

IBM MQ Managed File Transfer

MQMFT CreateTransfer GUI (mqjfx)

Ver 1.1
28 Mar, 2017

S.N.Software Inc.
e-mail : support@pulsarintegration.com

Program Version 1.1.0.0

Tested MQ version / OS / Java of this program.

- Red Hat Enterprise Linux Server 6.7 / Oracle Java 1.8.0_121 / IBM MQ 8.0
- Windows 7 Professional / Oracle Java 1.8.0_121 / IBM MQ 9.0

*This program uses JavaFX library for GUI. Since IBM Java 8 does not support JavaFx at this time, it can not be used for this program.

Table of contents

About this program	5
Note	5
1. The program execution environment.....	8
MQ Install Environment.....	8
Reference of MQ library	8
User group.....	9
2. Common settings.....	10
Setting connection parameters to the command queue manager	11
Table 2.1 Connection parameters to the command queue manager.....	11
Setting connection parameters to the coordination queue manager	12
Table 2.2 Connection parameters to the coordination queue manager	13
Saving connection parameters	13
Table 2.3 XML Mapping of Connection Information	15
Subscribe to File Transfer Log	15
Specifying agent properties.....	16
Table 2.4 Agent Properties	17
3. How to create a transfer	18
Procedure for transfer request	18
• Setup XML.....	18
• Specifying agent properties	19
• Specifying parameters for transfer.....	20
Table 3.1 Example of transfer request parameters	20
• Confirmation of the transfer result	22
Saving transfer parameters	26
Table 3.2 XML mapping of transfer requests.....	27
Generate transfer request XML file.....	34
Table 3.3 List of menu items	35
4. Example of transfer request.....	35

Ex. 1: Transfer specifying the source / destination queue manager	35
Ex. 2: Scheduling transfer (Case 1).....	39
Ex. 3: Scheduling transfer (Case 2).....	43
Ex. 4: Trigger transfer.....	47
Ex. 5: Transfer when the destination is Queue (Case 1).....	50
Table 4.1 Message property of destination message	53
Table 4.2 Return Codes	58
Table 4.3 Intermediate reply codes.....	58
Ex. 6: Transfer when the destination is Queue (Case 2)	59
Ex. 7: Invoke the program.....	62
Ex. 8: Using transfer definition file (Case 1) (Specify only source and destination) ...	73
Ex. 9: Using transfer definition file (Case 2) (Specify all required information)	78
Ex.10: Specify code conversion	82
Ex.11: Recursively transfer files in subdirectories	85
Ex.12: Handle group messages when the source specification is queue (Case 1)	90
Ex.13: Handle group messages when the source specification is queue (Case 2)	93
Conclusion.....	98

About this program

This program executes on the Java GUI the same function as the `fteCreateTransfer` (create new file transfer) command provided by IBM MQ Managed File Transfer. It creates a transfer request XML, sends it to the command queue of the command queue manager, and receives a response message.

In addition, it also has the following functions.

- It subscribes the transfer log from the coordination queue manager and displays it on the GUI.
- The connection parameters to the command queue manager and the coordination queue manager can be saved / loaded.
- Transfer parameters can be saved / loaded.
- It can output transfer request XML file.
- Partial information of Agent Properties of Source / Dist. Agent can be specified from the GUI.
- When a transfer request is made, it is possible to specify a transfer definition file. (Also possible with `fteCreateTransfer` command).
- Receive a response message such as acknowledgment of transfer completion, and display its contents on the GUI.

This document does not describe the details of IBM MQ Managed File Transfer itself. Please refer to the product manual as necessary.

All documents of product versions are referenced from the following URL.

IBM MQ and WebSphere MQ

<http://www-01.ibm.com/software/integration/wmq/library/index.html>

The MQI test program (`mqqpgf`) command and the MQAI program (`mqqpcf`) command is also used in the test example of this document. For details of `mqqpgf` and `mqqpcf` command, please refer to the document "MQI test program (`mqqpgf`)" and "MQAI program (`mqqpcf`)".

Note

*In this document, express the XML hierarchy by connecting tags with '-'.

For example, in the case of XML below, it is expressed as "request - managedTransfer - originator - hostname", but since the two tags at the head of "request - managedTransfer"

are the root elements of all other tags, they are omitted in the following description Then it is written like the "originator - hostname".

```
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer mqPassword="password" mqUserId="mftuser" transferDefin
itionFile="c:\work\javafx\TransferDefinitionFile.xml" transferTemplateFilePath="
C:\work\javafx\work1.xml">
    <originator>
      <hostName>localhost</hostName>
      ....
    
```

1. The program execution environment

As a prerequisite to using mqjfx, please use the appropriate JRE environment described on page 2. In addition, it is necessary that the MQJava library (jar file) that matches the system requirements of its JRE is installed by appropriate method. (If you are using Oracle Java 8, jar files provided with MQ 8.0.0.4 or higher is required.)

The test target is MQFTE 7.0.4 or WebSphere MQ 7.5 or higher Managed file transfer.

Please refer to mqjfx_start.ksh(Unix), mqjfx_start.bat(Windows) for how to start the program.

MQ Install Environment

Depending on the environment in use, it may be necessary to set up the environment of the MQ installation to be used. For example, when connecting to the command or coordination queue manager in bind mode, it is necessary to refer to the native library. When bind mode is used, if the native library can not be referenced, the following exception is raised:

```
Com.ibm.mq.jmqi.local.LocalMQ $4: CC=2; RC=2495; AMQ8598: Loading
WebSphere MQ native JNI library 'mqjbnd' failed.
```

If MQ instration environment has not been loaded by the startup environment such as the login shell, please run the following command and set up the MQ environment.

```
$ . <MQ Install Directory>/bin/setmqenv -s -k
```

It can also be set up in the start script of this program. (See mqjfx_start.ksh / mqjfx_start.bat)

Reference of MQ library

Please check CLASSPATH setting to the following Jar file.

```
<MQ Install Directory>/lib/com.ibm.mq.headers.jar
<MQ Install Directory>/lib/com.ibm.mq.jar
<MQ Install Directory>/lib/com.ibm.mq.jmqi.jar
<MQ Install Directory>/lib/com.ibm.mq.pcf.jar
<MQ Install Directory>/lib/com.ibm.mqjms.jar
```


mqjfx11.jar

User group

To run the program, it is necessary for the executing user to have appropriate access authority to the queue manager. If you do not know the details of the required authorities, please use the user who is a member of the mqm group (MQ administrator) or add the user in the mqm group.

2. Common settings

When starting mqjfx for the first time, the following Window will be displayed.

Before creating the transfer, specify connection parameters for the command queue manager to put a transfer XML and for the coordination queue manager to subscribe to the transfer log.

The screenshot shows a window titled "WebSphere MQ Managed File Transfer Test program". It has a menu bar with "File", "Edit", and "Help". Below the menu bar are several input fields and buttons:

- Command Queue Manager: [text box] Queue: [text box]
- ☒ Bind Mode ☐ Client Mode
- Host Name: [text box] Channel: [text box]
- Port: [text box] User Id: [text box]
- [Execute] [Cancel]

Below these fields is a tabbed interface with four tabs: "Create Transfer" (selected), "Agent Properties", "Response Message", and "File Transfer Log".

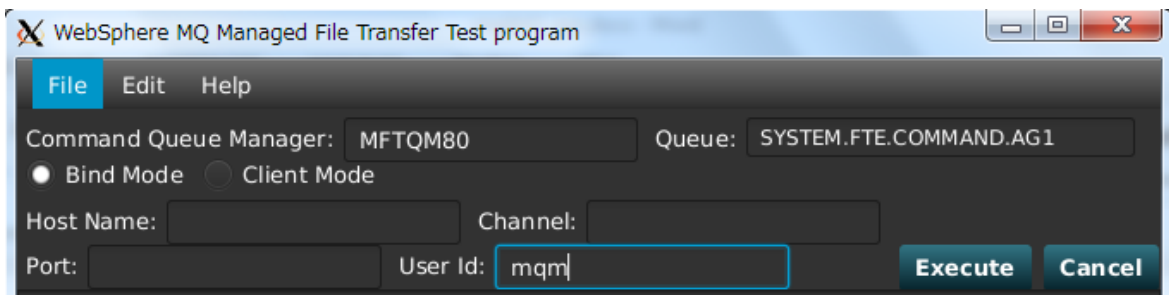
The "Create Transfer" tab contains the following sections and fields:

- Source Specification: [text box]
- Agent Specification
 - Source Agent: [text box]
 - Source Agent Qmgr: [text box]
 - Destination Agent: [text box]
 - Destination Agent Qmgr: [text box]
- Generating Transfer Templates
 - Transfer Template File Path: [text box]
- Scheduling Transfers
 - Schedule Start Time: [text box]
 - Time Base: [dropdown menu]
 - Occurrence Interval: [dropdown menu]
 - Occurrence Frequency: [text box]
 - Occurrence Count: [text box]

Setting connection parameters to the command queue manager

Below is a example for bind connection. (Client connection is also available.)
Note) "Queue:" specified here is the command queue of the source agent.

Command Queue Manager: MFTQM80
Queue: SYSTEM.FTE.COMMAND.AG1
Bind Mode: check
User Id: mqm



If both "MQ UserId" and "MQ Password" on [Create Transfer] tab are specified, it will be connected to the Command Queue Manager using MQCSP authentication mode.

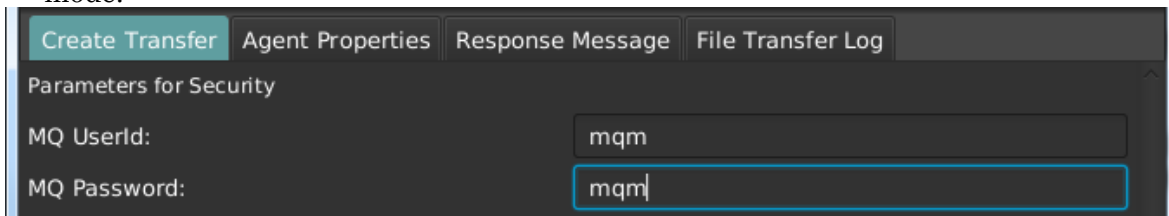


Table 2.1 Connection parameters to the command queue manager

Field	Description	Note	Mandatory/Optional
Command Queue Manager	Name of the command queue manager.	When creating transfer request XML, the [Create Transfer] tab "Source Agent Qmgr:" is given priority.	Mandatory
Queue	Source agent command queue	Always specify the queue on the source agent side.	Mandatory

Bind Mode	Connect with binding mode	Can be specified when the same node as the command queue manager.	Select either "Bind Mode" or "Client Mode"
Client Mode	Connect with Client Mode	Host Name, Channel, Port must be specified	same as above
Host Name	Host Name or IP address		Required for Client Mode
Channel	Server Connect Channel Name		same as above
Port	Listener Port Number		same as above
User Id	Used for client connection		

Setting connection parameters to the coordination queue manager

In order to subscribe to the transfer log, set the following as parameters to connect to the coordination queue manager. Below is an example for bind connection. (Client connection is also available.)

