `-r FileName` option), or modify the behavior (use the `-y` option).

- y** Replaces an existing file without prompting. This option modifies the default behaviour on the `-g` (get) option or the `-e` (extract) option so that an existing file can be replaced. There is no prompting when the file is replaced.
- c** Selects files by their 24-byte correlation ID (CorrelId). The data is represented by a 48-character hexadecimal string, for example: `-c 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF`. The default is none. An 'all zero' CorrelId is assumed to be an error because it cannot be used to correlate files. This option can also be combined with the `MsgId`, `UserData` and `FileName`, to form a logical AND when making a file selection.
- u** Selects an 'other' message by its 24-byte message ID (MsgId). The data is represented by a 48-character hexadecimal string, for example: `-u 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF`. The default is none. This option can also be combined with the `CorrelId`, `UserData` and `FileName`, to form a logical AND when making a file selection, however the unique `MsgId` is normally sufficient when used on its own for selecting a single 'other' message.
- s** Specifies a string of arbitrary user data that is used to distinguish between files. This can be useful for differentiating between two files with the same name on the specified queue, for example:

```
"XYZ 'Internal' sales-figures for 2003"  
XYZ  
" 'Internal' "  
"sale"  
"sales-figures"  
2003  
03
```

This option selects by locating an occurrence of the specified `UserData` with part or all of the sender's `UserData`. The match is case-sensitive. Wild cards are not supported. The default is any. This option can also be combined with the `CorrelId`, `MsgId`, and `FileName`, to form a logical AND when making a file selection.
- r** Renames or moves a file to a new location. The directory must already have been created. The default filename used for the renamed or moved file is the same as the filename used in the current directory.
- f** Selects a file to receive by its filename (`FileName`). The filename can include a path to the file. A filename with embedded spaces must be quoted, for example: `"Long File Names.txt"`. There is no wildcard expansion. The default is any file on the queue. This option can also be combined with the `CorrelId`, `MsgId` and `UserData`, to form a logical AND when making a file selection. This option only receives one file.

Usage

```
mqftrcv -q QueueName [-m QMgrName] [-c CorrelId] [-u MsgId] [-s Comments]  
          [-v] [-a | -l | -i | -o | [[-g | -e] [-y]] | -d] [-r FileName]  
          [-f FileName]
```

Examples

This example lists all files from MY.QUEUE:

```
mqftrcv -q MY.QUEUE -a
```

This example gets the first complete file:

```
mqftrcv -q MY.QUEUE -g
```

This example gets the complete file spreadsheet.xls:

```
mqftrcv -q MY.QUEUE -g -f spreadsheet.xls
```

This example gets the complete file spreadsheet.xls also marked 'urgent':

```
mqftrcv -q MY.QUEUE -g -f spreadsheet.xls -s "URGENT"
```

Usage notes

- Before using this command, ensure that the required WebSphere MQ queue managers, queues and channels are available.
- Use this command to receive binary files and fixed or variable record length files.
- Options must have a leading dash or a forward slash.
- Options for Windows are not case sensitive.
- Options for UNIX are case sensitive.
- Options cannot be repeated.
- Options and arguments must be separated by one or more spaces.
- If running a WebSphere MQ Express v5.3 client, use the "Receive file command (mqftrcv)" on page 32 to receive files.

Send file command (mqftsndc)

Purpose

Use this command to put a file onto a specified queue when sending from a WebSphere MQ Client. Before you can use this command an administrator must set the MQ Server Environment Variables or the Client Channel definition. When you use this command, you must specify the QueueName and FileName. Additional options are also available.

Syntax

```
>>-mqftsndc -q QueueName ---+-----+-----+-----+----->
                        '- -m QMgrName --'   '- -t TargetQMGrName -'

                        .- -p yes ---.
>-----+-----+-----+-----+-----+-----+----->
                        '- -v -'   '- -l MsgLength -+   +- -p no ----+
                        '- -p queue -'

>-----+-----+-----+----- -f FileName -----><
                        '- -s Comments --+
```

Options

-q QueueName

The name of the local queue for putting messages. There is no default. It is recommended that a queue is dedicated for file transfer. This value is mandatory.

-m QMgrName

The name of a local queue manager. If a name is not specified, the application uses the default queue manager.

-t TargetQMGrName

The name of the target queue manager. The target queue manager defaults to the name of the queue manager you are connected to. The Express File Transfer does not attempt to create a channel between the queue manager and the target queue manager. This field is only used if sending to a different queue manager to the one you are sending from.

-v Verbose option that shows the message correlation ID (CorrelId). The default is not to show this.

-l MessageSize

Restricts the size of message data. Large files are sent as multiple messages. The minimum size is restricted by the need to send some of the data and a small header (approximately 250 bytes) in each message. An error is reported if this is not possible. If the queue manager configuration does not allow messages of MessageSize to be put to the target queue, the request may fail. The default value is 100000 bytes. Setting a very small message size disproportionately increases the number of messages required to send the file, because each message includes the overhead of a small (approximately 250 bytes) message header.

-p Persistence

The persistence for new messages.

- y (or yes): messages are persistent
- n (or no): messages are not persistent.
- q (or queue): message persistence as defined by the queue.

The default is for persistent messages.

-s Comments

Specifies a string of arbitrary user data that is used to distinguish between files. This can be useful for differentiating between two files with the same name on the specified queue, for example: "XYZ Internal sales-figures for 2003".

-f FileName

The name of a file to send. The source file is not deleted. The filename can include a path to the file. A filename with embedded spaces must be quoted, for example: "Long File Names.txt". There is no wildcard expansion, no default, and only one file is sent. This value is mandatory.

Usage

```
mqftsndc -q QueueName [-m QMgrName] [-t TargetQMgrName] [-v]
          [-l MsgLength] [-p yes | no | queue] [-s Comments] -f FileName
```

Examples

This example sends a file to the default queue manager to MY.QUEUE on your default local queue manager:

```
mqftsndc -q MY.QUEUE -f spreadsheet.xls
```

This example sends a file and displays the correlID to standard output for audit purposes:

```
mqftsndc -q MY.QUEUE -v -f spreadsheet.xls
```

Usage notes

- Before using this command, ensure that the required queue managers, queues and channels are available. In addition, ensure the MQSERVER environment variable or channel definition tables have been set.
- Use this command to send binary files and fixed or variable record length files.
- Options must have a leading dash or a forward slash.
- Options for Windows are not case sensitive.
- Options for UNIX are case sensitive.
- Options cannot be repeated.
- Options and arguments must be separated by one or more spaces.
- If running a WebSphere MQ Express v5.3 Server, use the "Send file command (mqftsnd)" on page 26 to send files.