[File Transfer Log] Tab

Coordination Queue Manager: MFTQM80

Bind Mode: check

User Id: mqm

The screenshot shows a software interface with a dark background and light-colored text. At the top, there are four tabs: "Create Transfer", "Agent Properties", "Response Message", and "File Transfer Log" (which is highlighted in light blue). Below the tabs, the text "Coordination Queue Manager:" is followed by a text box containing "MFTQM80". To the right of this are two buttons: "Start Monitor" and "Stop Monitor". Below these, there are two radio buttons: "Bind Mode" (which is selected, indicated by a filled circle) and "Client Mode". To the right of the radio buttons, the word "Stopped" is written in red. Below the radio buttons, there are four text boxes: "Host Name:", "Channel:", "Port:", and "User Id:". The "User Id:" box contains the text "mqm" and is highlighted with a blue border.

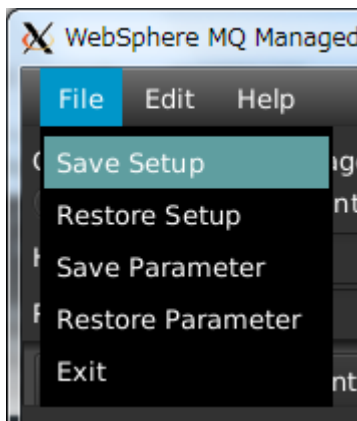
Table 2.2 Connection parameters to the coordination queue manager

Field	Description	Note	Mandatory/Optional
Coordination Queue Manager	Name of Coordination Queue Manager		Mandatory
Bind Mode	Connect with binding mode	Can be specified when the same node as the coordination queue manager.	Select either "Bind Mode" or "Client Mode"
Client Mode	Connect with Client Mode	Host Name, Channel, Port must be specified	same as above
Host Name	Host Name or IP address		Required for Client Mode
Channel	Server Connect Channel Name		same as above
Port	Listener Port Number		same as above
User Id	Used for client connection		

Saving connection parameters

Save the connection parameters entered on the application window in a file.

This program (mqjfx) automatically reads "defaultSetup.xml" in the current directory at startup, so manual input from the next time onwards can be omitted.



In the dialog displayed, go to the startup directory of the program and save it as "defaultSetup.xml".

*If "defaultParameter.xml" (described later) exists, "hostname" and "User Id" of Command Queue Manager are overwritten by the originator - hostname tag and userID tag on the defaultParameter.xml file.

```
----
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi=....>
  <managedTransfer ....>
    <originator>
      <hostName>host1</hostName>
      <userID>mqm</userID>
    </originator>
  </managedTransfer>
</request>
----
```

Saved connection information :

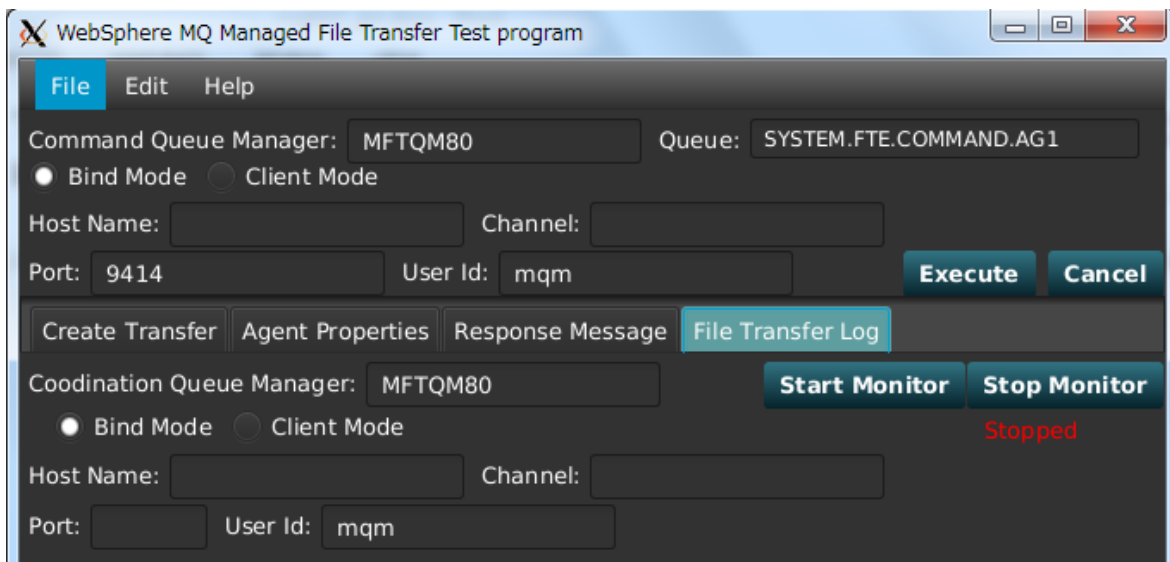
```
----
$ cat ./mqjfx2.0/mqjfx11/dist/defaultSetup.xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<initialize>
  <CommandQmgrEntry mode="bind">
    <CmdQmgr>MFTQM80</CmdQmgr>
    <CmdQueue>SYSTEM.FTE.COMMAND.AG1</CmdQueue>
    <CmdQmgrHost/>
    <CmdQmgrChannel/>
    <CmdQmgrPort/>
    <CmdQmgrUserID>mqm</CmdQmgrUserID>
  </CommandQmgrEntry>
  <CoordinationQmgrEntry mode="bind">
    <CoordQmgr>MFTQM80</CoordQmgr>
    <CoordQmgrHost/>
    <CoordQmgrChannel/>
    <CoordQmgrPort/>
    <CoordQmgrUserID>mqm</CoordQmgrUserID>
  </CoordinationQmgrEntry>
</initialize>
----
```

Table 2.3 XML Mapping of Connection Information		
Field	XML tag / Attribute	Note
Connection parameters to the command queue manager Initialize - CommandQmgrEntry		
Command Queue Manager	CmdQmgr	
Queue	CmdQueue	Agent queue manager command queue
Bind Mode/Client Mode	mode attribute of CommandQmgrEntry	"bind"/"client"
Host Name	CmdQmgrHost	
Channel	CmdQmgrChannel	
Port	CmdQmgrPort	
User Id	CmdQmgrUserID	
Connection parameters to the coordination queue manager Initialize - CoordinationQmgrEntry		
Coordination Queue Manager	CoodQmgr	
Bind Mode/Client Mode	mode attribute of CoordinationQmgrEntry	"bind"/"client"
Host Name	CoodQmgrHost	
Channel	CoodQmgrChannel	
Port	CoodQmgrPort	
User Id	CoodQmgrUserID	

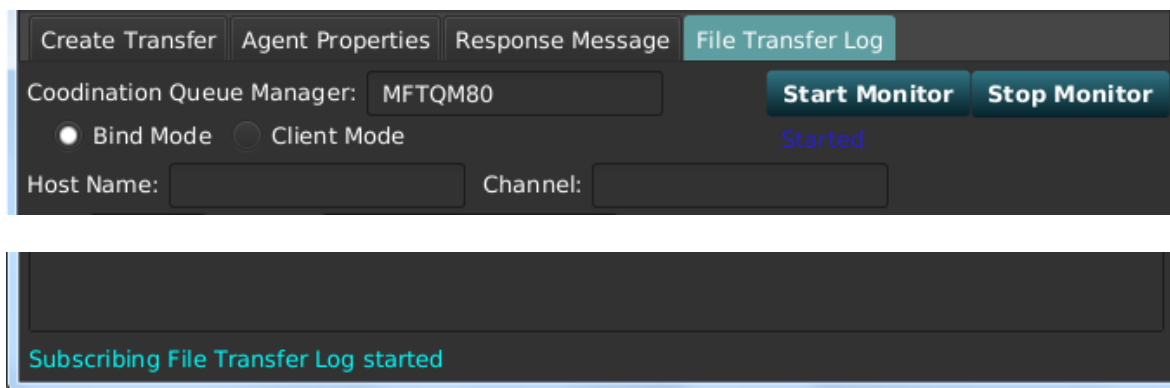
Subscribe to File Transfer Log

Start subscribing to the file transfer log.

Select the "File Transfer Log" tab.



Click the "Start Monitor" toggle button.



When the subscribe is successfully started, the status display changes from "Stopped" to "Started" and "Subscribing File Transfer Log started" appears in the status bar.

Specifying agent properties

The same parameters as the agent.properties can be specified for the transfer request XML.

Select "Agent Properties" tab.

Create Transfer
Agent Properties
Response Message
File Transfer Log

Source Agent

agentQMgr: MFTQM80

agentQMgrPort:

agentQMgrHost: hostname

agentQMgrChannel:

Destination Agent

agentQMgr: MFTQM80

agentQMgrPort: 9414

agentQMgrHost: hostname

agentQMgrChannel: MFTQM80.SVRCONN

The parameters are set as attributes of the originator - sourceAgent and destinationAgent of the transfer request XML.

Table 2.4 Agent Properties

Field	Description	XML Attribute	Note
Source Agent			
agentQmgr	Agent queue manager name	QMgr	When creating transfer request XML, [Create Transfer] tab "Source Agent Qmgr:" is given priority.
agentQMgrPort	Port number used for client connection to the agent queue manager		Currently unused
agentQMgrHost	Host name or IP address of agent queue manager	hostName	When createing transfer request XML, Command Queue Manager [Host Name:] is given priority.
agentQMgrChannel	SVRCONN channel		Currently unused

	name used to connect to the agent queue manager		
Destination Agent			
agentQmgr	Agent queue manager name	QMgr	When creating transfer request XML, [Create Transfer] tab "Destination Agent Qmgr:" is given priority.
agentQMGrPort	Port number used for client connection to the agent queue manager	portNumber	
agentQMGrHost	Host name or IP address of agent queue manager	hostName	
agentQMGrChannel	SVRCONN channel name used to connect to the agent queue manager	channel	

3. How to create a transfer

As an example, create a transfer specifying the transfer destination directory by the `fteCreateTransfer` command as shown below.

```
fteCreateTransfer -p MFTQM -sa AG1 -da AG2 -dd /home/mftuser/mft/to
/home/mftuser/mft/from/test1.txt
```

Procedure for transfer request

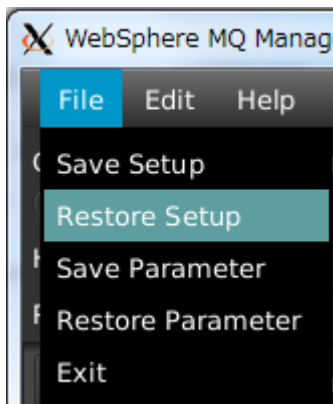
- Setup XML

The setup XML to use is below.

```
$ cat ExampleSetup.xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<initialize>
  <CommandQmgrEntry mode="bind">
    <CmdQmgr>MFTQM80</CmdQmgr>
    <CmdQueue>SYSTEM.FTE.COMMAND.AG1</CmdQueue>
    <CmdQmgrHost/>
    <CmdQmgrChannel/>
    <CmdQmgrPort/>
    <CmdQmgrUserID>mqm</CmdQmgrUserID>
  </CommandQmgrEntry>
  <CoordinationQmgrEntry mode="bind">
    <CoordQmgr>MFTQM80</CoordQmgr>
    <CoordQmgrHost/>
    <CoordQmgrChannel/>
    <CoordQmgrPort/>
    <CoordQmgrUserID>mqm</CoordQmgrUserID>
  </CoordinationQmgrEntry>
</initialize>
----
```

Load ExampleSetup.xml.

If defaultSetup.xml has the same contents, you can omit this step.



In the dialog displayed, specify "ExampleSetup.xml".

*If "defaultParameter.xml" (described later) exists, "hostname" and "User Id" of Command Queue Manager are overwritten by the originator - hostname tag and the userID tag.

- **Specifying agent properties**

See "Specifying Agent Properties" earlier in chapter.2. "Common settings".