Receive file command (mqftrcvc)

Purpose

Use this command to get a file from a specified queue when working from a WebSphere MQ Client. Before you can use this command an administrator must set the MQ Server Environment Variables or the Client Channel definition. When using this command, you must specify a QueueName. Additional options are also available.

Syntax diagram

```

                                .- -a -.
>>-mqftrcvc -q QueueName --+-----+-----+-----+-----+-----+----->
                             '- -m QMgrName -' '- -v -' +- -l +- !
                             +- -i +- !
                             +- -o +- !
                             +- -d -' !
                             ! !
                             +- -g +-+-----+
                             '- -e -' '- -y -'

>-----+-----+-----+-----+-----+----->
          '- -c CorrelId -' '- -u MsgId ----' '- -s Comments ---'

>-----+-----+-----+-----+-----+-----<
          '- -r FileName -+ '- -f FileName ---'
```

Options

-q QueueName

The name of the local queue for getting messages. There is no default queue name. The receiving program tries to dead letter queue 'other' messages placed on the receiver queue by other applications. The use of a dedicated queue is recommended. This value is mandatory.

-m QMgrName

The name of a local queue manager. If a name is not specified, the application uses the default queue manager.

-v Verbose option that shows the message Correlation ID and Message ID for MQ messages that do not consist of files or part files. For messages that consist of files or part files, only the Correlation ID is shown. The default is not to show these attributes. The Message ID is not shown for files because they are uniquely identified by their Correlation ID.

-a Lists all files by status: complete/incomplete/other, and then the queue order. This is the default action and does not have to be specified.

-l Lists complete files with status in queue order.

-i Lists incomplete files with status in queue order. An incomplete file is a file that has not yet been fully received by the queue manager. Rerunning the command with the **-l** option at a later time should show the file as complete, because the Express File Transfer assures delivery.

-o Lists other messages with status in queue order. In this situation, the receiver has attempted to remove all 'other' messages to the dead letter queue. If this fails, they can be manually deleted. An 'other' message is a bogus message on the specified queue (a message that is not part of any file that was previously sent by the Express File Transfer).

-d Deletes a specified file. There is no 'default' file. If multiple files match the selection, no files are deleted and the application returns a non zero return code.

-g Gets a complete file. The messages associated with the file are removed from the queue. By default use of this option does not replace an existing file. If the file exists, either rename the file when it is received (use the **-r FileName** option), or modify the behaviour (use the **-y** option).

-e Extracts (gets) a complete or incomplete file without deleting any messages from the queue. The

messages associated with the file are retained on the queue, so that the -g (get) option can be used to retrieve the complete file at a later date. The file is reassembled using data from the beginning of the file and is truncated at the lowest offset where data is not yet available. Data at a higher offset in the file, is ignored if any data at a lower offset is currently unavailable.

The command fails if the data at offset 0 is unavailable and the file length is non-zero. The return code is non-zero indicating that the command failed. The return code is non-zero if the file is incomplete, in order to distinguish it from a complete file that has been extracted.

By default use of this option does not replace an existing file. If the file exists, either rename the file when it is received (use the -r FileName option), or modify the behavior (use the -y option).

- y Replaces an existing file without prompting. This option modifies the default behaviour on the -g (get) option or the -e (extract) option so that an existing file can be replaced. There is no prompting when the file is replaced.
- c Selects files by their 24-byte correlation ID (CorrelId). The data is represented by a 48-character hexadecimal string, for example: -c 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF. The default is none. An 'all zero' CorrelId is assumed to be an error because it cannot be used to correlate files. This option can also be combined with the MsgId , UserData and FileName, to form a logical AND when making a file selection.
- u Selects an 'other' message by its 24-byte message ID (MsgId). The data is represented by a 48-character hexadecimal string, for example: -u 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF. The default is none. This option can also be combined with the CorrelId, UserData and FileName, to form a logical AND when making a file selection, however the unique MsgId is normally sufficient when used on its own for selecting a single 'other' message.
- s Specifies a string of arbitrary user data that is used to distinguish between files. This can be useful for differentiating between two files with the same name on the specified queue, for example:

```
"XYZ 'Internal' sales-figures for 2003"
XYZ
"'Internal'"
"sale"
"sales-figures"
2003
03
```

This option selects by locating an occurrence of the specified UserData with part or all of the sender's UserData. The match is case-sensitive. Wild cards are not supported. The default is any. This option can also be combined with the CorrelId, MsgId, and FileName, to form a logical AND when making a file selection.
- r Renames or moves a file to a new location. The directory must already have been created. The default filename used for the renamed or moved file is the same as the filename used in the current directory.
- f Selects a file to receive by its filename (FileName). The filename can include a path to the file. A filename with embedded spaces must be quoted, for example: "Long File Names.txt". There is no wildcard expansion. The default is any file on the queue. This option can also be combined with the CorrelId, MsgId and UserData, to form a logical AND when making a file selection. This option only receives one file.

Usage

```
mqftrcvc -q QueueName [-m QMgrName] [-c CorrelId] [-u MsgId] [-s Comments]
          [-v] [-a | -l | -i | -o | [[-g | -e] [-y]] | -d] [-r FileName]
          [-f FileName]
```

Examples

This example lists all files from MY.QUEUE:

```
mqftrcvc -q MY.QUEUE -a
```

This example gets the first complete file:

```
mqftrcvc -q MY.QUEUE -g
```

This example gets the complete file spreadsheet.xls:

```
mqftrcvc -q MY.QUEUE -g -f spreadsheet.xls
```

This example gets the complete file spreadsheet.xls also marked 'urgent':

```
mqftrcvc -q MY.QUEUE -g -f spreadsheet.xls -s "URGENT"
```

Usage notes

- Before using this command, ensure that the required WebSphere MQ queue managers, queues and channels are available. In addition, ensure the MQSERVER environment variable or channel definition tables have been set.
- Use this command to receive binary files and fixed or variable record length files.
- Options must have a leading dash or a forward slash.
- Options for Windows are not case sensitive.
- Options for UNIX are case sensitive.
- Options cannot be repeated.
- Options and arguments must be separated by one or more spaces.
- If running a WebSphere MQ Express v5.3 Server, use the "Receive file command (mqftrcv)" on page 28 to receive files.

How Express File Transfer works

Express File Transfer sender does the following:

1. Finds the file to send in the local file system.
2. Reads the contents of the file into memory.
3. Breaks the file into manageable pieces, and puts each piece into a separate message.
4. Puts each message onto the specified queue.