- **Specifying parameters for transfer**

This program does not automatically set defaults for required parameters. It is necessary to set all parameters.

For this time, specify the parameters in the table below from the GUI.

Table 3.1 Example of transfer request parameters			
Parameter	Description	Sample Parameter	Note
Source Specification	One or more file specifications that determine the source, or sources, for the file transfer.	/home/mft/from/test1.txt	mandatory
Agent Specification			
Source Agent	The name of the agent that the source files are transferred from.	AG1	mandatory
Destination Agent	The name of the agent that the files are transferred to.	AG2	mandatory
Specifying Transfer Options			
Checksum Algorithm	Specifies whether a checksum algorithm is run on the file transfer data to check the integrity of the transferred files.	MD5	
Specifying the Destination			
Destination Type	Specify one of the following: •Transfer Definition File •file •directory •dataset	directory	mandatory

	<ul style="list-style-type: none"> •pds •filespace •queue 		
Destination Directory	The name of the directory the file is transferred to.	/home/mft/to	
Destination file behavior	Specifies the action that is taken if a destination file exists on the destination system.	overwrite	
Type of File Transfer	Specifies the type of file transfer: binary mode or text mode.	binary	
Specifying the Source			
Source File Disposition	Specifies the action that is taken on a source file when that source file is successfully transferred to its destination.	leave	

Create Transfer

Agent Properties

Response Message

File Transfer Log

Source Specification:

/home/mft/from/test1.txt

Agent Specification

Source Agent:

AG1

Source Agent Qmgr:

Destination Agent:

AG2

Destination Agent Qmgr:

Specifying Transfer Options

Job Name:

User-defined Metadata:

Checksum Algorithm:

MD5

Specifying the Destination

Destination Type:

directory

Transfer Definition File:

Destination File:

Destination Directory:

/home/mft/to

Destination Sequential Data Set:

Destination Partitioned Data Set:

Destination User:

Destination Queue:

Persistent:

Hexadecimal Delimiter:

Pattern:

Position:

Destination File Behavior:

overwrite

Type of File Transfer:

binary

Specifying the Source

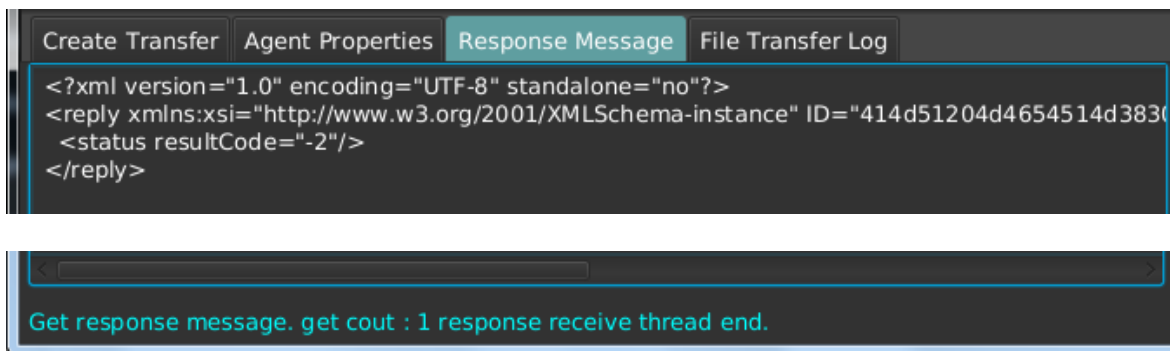
Source File Disposition:

leave

- **Confirmation of the transfer result**

Click the "Execute" button to start the transfer.

On the Response Message tab, verify that the response message was successfully received.

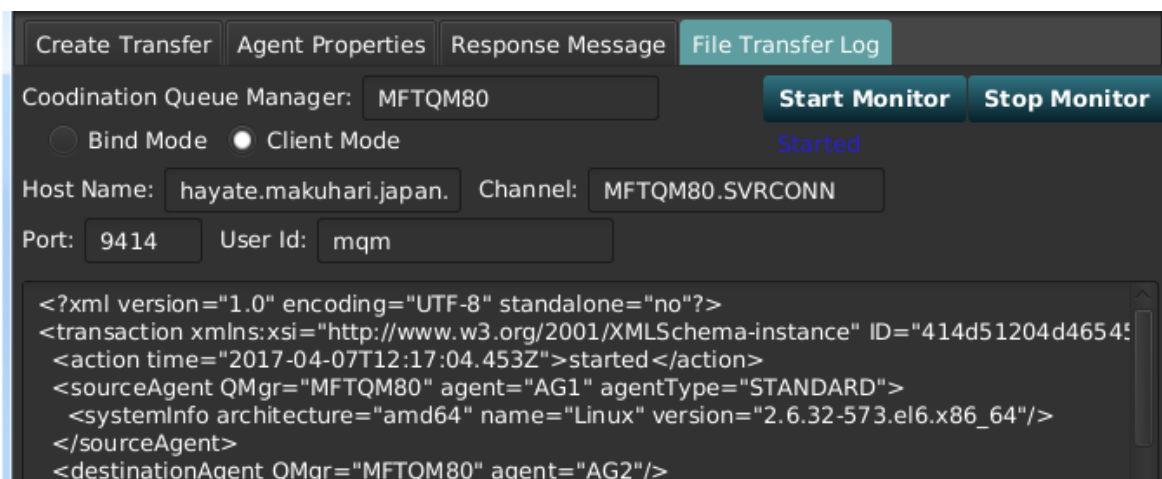


Below is a sample response message.

```
-----
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<reply xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ID="414d51204d4654514d383020202020d562e758032d0020" version="6.00" xsi:noNamespaceSchemaLocation="Reply.xsd">
  <status resultCode="-2"/>
</reply>
-----
```

"-2" in resultCode indicates ACK of Intermediate reply codes (The request has been received but is pending completion).

In the [File Transfer Log] tab, check the transfer log.



Three transfer logs with the <action> tags "started", "progress", "completed" are published.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<transaction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ID="414d51
204d4654514d383020202020d562e758032d0020" agentRole="sourceAgent" version
="6.00" xsi:noNamespaceSchemaLocation="TransferLog.xsd">
  <action time="2017-04-07T10:10:03.646Z">started</action>
  <sourceAgent QMgr="MFTQM80" agent="AG1" agentType="STANDARD">
    <systemInfo architecture="amd64" name="Linux" version="2.6.32-573.el6.x86_6
4"/>
  </sourceAgent>
  <destinationAgent QMgr="MFTQM80" agent="AG2"/>
  <originator>
    <hostName>hostname</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <transferSet bytesSent="0" startTime="2017-04-07T10:10:03.647Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">AG1</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">AG2</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">hostname</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d51204d4654514d38302020
202020d562e758032d0020</metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
</transaction>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<transaction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ID="414d51
204d4654514d383020202020d562e758032d0020" agentRole="sourceAgent" version
="6.00" xsi:noNamespaceSchemaLocation="TransferLog.xsd">
  <action time="2017-04-07T10:10:04.785Z">progress</action>
  <sourceAgent QMgr="MFTQM80" agent="AG1" agentType="STANDARD">
    <systemInfo architecture="amd64" name="Linux" version="2.6.32-573.el6.x86_6
4"/>
  </sourceAgent>
  <destinationAgent QMgr="MFTQM80" agent="AG2" agentType="STANDARD">
    <systemInfo architecture="amd64" name="Linux" version="2.6.32-573.el6.x86_6
4"/>
  </destinationAgent>
  <originator>
    <hostName>hostname</hostName>
```



```

    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <transferSet bytesSent="174" index="0" size="1" startTime="2017-04-07T10:10:0
3.647Z" total="1">
    <item mode="binary">
      <source disposition="leave" type="file">
        <file last-modified="2017-03-29T11:02:51.000Z" size="174">/home/mft/from/t
est1.txt</file>
        <checksum method="MD5">a091aaf05ca2bff25d077ab65fd3252b</checksum
      >
      </source>
      <destination exist="overwrite" type="file">
        <file last-modified="2017-04-07T10:10:04.000Z" size="174">/home/mft/to/test
1.txt</file>
        <checksum method="MD5">a091aaf05ca2bff25d077ab65fd3252b</checksum
      >
      </destination>
      <status resultCode="0"/>
    </item>
  </transferSet>
</transaction>

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<transaction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ID="414d51
204d4654514d383020202020d562e758032d0020" agentRole="sourceAgent" version
="6.00" xsi:noNamespaceSchemaLocation="TransferLog.xsd">
  <action time="2017-04-07T10:10:04.893Z">completed</action>
  <sourceAgent QMgr="MFTQM80" agent="AG1" agentType="STANDARD">
    <systemInfo architecture="amd64" name="Linux" version="2.6.32-573.el6.x86_6
4"/>
  </sourceAgent>
  <destinationAgent QMgr="MFTQM80" agent="AG2" agentType="STANDARD">
    <systemInfo architecture="amd64" name="Linux" version="2.6.32-573.el6.x86_6
4"/>
  </destinationAgent>
  <originator>
    <hostName>hostname </hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <status resultCode="0">
    <supplement>BFGRP0032I: </supplement>
  </status>
  <transferSet bytesSent="174" startTime="2017-04-07T10:10:03.647Z" total="1">

```

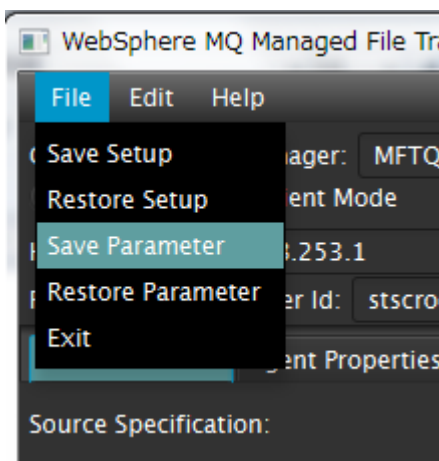
```