Each message contains a header (control information) in addition to the file data. Because the file data is binary, no data manipulation or conversion is required.

The Express File Transfer receiver does the following:

1. Receives all queued messages that make up the original file.
2. Reconstructs the original file from these messages in the correct order, and writes the file to the local file system.
3. Deletes the messages from the local queue.

A command option also exists for extracting the file without deleting it from the queue. If there is a problem writing the file, all messages that comprise the original file remain on the queue. The received file is an identical copy of the file that was originally sent.

Understanding return codes

The command line programs mqftsnd, mqftsndc, mqftrcv and mqftrcvc might return values indicating the success or failure of the command. These are return codes, and are useful when writing scripts or batch files that run the Express File Transfer commands.

For a more comprehensive list of return codes, see the product readme.

Code	Message	Comments
0	Successful operation	The file was successfully sent or received.
36	Invalid arguments supplied	One or more options were incorrectly specified when issuing the send or receive command. Check the options used and reissue the command.
40	Queue manager not available	Check that the queue manager exists and that the listener is running.
69	Storage not available	There is insufficient memory to perform the requested action. 1. Check the message size is not excessive 2. Close other applications and try the command again
71	Unexpected error	Retry the send or receive action and if it still fails. Stop and restart everything and try again.
163	Queue name required	A queue name was not specified when issuing a send or receive command. Reissue the command with the QueueName option.
164	Cannot open queue	Check that the queue exists.
165	Cannot open file	Check that the file exists, that it is in the correct location and has the appropriate file permissions.
166	Cannot put to queue	1. Check that Queue Manager has the sufficient log space in case of large persistence files 2. Check that the queue does not have put inhibited 3. Check the queue is not full 4. Check the message size on the send command does not exceed the message size of the queue 5. Check the user has sufficient WMQ authority to put messages on the queue
167	No file name specified (Send)	A file name was not specified when issuing a send command. Reissue the command with the FileName option.
168	Message length is too small to send data	Increase the message size and resend with a send command, using the -l MessageSize option to specify a larger message size.
169	Sending file has changed	The file being sent has been changed before the complete file has been sent. Check the file for integrity and reissue the send command.
170	Cannot get from queue	The list, get, delete or extract request has failed. 1. Check the queue does have get inhibited 2. Check the user has sufficient WMQ authority to get messages from the queue
171	Cannot write to file	The get or extract request has failed. 1. Check that the file is not write-protected. In Windows Explorer, right-click the file name and select Properties . Check the user has sufficient authority to write to the destination file system. 2. Check the destination file system exists 3. Check the destination file system is not full

Code	Message	Comments
172	CorrelId is invalid	Check that a valid correlation ID has been specified when receiving a file with the -c option. It must be 48 characters in length. Use the -v option of the receive command to display the correlation ID.
173	MsgId is invalid	Check that a valid message ID has been specified when receiving an 'other' message with the -u option. It must be 48 characters in length.
174	No messages to receive	There are no FTA files on the specified queue. Check with the sender that the file was actually sent.
175	File for delete is not unique	There is more than one file on the specified queue with the same file name. Use the -s or -c options to uniquely identify the file.

Frequently asked questions (FAQ)

Can I send multiple versions of the same file to the same destination?	Yes you can send a file to the same destination more than once. You can use the Comments field to help identify different versions of the same file.
How do I differentiate between multiple versions of the same file?	The sender might have used the Comments field to help you identify different versions of the same file. Once you have received the file, you can view details of the file in the Session log .
What happens if a sent file is not completely received?	When a complete file is being received, an operation may fail for many reasons, for example a WebSphere MQ error, a problem writing the file because the disk is full, or a permissions failure. If this happens, the operation is backed out, meaning that all the messages comprising the original file are reinstated on the queue. This allows the problem to be fixed, and the operation to be retried until it succeeds, without affecting the availability of the file.
Can I look at a file on the queue rather than removing it from the queue?	Yes, by using the mqftrcv or mqftrcvc command with the -l option, to list all complete files with their status in queue order. Alternatively, you can use the Express File Transfer GUI.
Can I get an incomplete file?	Yes, by using the mqftrcv or mqftrcvc command with the -e option, to extract a copy of the file from the queue. Alternatively, you can use the Express File Transfer GUI.
What happens if the network goes down?	<p>If you are using Express File Transfer from a WebSphere MQ Server and a network failure occurs during a file transfer, if the file being sent is persistent, it remains on the sender queue ready to be resent when the network becomes operational again. Files are sent as persistent messages by default.</p> <p>If you are using Express File Transfer from a WebSphere MQ Client and the network fails, the send or receive command will fail and you will have to reissue the command.</p>
What is the maximum size of a file I can send?	It is possible to send files of up to 2 GB, although this is not recommended. For normal purposes it is possible to send and receive very large files (a file of 10 MB is regarded as large). The default message size is 100 KB.

<p>Why is the text file I sent from Linux to Windows corrupted?</p>	<p>The file is not corrupted. Remember that files are transferred without any modifications being made to them. The most likely explanation is that you are viewing the text file in an editor that does not distinguish between how different platforms encode line endings. On Windows, a text file has two characters: a carriage return and a line feed, at the end of each line of text. Linux and UNIX systems use a different encoding where each line ends in a single line feed character. Most text editors can be set up to perform the conversion automatically, but some editors do not allow this conversion. In this case, a Linux text file may appear to be corrupted (the editor is looking for a carriage return and a line feed but only sees a line feed. You should refer to the documentation or help for the editor you are using.</p> <p>If the text file was sent from a Windows machine to a Linux machine, most editors on Linux show the carriage return character as a control character.</p>
--	---

Tutorials

These tutorials show how to perform basic tasks such as creating a queue manager, creating a queue, creating a channel, putting a message onto a queue, and getting a message from a queue.

Each tutorial is dividing into several sub-tasks, which can be achieved using either:

- The WebSphere MQ Explorer graphical interface (only available with WebSphere MQ Server on Windows platforms).
- The WebSphere MQ Script Commands (MQSC) command line interface (available on Windows and Linux platforms with WebSphere MQ Server or Client installed).

Each tutorial in this set builds upon MQ objects that have been set up during previous tutorials. It is therefore recommended that these tutorials are completed sequentially.

The tutorials are designed to get you started with WebSphere MQ Express v5.3, and do not cover the more complex messaging scenarios.

Sending a message to a local queue

This tutorial shows you how to set up a queue manager, and queues on a local, standalone installation that has no communication links with any other installations.

When you have set up these objects, you will put a test message onto the queue, and verify that the test message was received.

When you have completed this tutorial, you should have a basic understanding of how WebSphere MQ messaging works in a simple messaging topology that has a queue manager with local queues.

Creating the queue manager

Before creating a queue manager, you must ensure WebSphere MQ Express v5.3 is correctly installed.

This topic shows how to create a queue manager called QM_APPLE.

This task can be achieved by using either:

- The WebSphere MQ Explorer graphical interface
- WebSphere MQ Script Command (MQSC) command line interface

Creating the queue manager using MQ Explorer

1. Start the WebSphere MQ Explorer.
2. Expand the **Queue Managers** folder.
3. From the popup menu select **New > Queue Manager**.
4. Type QM_APPLE into the **Queue Manager** field.
5. Select the check box **Make this the default queue manager**.
6. Click the **Next** button twice to go to the Step 3 dialog.
7. Ensure that the **Start Queue Manager** check box is selected.
8. Click the **Next** button to go to the Step 4 dialog.
9. Ensure that the **Create listener configured for TCP/IP** check box is selected.
10. Click **Finish**.

An icon representing this queue manager appears in the MQ Explorer, and the queue manager automatically starts running a few seconds after you create it.

Creating the queue manager using MQSC

Open a command prompt, and follow these steps:

1. Create a default queue manager called QM_APPLE by typing the command:

```
crtmqm -q QM_APPLE
```

Messages tell you that the queue has been created and that the default WebSphere MQ objects have been created.

2. Start this queue manager by typing the command:

```
strmqm
```

A message tells you when the queue manager has started.

You have now created a queue manager with the name QM_APPLE. The next task is to create a local queue that this queue manager will manage.

Creating the local queue

Before creating a local queue on the queue manager, you must already have created the queue manager in the previous task.

This topic shows how to create a local queue called Q1 on the queue manager called QM_APPLE

This task can be achieved by using either:

- The WebSphere MQ Explorer graphical interface
- WebSphere MQ Script Command (MQSC) command line interface

Creating the local queue using MQ Explorer

1. In the left MQ Explorer pane, expand the **Queue Managers** folder.
2. Select the queue manager QM_APPLE that you created earlier.
3. Right click the **Queues** folder.
4. From the popup menu, select **New > Local Queue**.
5. In the **Queue Name** field, type Q1.
6. Click **OK**.

Creating the local queue using MQSC

Open a command prompt and follow these steps:

1. Enable MQSC commands by typing the command:

```
runmqsc
```

2. Type the following command:

```
define qlocal (Q1)
```

Messages tell you that the queue has been created and that the default WebSphere MQ objects have been created.

3. Stop MQSC by typing the command:

```
end
```

You have now created a local queue called Q1. The next task is to put a test message to this newly created local queue.

Putting a test message onto the local queue

Before putting a test message onto the queue, you must already have completed the other tasks in the tutorial (create the queue manager, create the queue).

This topic shows how to put a test message on the local queue Q1.

This task can be achieved by using either:

- The WebSphere MQ Explorer graphical interface
- WebSphere MQ Script Command (MQSC) command line interface

Putting a test message onto the queue using MQ Explorer

1. In the left MQ Explorer pane, expand the **Queue Managers** folder.
2. Expand queue manager QM_APPLE you created earlier.
3. Select the **Queues** folder.
4. In the right pane, right click the local queue Q1 you created earlier.
5. From the popup menu, select **Put Test Message**.
6. In the **Message Data** field, type some text, for example this is a test message.
7. Click **OK**.

Putting a test message onto the queue using MQSC

The **amqspout** sample program is used to put a message onto the queue that you created.

(On Windows the sample programs are installed by default with WebSphere MQ Server or Client. On Linux, the samples programs RPM need to be installed.)

Open a command prompt and follow these steps:

1. Start the **amqspout** sample program as follows:
 - On Linux, change to the `/opt/mqm/samp/bin` directory and type the command: `./amqspout Q1`
 - On Windows, type the command: `amqspout Q1`

The following messages are displayed:

```
Sample amqspu0 start
target queue is Q1
```

2. Type some message text on one or more lines, then press **Enter** twice. The following message is displayed:

```
Sample amqspu0 end
```

You have now created a test message and put it onto the local queue. The next task is to verify that the test message was received.

Verifying that the test message was sent

Before getting the test message from the local queue, you must already have completed the other tasks in the tutorial (create the queue manager, create the queue, and put the test message on the local queue).

This topic shows how to verify that the test message was sent.

This task can be achieved by using either:

- The WebSphere MQ Explorer graphical interface
- WebSphere MQ Script Command (MQSC) command line interface

Verifying that the test message was sent using MQ Explorer

1. Expand the **Queue Managers** folder.
2. Expand QM_APPLE.
3. Select the **Queues** folder.
4. In the right pane, double-click Q1.
5. The Message browser opens to show the list of messages on Q1. Double-click the last message to view its contents.

Verifying that the test message was sent using MQSC

The **amqsget** sample program is used to get the message back from the queue.

Open a command prompt and follow these steps:

Start the **amqsget** sample program:

- On Windows, type the following command: `amqsget Q1`
- On Linux, change to the `/opt/mqm/samp/bin` directory and type the following command: `./amqsget Q1`

The sample program starts, and your message is displayed along with any other messages on this queue. After a pause of 15 seconds, the sample ends and the command prompt is displayed again.

Congratulations! You have now completed this tutorial.

Sending a message to a remote queue

This tutorial builds upon MQ objects that will have been set up during the previous tutorial; you will need to have completed “Sending a message to a local queue” on page 37.

This tutorial shows you how to set up messaging between queues on two machines, so that a message created on one machine is delivered to a queue on the other machine (this queue is referred to as a *remote* queue).

Important: During this tutorial, you will use the machine on which you created queue manager QM_APPLE and local queue Q1 as the receiving machine.

You will set up a queue manager and queues (a remote queue definition and a transmission queue) on the sending machine. You will then define a message channel. Finally, you will put a test message onto the sending machine, and get it from the queue on the receiving machine.

Before starting this tutorial, you will need to find out the name which identifies the receiving machine on the network from your system administrator.

When you have completed this tutorial, you should have a basic understanding of how to set up and use WebSphere MQ messaging using a remote queue definition.

Creating the queue manager

Before creating a queue manager on the sending machine, you must ensure the WebSphere MQ Express v5.3 Server is correctly installed.

In this part of the tutorial you will create a queue manager QM_ORANGE on the sending machine.