<metaDataSet>
  <metaData key="com.ibm.wmqfte.SourceAgent">AG1</metaData>
  <metaData key="com.ibm.wmqfte.DestinationAgent">AG2</metaData>
  <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
  <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
  <metaData key="com.ibm.wmqfte.OriginatingHost">hostname </metaData>
  <metaData key="com.ibm.wmqfte.TransferId">414d51204d4654514d38302020
2020d562e758032d0020</metaData>
  <metaData key="com.ibm.wmqfte.Priority">0</metaData>
</metaDataSet>
</transferSet>
<statistics>
  <actualStartTime>2017-04-07T10:10:04.225Z</actualStartTime>
  <retryCount>0</retryCount>
  <numFileFailures>0</numFileFailures>
  <numFileWarnings>0</numFileWarnings>
</statistics>
</transaction>
----
```

Saving transfer parameters

You can save the transfer request in an xml file of arbitrary name in order to omit the input when the next same test is carried out. If you save it as "defaultParameter.xml" in the startup directory, it will be loaded automatically at startup.

The parameters of "Create Transfer" and "Agent Properties" tab are saved.



In the dialog displayed, move to the program start directory and save it as "defaultParameter.xml".

Saved transfer parameters:

\$ cat defaultParameter.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00" xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>hostname</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="MFTQM80" agent="AG1"/>
    <destinationAgent QMgr="MFTQM80" agent="AG2" channel="MFTQM80.SVRC
ONN" hostName="hostname" portNumber="1414"/>
    <transferSet>
      <item checksumMethod="MD5" mode="binary">
        <source disposition="leave" recursive="false">
          <file>/home/mft/from/test1.txt</file>
        </source>
        <destination exist="overwrite" type="directory">
          <file>/home/mft/to</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Table 3.2 XML mapping of transfer requests		
XML Tag /(Attribute)	Field	Note
request - managedTransfer (The following attributes of this tag are unique to "Save Parameter" of this test program)		
(transferDefinitionFile)	[Specifying the Destination] Transfer Definition File	
(transferTemplateFilePath)	[Generationg Transfer Templates] Transfer Template File Path	

(mqUserId)	[Parameters for Security] MQ UserId	If both mqUserId and mqPassword are specified, it is connected to the command queue manager using MQCSP authentication mode.
(mqPassword)	[Parameters for Security] MQ Password	same as above
request - managedTransfer - originator		
hostName	[Command Queue Manager] Host Name	If not specified, [Agent Properties]Tab - [Source Agent] - AgentQmgrHost is set.
userID	user.name	Java System Properties
request - managedTransfer - schedule		
submit	[Scheduling Transfers] Schedule Start Time	yyyy-mm-ddThh:mm
(timezone)	user.timezone	It is specified with Java system property "-Duser.timezone = ...".
(timebase)	[Scheduling Transfers] Time Base	The following can be selected: Admin, source, UTC If not specified, "admin" is set.
request - managedTransfer - schedule - repeat		
Frequency	[Scheduling Transfers] Occurrence Frequency	
Frequency/(interval)	[Scheduling Transfers] Occurrence Interval	The following can be selected: Minutes, hours, days, weeks, months, years
expireCount	[Scheduling Transfers] Occurrence Count	
expireTime	[Scheduling Transfers] Schedule End Time	yyyy-mm-ddThh:mm
request - managedTransfer - sourceAgent		
(QMgr)	[Agent Specification] Source Agent Qmgr	If not specified, [Agent Properties]Tab - [Source Agent] - agentQmgr is set.
(agent)	[Agent Specification] Source Agent	
request - managedTransfer - destinationAgent		

(QMgr)	[Agent Specification] Destination Agent Qmgr	If not specified, [Agent Properties]Tab - [Destination Agent] - agentQmgr is set.
(agent)	[Agent Specification] Destination Agent	
(channel)	[Agent Properties]Tab [Destination Agent] agentQMgrChannel	
(portNumber)	[Agent Properties]Tab [Destination Agent] agentQMgrPort	
(hostName)	[Agent Properties]Tab [Destination Agent] agentQMgrHost	
request - managedTransfer - trigger		
(log)	[Triggering Transfers] Create Transfer Log Entries	
fileExist	[Triggering Transfers] Condition, Namelist	Format: Condition,Namelist
fileExist/ (value)	Available Condition is the following three types: file=exist file!=exist filesize>=size	"Namelist" is set in "fileExist".
fileExist/ (comparison)	It is also possible to specify KB, MB, GB for size. Example: file=exist,c:¥work¥file1.txt,c: ¥work¥file2.txt filesize>=10KB,c:¥work¥file1. txt,c:¥work¥file2.txt	In the case of "file", "exist" is set. In the case of "filesize", "size" is set like "10KB". Either "=", "!=" , "> =" is set.
request - managedTransfer - transferSet		
(priority)	[Specifying Transfer Options] Transfer Priority	
request - managedTransfer - transferSet - metaDataSet		

metadata (key)	[Specifying Transfer Options] User-defined Metadata	Multiple specifications can be specified by separating key = value with a comma as shown below: meta1=data1,meta2=data2,meta3=dat a3 The specified number of metadata tags are created.
request - managedTransfer - transferSet - preSourceCall - command request - managedTransfer - transferSet - postSourceCall - command request - managedTransfer - transferSet - preDestinationCall - command request - managedTransfer - transferSet - postDestinationCall - command		
(type)	[Invoking Programs] Pre Source Cell	[type:] executable(default), antscript, jcl
(name)	Specify as follows. (Same as fteCreateTransfer) [type:]commandspec[,retrycount][,retrywait][,successrc]]	command of <i>Commandspec</i> executable: command[(arg1,arg2,...)] antscript: command[(name1=var1 target1,name2=var2 target2,...)] jcl: command os4690background: command[(arg1,arg2,...)]
(retryCount)		[retrycount]
(retryWait)		[retrywait]
(successRC)		[,successrc]
property	Used if "type" is antscript.	
property/ (name)		command[(name1 =var1 target1, name2 =var2 target2,...)]
property/ (value)		command[(name1= var1 target1 ,name2= var2 target2 ,...)]
argument	Used if "type" is executable, os4690background.	command[(arg1,arg2 ,...)]
request - managedTransfer - transferSet - item *This tag is created as many as the Source Specification		
checksumM ethod	[Specifying Transfer Options] Checksum Algorithm	The following can be selected: MD5, none

mode	[Specifying the Destination] Type of File Transfer	The following can be selected: binary, text
request - managedTransfer - transferSet - item - source		
(disposition)	[Specifying the Source] Source File Disposition	The following can be selected: leave, delete
(recursive)	[Specifying the Source] Recursively transfer files in subdirectories	true or false
file	Source Specification	If "The source of a transfer is a queue" is not selected, a file name is specified on the left. When multiple items are specified by delimiting them with ";", this tag is created corresponding to that number.
file/ (encoding)	[Specifying the Source] Source Character Encoding	
file/ (keepTrailingSpaces)	[Specifying the Source] Trailing spaces are kept on source records	true or false
file/ (delimiterType)	[Specifying the Source] Delimiter	"Binary" is used if the left parameter is specified.
file/ (delimiter)		It is specified in the following format: x0D,x0A
file/ (delimiterPosition)	[Specifying the Source] Position to insert source record delimiters	The following can be selected: prefix, postfix
queue	Source Specification	If "The source of a transfer is a queue" is selected, a queue name is specified on the left. When multiple items are specified by delimiting them with ";", this tag is created corresponding to that number.
queue/ (useGroups)	[Specifying the Source] The messages are grouped by WMQ groupID	If the left checkbox is checked, "true" is set.
queue/ (groupId)		\$ {GROUPID}" is set if the left is checked.

queue/ (delimiterType)	[Specifying the Destination] Type of File Transfer	The same value as "request - managedTransfer - transferSet - item - mode" is set.
queue/ (delimited)	*If "Type of File Transfer" is "text": [Specifying the Source] Text Delimiter	String literals Java escape sequence such as ¥u007d¥n can be included in the delimiter.
	*If "Type of File Transfer" is "binary": [Specifying the Source] Hexadecimal Delimiter	Specify like x08,xA4.
queue/ (delimiterPosition)	[Specifying the Source] Position of source text and binary delimiters	The following can be selected: prefix, postfix
queue/ (waitTime)	[Specifying the Source] Wait time	Specify in seconds.
request - managedTransfer - transferSet - item - destination		
(exist)	[Specifying the Destination] Destination File Behavior	The following can be selected: error, overwrite
(type)	[Specifying the Destination] Destination Type	The following can be selected: file, directory dataset, pds, filespace, queue
file	[Specifying the Destination] Destination File	If "type" is any of file, directory, dataset, pds, one of the left parameters is set.
	Destination Directory	
	Destination Sequential Data Set	
	Destination Partitioned Data Set	
file/ (truncateRecords)	[Specifying the Destination] Destination records are truncated	If "type" is file, directory dataset, pds, and the left check box is checked, truncateRecords = "true" is set.
filespace/ name	[Specifying the Destination] Destination User	Used when "type" is filespace.
queue	[Specifying the Destination] Destination Queue.	Used when "type" is queue.

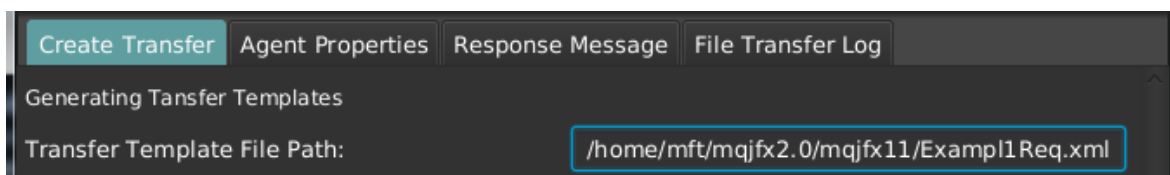
queue/ (delimiterType)	[Specifying Transfer Options] Message Size	If left parameter is specified, "size" is set.
	[Specifying the Destination] Hexadecimal Delimiter	If left parameter is specified, "binary " is set.
	[Specifying the Destination] Pattern	If left parameter is specified, "text " is set.
queue/ (delimiter)	[Specifying Transfer Options] Message Size	'B', 'K', 'M' can also be used like "10 K".
	[Specifying the Destination] Hexadecimal Delimiter	Specify a comma-separated list of hexadecimal bytes (eg x3e,x20,x20,xbf)
	[Specifying the Destination] Pattern	Java regular expression to use when splitting a text file into multiple messages.
queue/ (setMqProps)	[Specifying Transfer Options] Sets Message Properties	The following can be selected: true, false
queue/ (persistent)	[Specifying the Destination] Persistent	The following can be selected: true, false, qdef
queue/ (delimiterPosition)	[Specifying the Destination] Position	The following can be selected: prefix, postfix
queue/ (includeDelimiterInMessage)	[Specifying Transfer Options] Include the Delimiter	If the left checkbox is checked, "true" is set.
request - managedTransfer - transferSet - item - destination - file filespace queue		
(EOL)	[Specifying the Destination] Destination Line Ending	The following can be selected: LF, CRLF
(encoding)	[Specifying the Destination] Destination Character Encoding	If "noswaplfnl" is specified, the behavior of character set mapping with EBCDIC LF 0x25 characters will change.
attribute	[Specifying the Destination] Attribute	Multiple specifications can be specified, separated by ";"

Generate transfer request XML file

It is possible to save the transfer request XML in a file in order to check the format without executing the transfer request (Put the transfer request XML to the command queue).

When specifying a transfer definition file, the XML format to be saved differs from the above "Saving transfer parameters" ([file] menu -> [Save Parameter]). In [Save Parameter], the transfer definition file path is specified in this program's original XML attribute "transferTemplateFilePath" in "managedTransfer" tag, whereas in the case of this "Generate transfer request XML file", the actual transfer request XML file that a transfer definition file was expanded is created.

Set the output file name to [Generating Transfer Templates] - "Transfer Template File Path:" and click the "Execute" button.



This "Transfer Template File Path:" is the same as the "-gt transfer_template_file_path" of fteCreateTransfer.

The XML output from this program is the same as the transfer request XML except that it does not have the following "reply" tag for receiving a response from the command queue manager.

Example of request - managedTransfer - replay tag:

```
<reply QMGR="QM_AG1">WMQFTE.57AE0A1320003102</reply>
```

"Transfer definition file" has a method of defining "whole transfer" and "only transfer source file and destination file". "A file that defines "whole transfer" contains a "managedTransfer" tag in its XML definition and the generated transfer request XML file is also generated from the transfer definition file, not "Table 3.2 XML mapping of transfer requests". If "only transfer source file and destination file" is specified, only under request - managedTransfer - transferSet - item hierarchy is generated from the transfer definition file, not "Table 3.2 XML mapping of transfer requests".

The table below is a list of functions that can be executed by selecting menu items.

Table 3.3 List of menu items	
Menu item	function
[File] - [Save Setup]	Save setup parameters
[File] - [Restore Setup]	Restore setup parameters
[File] - [Save Parameter]	Save transfer parameters
[File] - [Restore Parameter]	Restore transfer parameters
[File] - [Exit]	Exit program
[Edit] - [Clear Setup]	Clear setup parameters
[Edit] - [Clear Parameter]	Clear transfer parameters
[Edit] - [Clear Logs]	Clear received XML message on [Response Message] and [File Transfer Log] tab
[Edit] - [Clear All]	Clear Setup parameters, Transfer parameters and received XML message on [Response Message] and [File Transfer Log] tab
[Help] - [About]	Display version information

4. Example of transfer request

Ex. 1: Transfer specifying the source / destination queue manager

If there is no source / destination agent on the connected command Qmgr, it is necessary to specify each queue manager.

Command Sample :