This task can be achieved by using either:

- The WebSphere MQ Explorer graphical interface
- WebSphere MQ Script Command (MQSC) command line interface

Creating the sending queue manager using MQ Explorer:

On the sending machine:

1. Start the WebSphere MQ Explorer.
2. Expand **WebSphere MQ**.
3. Right-click on the **Queue Managers** folder.
4. From the popup menu select **New > Queue Manager**.
5. Type QM_ORANGE into the **Queue Manager** field.
6. Select the check box **Make this the default queue manager**.
7. Click the **Next** button twice, to go to the Step 3 dialog.
8. Ensure the **Start Queue Manager** check box is selected.
9. Click the **Next** button to go to the Step 4 dialog.
10. Ensure that the **Create listener configured for TCP/IP** check box is selected
11. Click **Finish**.

An icon representing this queue manager appears in the MQ Explorer, and the queue manager automatically starts running a few seconds after you create it.

Creating the sending queue manager using MQSC:

Open a command prompt on the sending machine and follow these steps:

1. Create a default queue manager called QM_ORANGE by typing the command:

```
crtmqm -q QM_ORANGE
```

Messages tell you that the queue has been created and that the default WebSphere MQ objects have been created.

2. Start this queue manager by typing the command:

```
strmqm
```

A message tells you when the queue manager has started.

You have now created the sending queue manager. The next task is to create the queues on this queue manager.

Creating the queues on the sending queue manager

Before creating the queues on the sending queue manager, you must already have created the queue manager in the previous task.

In this part of the tutorial you will create a remote queue definition and a transmission queue on the sending queue manager.

This task can be achieved by using either:

- The WebSphere MQ Explorer graphical interface
- WebSphere MQ Script Command (MQSC) command line interface

Creating the queues on the sending queue manager using MQ Explorer:

On the sending queue manager:

1. Expand the **Queue Managers** folder.
2. Expand the queue manager **QM_ORANGE** that you created earlier.
3. Right click the **Queues** folder.
4. From the popup menu, select **New > Remote Queue Definition**.
5. In the **Queue Name** field, type **Q1**.
6. In the **Remote Queue Name** field, type **Q1**.
7. In the **Remote Queue Manager Name** field, type **QM_APPLE**.
8. Ensure that the **Transmission queue name** field is left blank.
9. Click **OK**. You have now created the remote queue definition.
10. Select the **QM_ORANGE** queue manager again.
11. Right click the **Queues** folder.
12. From the popup menu, select **New > Local Queue**.
13. In the **Queue Name** field, type **QM_APPLE**.
14. In the **Usage** field, select **Transmission** from the drop-down list.
15. Click **OK**. You have now created the transmission queue on the local machine.

Creating the queues on the sending queue manager using MQSC:

Open a command prompt on the sending machine and follow these steps:

1. Start MQSC by typing the command:

```
runmqsc
```

A message tells you that an MQSC session has started.

2. Define a local queue called **QM_APPLE** by typing the following command:

```
define qlocal (QM_APPLE) usage (xmitq)
```

A message tells you when the queue has been created.

3. Define a remote queue definition by typing the following command:

```
define qremote (Q1) rname (Q1) rqmname(QM_APPLE) xmitq (QM_APPLE)
```

You have now created the queues on the sending queue manager. The next task is to create the message channel between the sending and receiving queue managers.

Creating a message channel

Before creating a message channel you must already have completed the other tasks in the tutorial (create the sending queue manager and create the queues).

In this part of the tutorial you will create a message channel between the sending and receiving queue managers.

This task can be achieved by using either:

- The WebSphere MQ Explorer graphical interface
- WebSphere MQ Script Command (MQSC) command line interface

Creating the message channel using MQ Explorer:

On the receiving queue manager:

1. Expand **Queue Managers** folder.
2. Expand the queue manager **QM_APPLE** that you created earlier.
3. Expand the **Advanced** folder.
4. Right click the **Channels** folder.
5. From the popup menu select **New > Receiver Channel**.
6. In the **Channel Name** field, type **QM_ORANGE.QM_APPLE**.
7. Click **OK**. You have now created the receiver channel on the receiving machine.

On the sending queue manager:

8. Expand the **Queue Managers** folder.
9. Expand the queue manager **QM_ORANGE** that you created earlier.
10. Expand the **Advanced** folder.
11. Right click the **Channels** folder.
12. From the popup menu select **New > Sender Channel**.
13. In the **Channel Name** field, type **QM_ORANGE.QM_APPLE**.
14. In the **Connection Name** field, type the computer name or IP address of the receiving machine (you should already have obtained this with your system administrator's help).
15. In the **Transmission queue** drop down list, select **QM_APPLE**.
16. Click **OK**.
17. Click the channel folder.
18. Right-click **QM_ORANGE.QM_APPLE**.
19. From the pop-up menu, click **Start**.
20. Click **OK**. You have now created the sender channel on the sending machine.

Creating the message channel using MQSC:

Open a command prompt on the **receiving** machine and follow these steps:

1. Start MQSC by typing the command:

```
runmqsc
```

A message tells you that an MQSC session has started.

2. Define a receiving channel by typing the following command:
`define channel (QM_ORANGE.QM_APPLE) chltype (RCVR) trdtype (TCP)`

A message tells you when the channel has been created.

3. Stop MQSC by typing:
`end`

Some messages are displayed followed by the command prompt.

Open a command prompt on the **sending** machine and follow these steps:

4. Start MQSC by typing the command:

```
runmqsc
```

A message tells you that an MQSC session has started.

5. Define a sender channel by typing the following command:
`define channel (QM_ORANGE.QM_APPLE) chltype (sdr) conname ('con-name') xmitq (QM_APPLE) trdtype (tcp)`

The value *con-name* is the TCP address of the receiver queue manager.

6. Start the channel by typing the following command:
`start channel (QM_ORANGE.QM_APPLE)`
7. Stop MQSC by typing:
`end`

Some messages are displayed followed by the command prompt.

You have now created all the WebSphere MQ objects required for messages to be sent from the sending queue manager QM_ORANGE to the queue Q1 on the receiving queue manager QM_APPLE. The next task is to send a test message.

Putting a test message onto the queue

Before putting a test message onto the queue, you must already have completed the other tasks in the tutorial (create the sending queue manager and its queues, and the message channel).

This topic shows how to put a test message on the remote queue.

This task can be achieved by using either:

- The WebSphere MQ Explorer graphical interface
- WebSphere MQ Script Command (MQSC) command line interface

Putting a test message onto the queue using MQ Explorer:

1. In the left MQ Explorer pane, expand the **Queue Managers** folder.
2. Expand queue manager QM_ORANGE you created earlier.
3. Select the **Queues** folder.
4. In the right pane, right click the local queue Q1 you created earlier.
5. From the popup menu, select **Put Test Message**.
6. In the **Message Data** field, type some text, for example this is a test message.
7. Click **OK**.