```
fteCreateTransfer -p CDQMGR80 -sa AG801 -da AG901 -dd /home/mft/to
/home/mft/from/test1.txt -de overwrite -sm QMAG801 -dm QMAG901
```

Parameter description :

```
-p : Set of configuration options: CDQMGR80
-sa : Source Agent Name : AG801
-da : Destination Agent Name : AG901
-dd : Destination directory : /home/mft/to
    : Source specification : /home/mft/from/test1.txt
-de : Destination file behavior : overwrite
```

-sm : Source agent qmgr name : QMAG801
-dm : Destination agent qmgr name : QMAG901

*When this command is executed on the destination side (QMAG 901 side), the writing process is called with the following specification.

Queue Manager : QMAG901
Command Queue : SYSTEM.FTE.COMMAND.AG801 *The opposite side (source side) command queue
Object Queue Manager : QMAG801 *The opposite side (source side) queue manager

The transfer request XML must be written to the source agent's command queue.

When opening the queue for put, this test program sets the queue manager name specified in the following field to queueManagerName (object queue manager name) of MQQueueManager.accessQueue.

[Create Transfer]Tab
Agent Specification
Source Agent Qmgr :
or
[Agent Properties]Tab
Source Agent
agentQMgr :

As a result, if the transmission queue is created with the destination queue manager name, the message is transferred from the source queue manager to the object queue manager.

Setup XML: Example1Setup.xml

```
$ cat Example1Setup.xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<initialize>
  <CommandQmgrEntry mode="client">
    <CmdQmgr>QMAG901</CmdQmgr>
    <CmdQueue>SYSTEM.FTE.COMMAND.AG801</CmdQueue>
    <CmdQmgrHost>AG901_host</CmdQmgrHost>
    <CmdQmgrChannel>QMAG901.SVRCONN</CmdQmgrChannel>
    <CmdQmgrPort>1414</CmdQmgrPort>
    <CmdQmgrUserID>mqm</CmdQmgrUserID>
  </CommandQmgrEntry>
  <CoordinationQmgrEntry mode="client">
    <CoordQmgr>CDQMGR80</CoordQmgr>
    <CoordQmgrHost>CD_host</CoordQmgrHost>
```

```

    <CoodQmgrChannel>CDQMGR80.SVRCONN</CoodQmgrChannel>
    <CoodQmgrPort>1414</CoodQmgrPort>
    <CoodQmgrUserID>mqm</CoodQmgrUserID>
  </CoodinationQmgrEntry>
</initialize>
----
```

Parameter XML: Example1.xml

```

----
$ cat Example1.xml
<?xml version="1.0" encoding="UTF-8"?>
<request version="6.00" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" x
si:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent agent="AG801" QMgr="QMAG801"/>
    <destinationAgent agent="AG901" QMgr="QMAG901"/>
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>/home/mft/from/test1.txt</file>
        </source>
        <destination type="directory" exist="overwrite">
          <file>/home/mft/to</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
----
```

Transfer settings :

· Parameters for connection to the command queue manager

Command Queue Manager: QMAG901

Queue: SYSTEM.FTE.COMMAND.AG801

Client Mode: check

Host Name: AG901_host

Channel: QMAG901.SVRCONN

User Id: mqm

· **Parameters for connection to the coordination queue manager**

[File Transfer Log] Tab

Coordination Queue Manager: CDQMGR80

Client Mode: check

Host Name: CD_host

Channel: CDQMGR80.SVRCONN

User Id: mqm

Parameters for transfer :

· **Agent Properties**

[Agent Properties] Tab

Source Agent

agentQMgr: QMAG801

*If "Source Agent:" on the "Create Transfer" tab is not specified at the time of transfer instruction, this will be used instead.

Destination Agent

agentQMgr: QMAG901

*If "Destination Agent:" on the "Create Transfer" tab is not specified at the time of transfer instruction, this will be used instead.

· **Transfer Parameters**

[Create Transfer] Tab

Source Specification: /home/mft/from/test1.txt

Agent Specification

Source Agent: AG801

Destination Agent: AG901

Specifying Transfer Option

Checksum Algorithm: MD5

Specifying the Destination

Destination Type: directory

Destination Directory: /home/mft/to

Destination File Behavior: overwrite *Behavior when a file with the same name exists in the transfer destination

Type of File Transfer: binary *"Binary" or "Text" can be specified.

Specifying the Source

Source File Disposition: leave *Handling of the transfer source file after transfer completion ("leave" or "delete" can be specified)

Transfer Log :

(abridgement)

Ex. 2: Scheduling transfer (Case 1)

Specify the transfer start time, transfer interval, and transfer count.

Command Sample :

```
fteCreateTransfer -de overwrite -p CDQMGR80 -sa AG901 -da AG801 -dd /home/mft/to  
/home/mft/from/test1.txt -ss 2017-04-27T20:17 -tb admin -oi minutes -of 1 -oc 3 -sm  
QMAG901 -dm QMAG801
```

*Transfer three times from 2017-04-27T20:17 every 1 minute.

Parameter description :

- de : Destination file behavior : overwrite
- p : Set of configuration options: CDQMGR80
- sa : Source Agent Name : AG901
- da : Destination Agent Name : AG801
- dd : Destination directory : /home/mft/to
 : Source specification : /home/mft/from/test1.txt
- ss : Schedule start time : 2017-04-27T20:17
- tb : The time base for the scheduled file transfer : admin(the system time and date
 used by the local administrator)
- oi : The interval that the scheduled transfer occurs at : minutes
- of : occurrence frequency : 1
- oc : occurrence count : 3
- sm : Source agent qmgr name : QMAG901
- dm : Destination agent qmgr name : QMAG801

Setup XML: Example2Setup.xml

```
$ cat Example2Setup.xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
<initialize>  
  <CommandQmgrEntry mode="client">  
    <CmdQmgr>QMAG901</CmdQmgr>  
    <CmdQueue>SYSTEM.FTE.COMMAND.AG901</CmdQueue>  
    <CmdQmgrHost>AG901_host</CmdQmgrHost>  
    <CmdQmgrChannel>QMAG901.SVRCONN</CmdQmgrChannel>  
    <CmdQmgrPort>1414</CmdQmgrPort>  
    <CmdQmgrUserID>mqm</CmdQmgrUserID>  
  </CommandQmgrEntry>
```

```

<CoordinationQmgrEntry mode="client">
  <CoodQmgr>CDQMGR80</CoodQmgr>
  <CoodQmgrHost>CD_host</CoodQmgrHost>
  <CoodQmgrChannel>CDQMGR80.SVRCONN</CoodQmgrChannel>
  <CoodQmgrPort>1414</CoodQmgrPort>
  <CoodQmgrUserID>mqm</CoodQmgrUserID>
</CoordinationQmgrEntry>
</initialize>
----
```

Parameter XML: Example2.xml

```

----
$ cat Example2.xml
<?xml version="1.0" encoding="UTF-8"?>
<request xsi:noNamespaceSchemaLocation="FileTransfer.xsd" xmlns:xsi="http://www
w.w3.org/2001/XMLSchema-instance" version="6.00">
  <managedTransfer>
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <schedule>
      <submit timezone="Asia/Tokyo" timebase="admin"> 2017-04-28T18:30</submi
t>
      <repeat>
        <frequency interval="minutes">1</frequency>
        <expireCount>3</expireCount>
      </repeat>
    </schedule>
    <sourceAgent agent="AG901" QMgr="QMAG901"/>
    <destinationAgent agent="AG801" QMgr="QMAG801"/>
    <transferSet>
      <item checksumMethod="MD5" mode="binary">
        <source disposition="leave" recursive="false">
          <file>/home/mft/from/test1.txt</file>
        </source>
        <destination exist="overwrite" type="directory">
          <file>/home/mft/to</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
----
```


Transfer Parameters :

· Parameters for connection to the command queue manager

Command Queue Manager: QMAG901

Queue: SYSTEM.FTE.COMMAND.AG901

Client Mode: check

Host Name: AG901_host

Channel: QMAG901.SVRCONN

User Id: mqm

· Parameters for connection to the coordination queue manager

[File Transfer Log] Tab

Coordination Queue Manager: CDQMGR80

Client Mode: check

Host Name: CD_host

Channel: CDQMGR80.SVRCONN

User Id: mqm

Parameters for transfer :

· Agent Properties

[Agent Properties] Tab

Source Agent

agentQMgr: QMAG901

Destination Agent

agentQMgr: QMAG801

· Transfer Parameters

[Create Transfer] Tab

Source Specification: /home/mft/from/test1.txt

Agent Specification

Source Agent: AG901

Destination Agent: AG801

Scheduling Transfers

Schedule Start Time: 2017-04-28T18:30

Time Base: admin *the system time and date used by the local administrator

Occurrence Interval: minutes

Occurrence Frequency: 1

Occurrence Count: 3

Specifying Transfer Option

Checksum Algorithm: MD5

Specifying the Destination

Destination Type: directory

Destination Directory: /home/mft/to

Destination File Behavior: overwrite

Type of File Transfer: binary

Specifying the Source

Source File Disposition: leave

Transfer Log :

Schedule "submit" is published first.

```
<?xml version="1.0" encoding="UTF-8"?><schedulelog version="6.00" ID="2" xmlns:
xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocatio
n="ScheduleLog.xsd"><originator><hostName>sagx01.makuhari.japan.ibm.com</host
Name><userID>mqm</userID><mqmdUserID>mqm</mqmdUserID></originator><a
ction time="2017-04-29T03:56:06.436Z">submit</action><schedule>
```

*As of MQ 8.0.0.4, since this is output with malformed-xml which starts with "<schedule log> " and is closed with "<schedule>", the following error occurs and it can not parse.

[Fatal Error]: 9: 999: The XML document structure must start and end within the same entity.

Therefore, this program displays it on one line in the "File Transfer Log" tab.

<First transfer>

....

```
<action time="2017-04-29T03:56:36.549Z">started</action>
```

....

```
<action time="2017-04-29T03:56:38.061Z">progress</action>
```

....

```
<action time="2017-04-29T03:56:38.157Z">completed</action>
```

....

*Time is displayed in real time +9 for some reason.

<Second Transfer>

....

```
<action time="2017-04-29T03:57:08.246Z">started</action>
```

```

....
<action time="2017-04-29T03:57:08.558Z">progress</action>
....
<action time="2017-04-29T03:57:08.613Z">completed</action>
....
*"expire" is output before the last transfer.
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<schedulelog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ID="2" versi
on="6.00" xsi:noNamespaceSchemaLocation="ScheduleLog.xsd">
  <originator>
    <hostName>sagx01.makuhari.japan.ibm.com</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <action time="2017-04-29T03:58:08.618Z">expire</action>
  <sourceAgent QMgr="QMAG901" agent="AG901"/>
  <status resultCode="0"/>
</schedulelog>
----

```

<Third (Last) Transfer>

```

----
....
<action time="2017-04-29T03:58:08.702Z">started</action>
....
<action time="2017-04-29T03:58:09.015Z">progress</action>
....
<action time="2017-04-29T03:58:09.067Z">completed</action>
....
----

```

Ex. 3: Scheduling transfer (Case 2)

Specify the transfer start time, transfer interval, and transfer end time.

Command Sample :

```

fteCreateTransfer -de overwrite -p CDQMGR80 -sa AG901 -da AG801 -dd /home/mft/to
/home/mft/from/test1.txt -ss 2017-04-28T20:00 -tb source -oi minutes -of 1 -es 2017-04-
28T20:05 -sm QMAG901 -dm QMAG801

```

*Transfer every 1 minutes from 2017-04-28 20:00 to 2017-04-28 20:05.

Parameter description:

- de : Destination file behavior : overwrite
- p : Set of configuration options: CDQMGR80
- sa : Source Agent Name : AG901
- da : Destination Agent Name : AG801
- dd : Destination directory : /home/mft/to
: Source specification : /home/mft/from/test1.txt
- ss : Schedule start time : 2017-04-28T20:00
- tb : The time base for the scheduled file transfer : source (source agent)
- oi : The interval that the scheduled transfer occurs at : minutes
- se : Schedule end time : 2017-04-28T20:05
- sm : Source agent qmgr name : QMAG901
- dm : Destination agent qmgr name : QMAG801

Setup XML: Example2Setup.xml

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameter XML: Example3.xml

\$ cat Example3.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <schedule>
      <submit timebase="source" timezone="Asia/Tokyo">2017-04-
28T20:20</submit>
      <repeat>
        <frequency interval="minutes">1</frequency>
        <expireTime>2017-04-28T20:25</expireTime>
      </repeat>
    </schedule>
    <sourceAgent QMgr="QMAG901" agent="AG901"/>
    <destinationAgent QMgr="QMAG801" agent="AG801"/>
    <transferSet>
      <item checksumMethod="MD5" mode="binary">
        <source disposition="leave" recursive="false">
          <file>/home/mft/from/test1.txt</file>
```

```

        </source>
        <destination exist="overwrite" type="directory">
            <file>/home/mft/to</file>
        </destination>
    </item>
</transferSet>
</managedTransfer>
</request>
----
```

Transfer Parameters :

- **Parameters for connection to the command queue manager**

*See "Ex. 2: Scheduling Transfer (Case 1)"

- **Parameters for connection to the coordination queue manager**

[File Transfer Log] Tab

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameters for transfer :

- **Agent Properties**

[Agent Properties] Tab

Source Agent

agentQMgr: QMAG901

Destination Agent

agentQMgr: QMAG801

- **Transfer Parameters**

[Create Transfer] Tab

Source Specification: /home/mft/from/test1.txt

Agent Specification

Source Agent: AG9011

Destination Agent: AG801

Scheduling Transfers

Schedule Start Time: 2017-04-28T20:27

Time Base: source *source agent

Occurrence Interval: minutes

Occurrence Frequency: 1

Schedule End Time: 2017-04-28T20:32

Specifying Transfer Option

Checksum Algorithm: MD5

Specifying the Destination

Destination Type: directory

Destination Directory: /home/mft/to

Destination File Behavior: overwrite

Type of File Transfer: binary

Specifying the Source

Source File Disposition: leave

Transfer Log :

Schedule "submit" is published first.

```
<?xml version="1.0" encoding="UTF-8"?><schedulelog version="6.00" ID="6"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ScheduleLog.xsd"><originator><hostName>sagx0
1.makuhari.japan.ibm.com</hostName><userID>mqm</userID><mqmdUserID>mqm
</mqmdUserID></originator><action time="2017-04-
29T05:25:12.066Z">submit</action><schedule>
```

* See "Ex. 2: Scheduling Transfer (Case 1)" for the problem on this format

<First transfer>

....

```
<action time="2017-04-29T05:27:12.143Z">started</action>
```

....

```
<action time="2017-04-29T05:27:12.458Z">progress</action>
```

....

```
<action time="2017-04-29T05:27:12.510Z">completed</action>
```

*Time is displayed in real time +9 for some reason.

<Second Transfer>

....

```
<action time="2017-04-29T05:28:12.596Z">started</action>
```

....

```
<action time="2017-04-29T05:28:12.897Z">progress</action>
```

....

```
<action time="2017-04-29T05:28:12.956Z">completed</action>
```

....

*"expire" is created before the last transfer.

```

<action time="2017-04-29T05:32:14.338Z">expire</action>
....
----

<Last Transfer>
----
....
<action time="2017-04-29T05:32:14.401Z">started</action>
....
<action time="2017-04-29T05:32:14.721Z">progress</action>
....
<action time="2017-04-29T05:32:14.772Z">completed</action>
....
----

```

Ex. 4: Trigger transfer

Transfer is performed when the file specified in the name list exists.

Command Sample :

```

fteCreateTransfer -de overwrite -p CDQMGR80 -sa AG901 -da AG801 -dd /home/mft/to
/home/mft/from/test1.txt -tr file=exist,/home/mft/trg/trg1.txt,/home/mft/trg/trg2.txt -tl
yes -sm QMAG901 -dm QMAG801

```

*If trg1.txt, trg2.txt exists, transfer starts.

Parameter description:

```

-de : Destination file behavior : overwrite
-p : Set of configuration options: CDQMGR80
-sa : Source Agent Name : AG901
-da : Destination Agent Name : AG801
-dd : Destination directory : /home/mft/to
    : Source specification : /home/mft/from/test1.txt
-tr : A condition that must be true for this file transfer to take place.  :
file=exist,/home/mft/trg/trg1.txt,/home/mft/trg/trg2.txt(trg1.txt, trg2.txt must exist)
-tb : The time base for the scheduled file transfer : admin(the system time and date
used by the local administrator)
-tl : Whether trigger failures are written to the transfer log : yes
-sm : Source agent qmgr name : QMAG901
-dm : Destination agent qmgr name : QMAG801

```

Setup XML: Example2Setup.xml

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameter XML: Example4.xml

```
$ cat Example4.xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00" xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QMAG901" agent="AG901"/>
    <destinationAgent QMgr="QMAG801" agent="AG801"/>
    <trigger log="yes">
      <fileExist comparison="=" value="exist">/home/mft/trg/trg1.txt,/home/mft/trg/trg2.txt</fileExist>
    </trigger>
    <transferSet>
      <item checksumMethod="MD5" mode="binary">
        <source disposition="leave" recursive="false">
          <file>/home/mft/from/test1.txt</file>
        </source>
        <destination exist="overwrite" type="directory">
          <file>/home/mft/to</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Transfer Parameters :

- **Parameters for connection to the command queue manager**

*See "Ex. 2: Scheduling Transfer (Case 1)"

- **Parameters for connection to the coordination queue manager**

[File Transfer Log] Tab

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameters for transfer :

· **Agent Properties**
[Agent Properties] Tab
Source Agent
agentQMgr: QMAG901

Destination Agent
agentQMgr: QMAG801

· **Transfer Parameters**
[Create Transfer] Tab
Source Specification: /home/mft/from/test1.txt

Agent Specification
Source Agent: AG9011
Destination Agent: AG801

Triggering Transfers
Condition, Namelist: file=exist,/home/mft/trg/trg1.txt,/home/mft/trg/trg2.txt
*Condition can specify file! = Exist, filesize> = size in addition. Transfer starts when the files specified in Namelist matches the condition.
Create Transfer Log Entries: yes/no *If yes, the transfer log is also created for failed trigger transfers.

Specifying Transfer Option
Checksum Algorithm: MD5

Specifying the Destination
Destination Type: directory
Destination Directory: /home/mft/to
Destination File Behavior: overwrite
Type of File Transfer: binary

Specifying the Source
Source File Disposition: leave

Transfer Log :
<Case where transfer failed because trigger file does not exist>

....
 <action time="2017-05-03T05:39:44.349Z">**started**</action>
....
 <action time="2017-05-03T05:39:44.353Z">**progress**</action>
....
 <action time="2017-05-03T05:39:44.357Z">**completed**</action>
....

```

<status resultCode="42">
  <supplement>BFGCH0064W: A trigger definition file = exist, /home/mft/trg/trg
1.txt, /home/mft/trg/trg2.txt was not successful and the file transfer will not proce
ed.</supplement>
</status>
<transferSet bytesSent="0" startTime="2017-05-03T05:39:44.349Z" total="1"/>
</transaction>
----
```

<Case where transfer was successful>
it is the same as other successful examples

Ex. 5: Transfer when the destination is Queue (Case 1)

Specify the queue, not the file as the destination.

*It is necessary to add "enableQueueInputOutput=true" to agent.properties of the destination agent.

Command Sample :

```

fteCreateTransfer -p CDQMGR80 -sa AG901 -da AG801 -dq MFTTESTQ
/home/mft/from/test1.txt -jn JOB_NAME -cs none -pr 9 -qmp true -qs 10B -w 10 -sm
QMAG901 -dm QMAG801
```

Parameter description :

```

-p : Set of configuration options: CDQMGR80
-sa : Source Agent Name : AG901
-da : Destination Agent Name : AG801
-dq : Destination Queue : MFTTESTQ
    : Source specification : /home/mft/from/test1.txt
-jn : A Job name identifier that is added to the transfer log message : JOB_NAME
-cs : Checksum algorithm : none
-pr : Transfer priority : 9
-qmp : Whether the first message written to the destination queue by the transfer has
IBM MQ message properties set : true
-qs : Size to split the file : 10B
-w : Time to wait for the requested transfer to complete : 10(second) (* Items not in the
transfer XML, the get wait time waiting for a response from the command queue
manager)
-sm : Source agent qmgr name : QMAG901
-dm : Destination agent qmgr name : QMAG801
```

Setup XML: Example2Setup.xml

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameter XML: Example5.xml

```
$ cat Example5.xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00" xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QMAG901" agent="AG901"/>
    <destinationAgent QMgr="QMAG801" agent="AG801"/>
    <transferSet priority="9">
      <item checksumMethod="none" mode="binary">
        <source disposition="leave" recursive="false">
          <file>/home/mft/from/test1.txt</file>
        </source>
        <destination type="queue">
          <queue delimiter="10B" delimiterType="size" setMqProps="true">MFTTE
STQ</queue>
        </destination>
      </item>
    </transferSet>
    <job>
      <name>JOB_NAME</name>
    </job>
  </managedTransfer>
</request>
```

Transfer Parameters :

- Parameters for connection to the command queue manager

*See "Ex. 2: Scheduling Transfer (Case 1)"

- Parameters for connection to the coordination queue manager

[File Transfer Log] Tab

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameters for transfer :

· Agent Properties

[Agent Properties] Tab

Source Agent

agentQMgr: QMAG901

Destination Agent

agentQMgr: QMAG801

· Transfer Parameters

[Create Transfer] Tab

Source Specification: /home/mft/from/test1.txt

Agent Specification

Source Agent: AG9011

Destination Agent: AG801

Specifying Transfer Option

Job Name: JOB_NAME

Checksum Algorithm: none

Transfer Priority: 9

Set Message Properties: true

Message Size: 10B

Configuration Options: Currentry no use *This is equivalent to "-p" of

fteCreateTransfer Pramerter

Timeout: check and 10000(ms)

Specifying the Destination

Destination Type: queue

Destination Queue: MFTTESTQ

Type of File Transfer: binary

Specifying the Source

Source File Disposition: leave

*Enabling Input / Output to Queue

If "enableQueueInputOutput=true" is not added to agent.properties of the destination agent, the following error message occurs and the transfer fails.

<action time="2017-05-03T06:02:07.920Z">progress</action>

....

<status resultCode="1">

<supplement>BFGTR0072E: The transfer failed to complete due to the exception : BFGIO0197E: An attempt to write to a queue was rejected by the destination agent. The agent must have enableQueueInputOutput=true set in the agent.properties file to support transferring to a queue.</supplement>

Confirm the destination message:

The properties shown in the table below are set in the first message.

Table 4.1 Message property of destination message		
Property	Value	Note
usr.WMQFTETransferId	The unique hexadecimal transfer ID.	
usr.WMQFTETransferMode	The type of file transfer: binary mode or text mode.	
usr.WMQFTESourceAgent	The name of the source agent.	
usr.WMQFTEDestinationAgent	The name of the destination agent.	
usr.WMQFTEFileName	The name of the source file.	
usr.WMQFTEFileSize	The size of the source file in bytes.	
usr.WMQFTEFileLastModified	The last modified time of the source file. This value is in units of milliseconds, measured from 00:00:00 UTC, January 1, 1970.	
usr.WMQFTEFileIndex	The index of the current file in the list of files that are being transferred. The first file in the list has index 0.	
usr.WMQFTEMqmdUser	The MQMD user ID of the user that submitted the transfer request.	

Below is an example of returning a value in NameValueData of RFH2.