Putting a test message onto the queue using MQSC:

The **amqsput** sample program is used to put a message onto the queue that you created.

On Windows the sample programs are installed by default with WebSphere MQ Server or Client. On Linux, the samples programs RPM need to be installed.

Open a command prompt and follow these steps:

1. Start the **amqsput** sample program as follows:

- On Linux, change to the /opt/mqm/samp/bin directory and type the command: `./amqsput Q1`
- On Windows, type the command: `amqsput Q1`

The following messages are displayed:

```
Sample amqsput0 start
target queue is Q1
```

2. Type some message text on one or more lines, then press **Enter** twice. The following message is displayed:

```
Sample amqsput0 end
```

You have now created a test message and put it onto the remote queue. The next task is to verify that the test message was received.

Verifying that the test message was sent

Before getting the test message from the queue, you must already have completed the other tasks in the tutorial (create the sending queue manager and its queues, create the message channel and put the test message on the remote queue).

This topic shows how to verify that the test message was sent.

This task can be achieved by using either:

- The WebSphere MQ Explorer graphical interface
- WebSphere MQ Script Command (MQSC) command line interface

Verifying that the test message was sent using MQ Explorer:

On the receiving queue manager:

1. Expand the **Queue Managers** folder.
2. Expand queue manager QM_APPLE.
3. Select the **Queues** folder.
4. In the right pane, double-click the queue Q1. The **Message browser** opens. It contains details of the test message in table form, including the message text.
5. Double click the last message in the list to view its contents

Verifying that the test message was sent using MQSC:

The **amqsget** sample program is used to get the message back from the queue.

Open a command prompt and follow these steps:

Start the **amqsget** sample program as follows:

- On Linux, change to the /opt/mqm/samp/bin directory and type the command: `./amqsget Q1`
- On Windows, type the command: `amqsget Q1`

The sample program starts, and your message is displayed along with any other messages on this queue. After a short pause, the sample program ends and the command prompt is displayed again.

Congratulations! You have now completed this tutorial.

Sending a message on a client-server configuration

This tutorial builds upon MQ objects that will have been set up during the previous tutorial - you will need to have completed "Sending a message to a remote queue" on page 40.

This tutorial shows you how to set up messaging between client and server machines.

Important: This tutorial shows you how to work with a client-server installation, where the client is a third machine with WebSphere MQ Client installed, and the server is the machine which has the queue manager QM_ORANGE defined on it.

You will set up the server by creating a server connection channel. You will then set up the client by defining the MQSERVER environment variable. Finally, you will put a test message from the Client onto QM_ORANGE which will send it to queue Q1 on QM_APPLE. Finally, you will verify the message was sent.

Before starting this tutorial, you will need to find out the name which identifies the server which hosts queue manager QM_ORANGE on the network from your system administrator.

When you have completed this tutorial, you should have a basic understanding of how to set up messaging on a WebSphere MQ Express v.5.3 client server configuration.

Setting up the server

In this part of the tutorial you will set up the queue manager QM_ORANGE on the server machine to enable client connections to it. This will involve configuring a server connection channel.

This task can be achieved by using either:

- The WebSphere MQ Explorer graphical interface
- WebSphere MQ Script Command (MQSC) command line interface

Setting up the server using MQ Explorer:

On the server machine that hosts queue manager QM_ORANGE:

1. Expand the **Queue Managers** folder.
2. Expand QM_ORANGE.
3. Expand the **Advanced** folder.
4. Right-click the **Channels** folder.
5. From the popup menu, select **New > Server Connection Channel**.
6. Type CLIENT.QM_ORANGE into the **Channel Name** field.
7. Click the **MCA** tab at the top of the page.
8. Type your Windows login name (or a user name in the mqm group) into the **MCA User ID** field.
9. Click **OK**.

For more information about the MCAUSER ID, see "Access control" in the *WebSphere MQ Clients* book.

Setting up the server using MQSC:

Open a command prompt on the receiving machine and follow these steps:

1. Start MQSC by typing the command:

```
runmqsc
```

A message tells you that an MQSC session has started. MQSC has no command prompt.

2. Define a server-connection channel by typing the following command on one line:

```
define channel(CLIENT.QM_ORANGE) chltype(SVRCONN) trptype(TCP) mcauser('mqm')
```

Windows users should type their Windows login name (or a valid mqm user name) in place of mqm.

A message tells you when the channel has been created.

3. Stop MQSC by typing:

```
end
```

Some messages are displayed followed by the command prompt.

4. Start a listener by typing the following command:

```
runmq1sr -t tcp
```

You have now finished setting up the server. The next task is to set up the client.

Setting up the client

Before setting up the Client to communicate with queue manager QM_ORANGE, you must ensure the WebSphere MQ Client has been installed on the client machine.

In this part of the tutorial you will set up the client component using the MQSERVER environment variable. You will need to find out the network name of the machine which hosts queue manager QM_ORANGE from your system administrator.

If the Client is on Windows:

1. Open the Control Panel:
 - On Windows 2000 or 2003, click **Start > Settings > Control Panel**
 - On Windows XP, click **Start > Control Panel**
2. Double-click **System**.
3. Click the **Advanced** tab.
4. Click **Environment Variables**.
5. In the User Variables pane, click **New**.
6. Type MQSERVER into the Variable Name field.
7. Type CLIENT.QM_ORANGE/TCP/*hostname* into the Variable Value field, where *hostname* is the computer name or IP address that identifies the machine hosting queue manager QM_ORANGE.
8. Click **OK**. The MQSERVER environment variable will appear in the User Variables pane.

If the Client is on Linux:

9. Log in as the user who will be running Express File Transfer, who must be a member of the mqm group.
10. Open a command prompt
11. Type `cd $HOME`

12. Use a text editor to edit the profile. This example assumes that you are using the bash shell, so you need to edit the file `$HOME/.bashrc`. However if you are using a different system shell, consult your system documentation. Add the following text to the bottom of the file:

```
MQSERVER=CLIENT.QM_ORANGE/TCP/'hostname' export MQSERVER
```

Replace *hostname* with the name that identifies the server machine on the network.

13. Close the command prompt.
14. Log out and log back in for the change to take effect.

You have now set up the client and server components needed. The next task is to send a message from the client to the server queue manager QM_ORANGE.

Putting a test message onto the queue

Before putting a test message onto the queue, you must already have completed the other tasks in the tutorial (setting up the client and server).

In this part of the tutorial, you will send a message destined for queue Q1 from the client to the server queue manager QM_ORANGE, which will use the remote queue definition and other MQ objects defined in earlier tutorials to route it onto queue manager QM_APPLE and to queue Q1.

(On Windows the sample programs are installed by default with WebSphere MQ Server or Client. On Linux, the sample programs RPM will need to be installed.)

Open a command prompt on the client and follow these steps:

1. Start the **amqsputc** sample program as follows:
 - On Linux, change to the `/opt/mqm/samp/bin` directory and type the command: `./amqsputc Q1`
 - On Windows, type the command: `amqsputc Q1`

The following messages are displayed:

```
Sample AMQSPUT0 start
target queue is Q1
```

2. Type some message text on one or more lines, then press **Enter** twice. The following message is displayed:

```
Sample AMQSPUT0 end
```

You have now created a test message and sent it to the server queue manager QM_ORANGE, which will route it onto queue Q1 on queue manager QM_APPLE. The next task is to verify that the test message was received.

Verifying that the test message was sent

Before getting the test message from the queue, you must already have completed the other tasks in the tutorial (set up the client and server and send the test message from the client to the server queue manager).

This topic shows how to verify that the test message was sent.

This task can be achieved by using either:

- The WebSphere MQ Explorer graphical interface
- WebSphere MQ Script Command (MQSC) command line interface

Verifying that the test message was sent using MQ Explorer:

On the machine that hosts the queue manager QM_APPLE:

1. Expand **WebSphere MQ**.
2. Expand the **Queue Managers** folder.
3. Expand QM_APPLE.
4. Select the **Queues** folder.
5. In the right pane, right double-click Q1. The **Message browser** opens to show the list of messages on Q1.
6. Double click the last message in the list to view its contents.

Verifying that the test message was sent using MQSC:

The **amqsget** sample program is used to get the message back from the queue.

Open a command prompt and start the **amqsget** sample program as follows:

- On Windows, type the following command: `amqsget Q1`
- On Linux, change to the `/opt/mqm/samp/bin` directory and type the following command: `./amqsget Q1`

The sample program starts, and your message is displayed along with any other messages on this queue. After a pause of 15 seconds, the sample ends and the command prompt is displayed again.

Congratulations! You have now completed this tutorial.

Exchanging files using Express File Transfer

This tutorial shows you how to use the Express File Transfer graphical interface to send and receive files using the WebSphere MQ environment set up in the previous tutorials.

This tutorial builds upon MQ objects that will have been set up during the previous tutorial; you will need to have completed the “Sending a message to a remote queue” on page 40 and “Sending a message on a client-server configuration” on page 46 tutorials.

Using this tutorial you will set up and use Express File Transfer to send and receive files on a server-server WebSphere MQ installation, and a client-server WebSphere MQ installation. During this tutorial, you will set up Express File Transfer, send a file, receive the file, and save it to a destination of your choice.

When you have completed the tutorial, you should have a basic understanding of how to work with Express File Transfer.

Server to server file exchange

In this part of the tutorial you will set up the Express File Transfer graphical interface to send files from queue manager QM_ORANGE to queue manager QM_APPLE, using the WebSphere MQ environment set up in previous tutorials.

Setting up Express File Transfer:

The first task is to set up Express File Transfer on the sending and receiving machines.

On the machine that hosts queue manager QM_ORANGE:

1. To Start Express File Transfer:
 - On Windows, click **Start > Programs > IBM WebSphere MQ Express > Express File Transfer**
 - On Linux, open a shell window, and start Express File Transfer by typing the command: `mqftapp`

2.
 - If this is the first time you have started Express File Transfer, the Welcome prompt is displayed. Click **OK** to display the **Setup** dialog.
 - If this is not the first time you have started Express File Transfer, select **File > Setup**, and click **Change**.
3. In the Queue Manager field, type QM_ORANGE.
4. Select **Local**.
5. Click **OK**.
6. In the **Destinations** list, select the check box for Q1.
7. Click **OK**. The Express File Transfer main window now opens, with the queue you selected listed as a destination.

On the machine that hosts queue manager QM_APPLE:

8. Start Express File Transfer.
9.
 - If this is the first time you have started Express File Transfer, the Welcome prompt is displayed. Click **OK** to display the **Setup** dialog.
 - If this is not the first time you have started Express File Transfer, select **File > Setup**, and click **Change**.
10. Type QM_APPLE into the **Queue Manager** field.
11. Select **Local**.
12. Click **OK**.
13. Click the **Sources** tab, and select the check box for Q1.
14. Click **OK**.

You have now setup Express File Transfer on the sending and receiving machines. The next task is to send a file using Express File Transfer.

Using Express File Transfer to send a file:

Before sending a file to the receiving machine using Express File Transfer, you must already have completed “Setting up Express File Transfer” on page 49.

This topic shows you how to use Express File Transfer to send a file to the receiving queue manager.

On the sending machine that hosts queue manager QM_ORANGE:

1. Start Express File Transfer. By default the **Send** tab is displayed.
2. Click **Browse**.
3. Select a file to send.
4. Click **Open**.
5. Add an optional comment, for example: Hi Cathy, here are today’s sales figures, in the **Comments** field.
6. Click Q1 in the **Destination** list.
7. Click **Send**.

When the file has been sent, an entry will appear in the **Session log**.

The next task is to receive the file on the receiving machine using Express File Transfer.

Using Express File Transfer to receive a file:

Before receiving a file from the sending machine, you must already have completed “Using Express File Transfer to send a file” on page 50.

This topic shows you how to use Express File Transfer to receive the file you sent from the sending machine.

On the receiving machine that hosts queue manager QM_APPLE:

1. Start Express File Transfer. By default the **Send** tab is displayed.
2. Select the **Receive** tab.
3. Ensure Q1 is selected in the **Files In** drop down list.
4. Select the file you have just sent from QM_ORANGE.
5. Click **Receive**.
6. Select a location where you want to save the file.
7. Click **Save**.

When the file has been received, an entry will appear in the **Session log**.

Congratulations! You have now completed this tutorial.

Client to server file exchange

In this part of the tutorial you will set up the Express File Transfer graphical interface to send files from a WebSphere MQ Client to the remote queue manager QM_APPLE, using the WebSphere MQ environment set up in previous tutorials.

Setting up Express File Transfer:

This topic shows you how to set up Express File Transfer on WebSphere MQ Client.