```
$ mqpgf -qm QMAG801 -q MFTTESTQ -brv
message number: 1
```

....

```
*StrucId[RFH ] Version[2] StrucLength[524] Encoding[273] CodedCharSetId[0]
```

```

Format[          ] Flags[0] NameValueCCSID[1208]
NameValueLength[484]
NameValueData[<usr><WMQFTEMqmdUser>mqm</WMQFTEMqmdUser><WMQF
TEDestinationAgent>AG801</WMQFTEDestinationAgent><WMQFTEFileLastModifi
ed>1492059915000</WMQFTEFileLastModified><WMQFTEFileName>/home/mft/fro
m/test1.txt</WMQFTEFileName><WMQFTESourceAgent>AG901</WMQFTESource
Agent><WMQFTETransferMode>binary</WMQFTETransferMode><WMQFTETransf
erId>414d5120514d4147393031202020202059095ede2351b313</WMQFTETransferId
><WMQFTEFileIndex>0</WMQFTEFileIndex><WMQFTEFileSize>75</WMQFTEFil
eSize></usr>  ]

```

....

*The message is divided into 10 bytes in the Destination queue, the same Group Id is assigned, MsgSeqNumber is 1 origin sequential number., MsgFlags is set MQMF_MSG_IN_GROUP=0x00000008 in all messages. In addition, the last message MQMF_LAST_MSG_IN_GROUP is ORed with 0x00000010 (8+16=24).

The file size of the transfer source is 75 bytes.

```

ls -l /home/ mft/from/test1.txt
-rw-r--r--    1 mqm      mqm              75  Apr 12 20:05 /home/mft/from/test1.txt

```

The number of messages after the transfer is 8. Because the message size was specified to 10B, it is divided into 8 messages.

```

$ mqpcfg que -qm QMAG801 -q MFTTESTQ CURDEPTH
1: QUEUE(MFTTESTQ) TYPE(QLOCAL) CURDEPTH(8)

```

The following is an excerpt from the situation where all messages are browsed from the transfer destination queue and confirmed.

It is getting with MQGMO_PROPERTIES_IN_HANDLE option.

```

>mqpgf -qm QMAG801 -q MFTTESTQ MQGMO_PROPERTIES_IN_HANDLE
MQGMO_VERSION_4 -brv -r
message number: 1

```

....

```

GroupId[0x414D5120514D414738303120202020205908620220003603]
MsgSeqNumber[1] Offset[0] MsgFlags[8] OriginalLength[-1]

```

****Message properties****

```

WMQFTEMqmdUser : 'mqm'

```

WMQFTEDestinationAgent : 'AG801'
WMQFTEFileLastModified : '1492059915000'
WMQFTEFileName : '/home/mft/from/test1.txt'
WMQFTESourceAgent : 'AG901'
WMQFTETransferMode : 'binary'
WMQFTETransferId : '414d5120514d4147393031202020202059095ede2351b313'
WMQFTEFileIndex : '0'
WMQFTEFileSize : '75'
MQINQMP faild : CompCd=02 ReasonCd=2471

data length: 10
00000000: B9E7 B7D7 2031 0A2D 7277 'ヶ邱ヲ 1.-rw '

message number: 2
....
GroupId[0x414D5120514D414738303120202020205908620220003603]
MsgSeqNumber[2] Offset[0] MsgFlags[8] OriginalLength[-1]

....
data length: 10
00000000: 2D72 2D2D 722D 2D20 2020 '-r--r-- '

message number: 3
....
GroupId[0x414D5120514D414738303120202020205908620220003603]
MsgSeqNumber[3] Offset[0] MsgFlags[8] OriginalLength[-1]

....
data length: 10
00000000: 3120 6D71 6D20 2020 2020 '1 mqm '

message number: 4
....
GroupId[0x414D5120514D414738303120202020205908620220003603]
MsgSeqNumber[4] Offset[0] MsgFlags[8] OriginalLength[-1]

....
data length: 10
00000000: 206D 716D 2020 2020 2020 ' mqm '

message number: 5
....
GroupId[0x414D5120514D414738303120202020205908620220003603]
MsgSeqNumber[5] Offset[0] MsgFlags[8] OriginalLength[-1]

....
data length: 10
00000000: 2020 2020 2020 3020 2034 ' 0 4 '

```

message number: 6
....
GroupId[0x414D5120514D414738303120202020205908620220003603]
MsgSeqNumber[6] Offset[0] MsgFlags[8] OriginalLength[-1]
....
data length: 10
00000000: B7EE 2031 32C6 FC20 2032          'キ. 12ニ. 2          '

message number: 7
....
GroupId[0x414D5120514D414738303120202020205908620220003603]
MsgSeqNumber[7] Offset[0] MsgFlags[8] OriginalLength[-1]
....
data length: 10
00000000: 303A 3035 2074 6573 7431          '0:05 test1          '

message number: 8
....
GroupId[0x414D5120514D414738303120202020205908620220003603]
MsgSeqNumber[8] Offset[0] MsgFlags[24] OriginalLength[-1]
....
data length: 5
00000000: 2E74 7874 0A          '.txt.          '
....
----
```

Transfer Log :

```

----
The queue manager and queue of the transfer destination are displayed into "pro
gress" as shown below.
The specified job name also.
....
  <action time="2017-05-09T05:00:35.065Z">started</action>
....
  <job>
    <name>JOB_NAME</name>
  </job>
</transaction>
....
  <action time="2017-05-09T05:00:35.343Z">progress</action>
....
    <destination type="queue">
      <queue delimiter="10B" delimiterType="size" groupId="414d5120514d41473
830312020202020591040a920002c03" messageCount="8" messageLength="10" persi
```



```

stent="true" setMqProps="true">MFTTESTQ@QMAG801</queue>
  <checksum method="none"/>
</destination>

<job>
  <name>JOB_NAME</name>
</job>
</transaction>
....
<action time="2017-05-09T05:00:35.394Z">completed</action>
<job>
  <name>JOB_NAME</name>
</job>
----

```

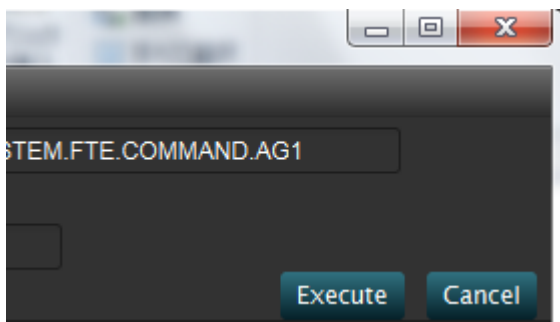
***Receive acknowledgment message for completed transfer**

By checking the following check box(Timeout), you can receive an acknowledgment message for transfer completion.

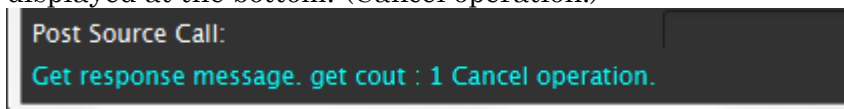
Specifying Transfer Options

Timeout(checkbox): Check and specify the maximum wait time in ms.

If you do not specify a wait time by checking, continue waiting indefinitely.
To forcibly end this wait, click the Cancel button below.



When you click the Cancel button, a message indicating that cancel was accepted is displayed at the bottom. (Cancel operation.)



When receiving normally, two responses are received as shown below, and the end of the receiving thread is displayed at the bottom.(response receive thread end.)

```
Get response message. get cout : 2 response receive thread end.
```

```

-2    ACK    The request has been received but is pending completion.
0     Success    The command was successful

```

Return code	Short name	Description
0	Success	The command was successful
1	Command unsuccessful	The command ended unsuccessfully.
2	Command timed out	The agent did not reply with the status of the command within a specified timeout. By default, this timeout is unlimited for managed call and transfer commands. For example, when you specify the <code>-w</code> parameter with the <code>fteCreateTransfer</code> command. By default, this timeout is 5 seconds for other commands.
3	Acknowledgement timed out	The agent did not acknowledge receipt of the command within a specified timeout. By default, this timeout is 5 seconds.

58

Table 4.3 Intermediate reply codes		
Return code	Short name	Description
-2	ACK	The request has been received but is pending completion.
-3	PROGRESS	The request is for a number of files and some are still pending completion.

Ex. 6: Transfer when the destination is Queue (Case 2)

Specify the delimiter for dividing the message.

Command Sample :

```
fteCreateTransfer -p CDQMGR80 -sa AG901 -da AG801 -dq MFTTESTQ
/home/mft/from/test2.txt -qi -dqdb x0d,x0a -dqdp postfix -sm QMAG901 -dm QMAG801
```

* Divide the file by CRLF (0x0d, 0x0a) and include the delimiter CRLF (0x0d, 0x0a) in the output message.

Parameter description :

-p : Set of configuration options: CDQMGR80
 -sa : Source Agent Name : AG901
 -da : Destination Agent Name : AG801
 -dq : Destination Queue : MFTTESTQ
 : Source specification : /home/mft/from/test2.txt
 -qi : Include the delimiter that is used to split the file into multiple messages in the messages.
 -dqdb : The hexadecimal delimiter to use when splitting a binary file into multiple messages. : x0d,x0a(CRLF)
 -dqdp : The expected position of destination text and binary delimiters when splitting files. : postfix
 -sm : Source agent qmgr name : QMAG901
 -dm : Destination agent qmgr name : QMAG801

Setup XML: Example2Setup.xml

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameter XML: Example6.xml