On the WebSphere MQ Client:

1. Start Express File Transfer.
2.
 - If this is the first time you have started Express File Transfer, the Welcome prompt is displayed. Click **OK** to display the **Setup** dialog.
 - If this is not the first time you have started Express File Transfer, select **File > Setup**, and click **Change**.
3. In the Queue Manager field, type QM_ORANGE.
4. Select **Remote**.
5. Click **OK**.
6. In the **Destinations** list, select the check box for Q1.
7. Click **OK**. The Express File Transfer main window now opens, with the queue you selected listed as a destination.

You have now set up Express File Transfer on the WebSphere MQ Client. The next task is to send a file to the receiving queue manager (QM_APPLE) using Express File Transfer.

Sending a file:

Before sending a file to the receiving machine using Express File Transfer, you must already have completed “Setting up Express File Transfer.”

This topic shows you how to use Express File Transfer to send a file to the receiving queue manager.

On the WebSphere MQ Client:

1. Start Express File Transfer. By default the **Send** tab is displayed.
2. Click **Browse**.
3. Select a file to send.
4. Click **Open**.
5. Click Q1 in the **Destinations** list.
6. Click **Send**.

When the file has been sent, an entry will appear in the **Session log**.

The next task is to receive the file on the receiving machine using Express File Transfer.

Receiving a file:

Before receiving a file from the WebSphere MQ Client, you must already have completed “Sending a file” on page 51.

This topic shows you how to use Express File Transfer to receive the file you sent from the WebSphere MQ Client.

On the receiving machine that hosts queue manager QM_APPLE:

1. Start Express File Transfer. By default the **Send** tab is displayed.
2. Select the **Receive** tab.
3. Ensure Q1 is selected in the **Files In** drop down list.
4. Select the file you have just sent from the WebSphere MQ Client.
5. Click **Receive**.
6. Select a location where you want to save the file.
7. Click **Save**. You have now saved the file on the remote computer.

When the file has been received, an entry will appear in the **Session log**.

Congratulations! You have now completed this tutorial.

Index

A

- about the tutorials 37
- about the WebSphere MQ Express Center 1
- about WebSphere MQ Express v5.3 4
- accessibility and the Express Center 2
- administering WebSphere MQ objects 17
- Administration 17
- application development 18
- application interaction with WebSphere MQ 16
- assured delivery 8
- asynchronous communication 7
- asynchrony 7

B

- balancing workload 16
- business integration 7
- business needs 6

C

- clients 13
- command line interface 26
- communicating using a WebSphere MQ Client 13
- communicating with a local queue manager 13
- communication 12, 13
- communications 7
- Creating WebSphere MQ objects 17

D

- developing applications 18

E

- Express Center, accessibility 2
- Express Center, trademarks 2
- Express Center, using the HTML Help version 1
- Express Center, using the PDF version 2
- Express File Transfer 19
 - file processing 34
 - frequently asked questions (FAQ) 36
 - listing sent and received files 23
 - receiving a file 22
 - sending a file 22
- Express File Transfer graphical interface options 23
- Express File Transfer GUI
 - initial setup 21
- Express File Transfer preparation 20
- Express File Transfer tutorials
 - about 49
 - getting a test message (client-server) 50

- external communications 12

F

- file processing in Express File Transfer 34
- file transfer using WebSphere MQ 19
- frequently asked questions 5
- frequently asked questions about Express File Transfer 36
- fundamentals of messaging 12

G

- graphical interface for Express File Transfer 23
- graphical user interface 21

I

- integration 7
- interacting with WebSphere MQ 16
- introduction to messages 9
- introduction to messaging and queuing 8
- introduction to queues 9

J

- Java Messaging Service (JMS) 19
- JMS (Java Messaging Service) 19

L

- listing sent and received files with Express File Transfer 23
- local queue manager 13

M

- making a queue location transparent 14
- message 9
- message delivery 8
- messaging 8
- messaging fundamentals 12
- messaging scenario 11
- messaging topology 10

P

- Preparing for Express File Transfer 20
- product tour 6

Q

- queue 9
- queue administration 15

- queue location 14
- queue managers 12
- queueing 8

R

- receive file command
 - a (list all files) 28, 32
 - c (select on correlation ID) 28, 32
 - d (delete) 28, 32
 - e (extract) 28, 32
 - f (filename) 28, 32
 - g (get) 28, 32
 - i (list incomplete files) 28, 32
 - l (list complete files) 28, 32
 - m (queue manager name) 28, 32
 - o (list other messages) 28, 32
 - q (queue name) 28, 32
 - r (rename) 28, 32
 - s (string) 28, 32
 - u (select on message ID) 28, 32
 - v (verbose) 28, 32
 - y (replace) 28, 32
 - mqftrcv 28
 - mqftrcvc 32
- receiving a file 19
- receiving a file with Express File Transfer 22
- remote queue 14

S

- scalability 8
- scenario for messaging 11
- send file command
 - f (filename) 26, 30
 - l (message size) 26, 30
 - m (queue manager name) 26, 30
 - p (persistence) 26, 30
 - q (queue name) 26, 30
 - s (string) 26, 30
 - t (target queue manager name) 26, 30
 - v (verbose) 26, 30
 - mqftsnd 26
 - mqftsndc 30
- sending a file 19
- sending a file with Express File Transfer 22
- sending a message to a remote queue 14
- setting up Express File Transfer GUI 21
- simplifying queue administration 15
- system scalability 8

T

- topology for messaging 10
- trademarks 2
- transferring a file as an MQ message 19

transferring files using WebSphere
MQ 19

tutorials

- about 37
- creating a message channel (remote queue) 43
- creating a queue (local queue) 38
- creating a queue manager (local queue) 37
- creating a transmission queue 42
- creating queue manager (remote queue) 41
- creating queues (remote queue) 42
- getting a test message (client-server) 49
- putting a test message (client-server) 48
- putting a test message (local queue) 39
- putting a test message (remote queue) 44
- sending a message (client-server) 46

tutorials (*continued*)

- sending a message (local queue) 37
- sending a message (remote queue) 40
- setting up the client (client-server) 47
- setting up the server (client-server) 46
- verifying a test message (client-server) 48
- verifying a test message (local queue) 40
- verifying a test message (remote queue) 45

U

Using the command line 26
using the Express Center 1, 2

Using the graphical interface 21

W

WebSphere MQ Express Center, about 1
WebSphere MQ Express Center,
performance restrictions 4
WebSphere MQ Express v5.3, about 4
WebSphere MQ Express v5.3, FAQ 5
WebSphere MQ Express v5.3, meeting
business needs 6
WebSphere MQ Express v5.3, product
tour 6
workload balacing between
programs 16