```

$ cat Example6.xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00" xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QMAG901" agent="AG901"/>
    <destinationAgent QMgr="QMAG901" agent="AG801"/>
    <transferSet>
      <item checksumMethod="MD5" mode="binary">
        <source disposition="leave" recursive="false">
          <file>/home/mft/from/test2.txt</file>
        </source>
        <destination type="queue">
          <queue delimiter="x0d,x0a" delimiterType="binary" includeDelimiterInMessage="true">MFTTESTQ</queue>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
----
```

Transfer Parameters :

- **Parameters for connection to the command queue manager**

*See "Ex. 2: Scheduling Transfer (Case 1)"

- **Parameters for connection to the coordination queue manager**

[File Transfer Log] Tab

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameters for transfer :

- **Agent Properties**

[Agent Properties] Tab

Source Agent

agentQMGr: QMAG901

Destination Agent

agentQMGr: QMAG801

· **Transfer Parameters**

[Create Transfer] Tab

Source Specification: /home/mft/from/test2.txt

Agent Specification

Source Agent: AG9011

Destination Agent: AG801

Specifying Transfer Option

Checksum Algorithm: MD5

Include the Delimiter: check

Specifying the Destination

Destination Type: queue

Destination Queue: MFTTESTQ

Hexadecimal Delimiter: x0d,x0a

Type of File Transfer: binary

Specifying the Source

Source File Disposition: leave

Confirm the source message:

\$ cat test2.txt

1234567890

abcdefghij

ABCDEFGHIJ

\$ od -x test2.txt

0000000 3132 3334 3536 3738 3930 0d0a 6162 6364

0000020 6566 6768 696a 0d0a 4142 4344 4546 4748

0000040 494a 0d0a

* Three records are included, separated by CRLF.

Confirm the destination message:

\$ mqpcf ques -qm QMAG801 -q MFTTESTQ CURDEPTH

1: QUEUE(MFTTESTQ) TYPE(QUEUE) CURDEPTH(3)

\$ mqpgf -qm QMAG801 -q MFTTESTQ -brv -r

message number: 1

....

GroupId[0x414D5120514D4147383031202020202059118D6320002E03]
MsgSeqNumber[1] Offset[0] MsgFlags[8] OriginalLength[-1]

data length: 12

00000000: 3132 3334 3536 3738 3930 0D0A '1234567890.. '

message number: 2

....

GroupId[0x414D5120514D4147383031202020202059118D6320002E03]
MsgSeqNumber[2] Offset[0] MsgFlags[8] OriginalLength[-1]

data length: 12

00000000: 6162 6364 6566 6768 696A 0D0A 'abcdefghij.. '

message number: 3

....T

data length: 12

00000000: 4142 4344 4546 4748 494A 0D0A 'ABCDEFGHJIJ.. '

no message available : MFTTESTQ CompCd=02 ReasonCd=2033

Transfer Log :
(abridgement)

Ex. 7: Invoke the program

Invoke a specific script before and after transfer respectively at source / destination agent. It also calls the user exit program and confirms the delivery of the specified metadata.

Command Sample :

```
fteCreateTransfer -de overwrite -p CDQMGR80 -sa AG901 -da AG801 -dd /home/mft/to  
/home/mft/from/* -md meta1=data1,meta2=data2,meta3=data3 -presrc  
/home/mft/script/presrc.ksh -predst /home/mft/script/predst.ksh -postsrc  
/home/mft/script/postsrc.ksh -postdst /home/mft/script/postdst.ksh -sm QMAG901 -dm  
QMAG801
```

Parameter description :

-de : Destination file behavior : overwrite
-p : Set of configuration options: CDQMGR80
-sa : Source Agent Name : AG901

-da : Destination Agent Name : AG801
 -dd : Destination directory : /home/mft/to
 : Source specification : /home/mft/from/* (Use an asterisk)
 -md : Specifies the user-defined metadata that is passed to the exit points run by the agent. : meta1=data1,meta2=data2,meta3=data3
 -presrc : Specifies a program to invoke at the source agent before the transfer starts. : /home/mft/script/presrc.ksh
 -predst : Specifies a program to invoke at the destination agent before the transfer starts. : /home/mft/script/predst.ksh
 -postsrc : Specifies a program to invoke at the source agent after the transfer has completed. : /home/mft/script/postsrc.ksh
 -postdst : Specifies a program to invoke at the destination agent after the transfer has completed. : /home/mft/script/postdst.ksh
 -sm : Source agent qmgr name : QMAG901
 -dm : Destination agent qmgr name : QMAG801

Sample script and its settings :

The four scripts only output system time to each naming file.

```
$ cat postsrc.ksh
```

```
#!/bin/ksh
```

```
/home/mft/script/currenttime > /home/mft/script/postsrc.out
```

```
exit 0
```

It is necessary to set the path where the script / program to be invoked is set in 'commandPath' of agent.properties in each agent.

```
$ cat /var/mqm/mqft/config/CDQMGR80/agents/AG901/agent.properties
```

```
....
```

```
commandPath=/bin:/home/mft/script:
```

```
.....
```

Sample userExit and its settings :

Invoke the source transfer start, source transfer end, destination transfer start, destination transfer end user exit routine.

It is necessary to set the user exit routine name to be invoked in agent.properties in each agent.

```
$ cat /var/mqm/mqft/config/CDQMGR80/agents/AG901/agent.properties
```

```
....
sourceTransferStartExitClasses=SampleStartExit
sourceTransferEndExitClasses=SampleEndExit
destinationTransferStartExitClasses=SampleDstStartExit
destinationTransferEndExitClasses=SampleDstEndExit
....
```

Place the class file of the user exit routine in the following location of each agent.

```
ls -l /var/mqm/mqft/config/CDQMGR80/agents/AG901/exits
total 30
-rw-r--r--  1 mqm      mqm      2736  May 10  19:57 SampleDstEndExit.class
-rw-r--r--  1 mqm      mqm      3308  May 11  19:16 SampleDstStartExit.class
-rw-r--r--  1 mqm      mqm      2710  May 10  19:57 SampleEndExit.class
-rw-r--r--  1 mqm      mqm      3480  May 11  19:16 SampleStartExit.class
```

Each user exit program outputs information to each file or log.
Environment metadata and Transfer metadata are passed to all user exits.
File metadata is passed only to the source transfer start exit.

SourceTransferStartExit: (Write to /tmp/Metadata.txt)
Environment metadata, Transfer metadata and File metadata

DestinationTransferStartExit: (Write to /tmp/MetadataD.txt)
Environment metadata, Transfer metadata and File List

SourceTransferEndExit: (Write to output0.log)
Environment metadata, Transfer metadata and File List

DestinationTransferEndExit: (Write to output0.log)
Environment metadata, Transfer metadata and File List

Setup XML: Example2Setup.xml

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameter XML: Example7.xml

```
----
$ cat Example7.xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00" x
si:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
```



```

<originator>
  <hostName>AG901_host</hostName>
  <userID>mqm</userID>
</originator>
<sourceAgent QMgr="QMAG901" agent="AG901"/>
<destinationAgent QMgr="QMAG801" agent="AG801"/>
<transferSet>
  <metaDataSet>
    <metaData key="meta1">data1</metaData>
    <metaData key="meta2">data2</metaData>
    <metaData key="meta3">data3</metaData>
  </metaDataSet>
  <preSourceCall>
    <command name="/home/mft/script/presrc.ksh" type="executable"/>
  </preSourceCall>
  <postSourceCall>
    <command name="/home/mft/script/postsrc.ksh" type="executable"/>
  </postSourceCall>
  <preDestinationCall>
    <command name="/home/mft/script/predst.ksh" type="executable"/>
  </preDestinationCall>
  <postDestinationCall>
    <command name="/home/mft/script/postdst.ksh" type="executable"/>
  </postDestinationCall>
  <item checksumMethod="MD5" mode="binary">
    <source disposition="leave" recursive="false">
      <file>/home/mft/from/*</file>
    </source>
    <destination exist="overwrite" type="directory">
      <file>/home/mft/to</file>
    </destination>
  </item>
</transferSet>
</managedTransfer>
</request>
----
```

Transfer Parameters :

· **Parameters for connection to the command queue manager**

*See "Ex. 2: Scheduling Transfer (Case 1)"

· **Parameters for connection to the coordination queue manager**

[File Transfer Log] Tab

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameters for transfer :

· **Agent Properties**

[Agent Properties] Tab

Source Agent

agentQMgr: QMAG901

Destination Agent

agentQMgr: QMAG801

· **Transfer Parameters**

[Create Transfer] Tab

Source Specification: /home/mft/from/*

Agent Specification

Source Agent: AG9011

Destination Agent: AG801

Specifying Transfer Option

User-defined Metadata: meta1=data1,meta2=data2,meta3=data3

Checksum Algorithm: MD5

Include the Delimiter: check

Invoking Programs

Pre Source Call: executable:/home/mft/script/presrc.ksh

Pre Destination Call: executable:/home/mft/script/predst.ksh

Post Source Call: executable:/home/mft/script/postsrc.ksh

Post Destination Call: executable:/home/mft/script/postdst.ksh

Specifying the Destination

Destination Type: directory

Destination Directory: /home/mft/to

Destination File Behavior: overwrite

Type of File Transfer: binary

Specifying the Source

Source File Disposition: leave

Confirm transfer source message :

\$ ls -l /home/mft/from

total 6

-rw-r--r-- 1 mqm mqm 75 Apr 12 20:05 test1.txt

```

-rw-r--r--  1 mqm      mqm          36  May  8  21:24 test2.txt
-rw-r--r--  1 mqm      mqm          33  May  8  21:15 test3.txt
-----

```

Confirm transfer destination message :

```

-----
$ ls -l mft/to
total 24
-rw-r--r--  1 mqm      mqm          75  May 12 20:39 test1.txt
-rw-r--r--  1 mqm      mqm          36  May 12 20:39 test2.txt
-rw-r--r--  1 mqm      mqm          33  May 12 20:39 test3.txt
-----

```

Verify script output :

```

-----
$ cat mft/script/presrc.out
2017/05/12 20:39:49:895
$ cat mft/script/postsrc.out
2017/05/12 20:39:51: 43
$ cat mft/script/predst.out
2017/05/12 20:39:46:729
$ cat mft/script/postdst.out
2017/05/12 20:39:47:118
-----

```

*It is invoked in the order of pre source / pre destination / post destination / post source.

Confirm that the metadata was passed to the user exit :

[Source agent output0.log]

```

-----
$ cat output0.log
....
[12/05/2017 20:39:49:859 GMT-09:00] 0000003c StdOut      I
SourceTransferStartExit Start!
[12/05/2017 20:39:49:861 GMT-09:00] 0000003c StdOut      I   VM Name :
21823@sagx01
[12/05/2017 20:39:49:862 GMT-09:00] 0000003c StdOut      I   PID : 21823
[12/05/2017 20:39:49:863 GMT-09:00] 0000003c StdOut      I
SenderEventListener414d5120514d4147393031202020202059167ec623cb6ab3
[12/05/2017 20:39:49:865 GMT-09:00] 0000003c StdOut      I
SourceTransferStartExit Stop!

```

```

[12/05/2017 20:39:51:005 GMT-09:00] 00000042 StdOut      I
SourceTransferEndExit Start!
[12/05/2017 20:39:51:007 GMT-09:00] 00000042 StdOut      I   VM Name :
21823@sagx01
[12/05/2017 20:39:51:008 GMT-09:00] 00000042 StdOut      I   PID : 21823
[12/05/2017 20:39:51:008 GMT-09:00] 00000042 StdOut      I
SenderEnd414d5120514d4147393031202020202059167ec623cb6ab3
*From the following, Environment metadata, Transfer metadata and File list output
of the Source transfer exit.
[12/05/2017 20:39:51:009 GMT-09:00] 00000042 StdOut      I   Environment
Meta Data: {com.ibm.wmqfte.ConfigurationDirectory=/opt/mqm/v90/mqft,
com.ibm.wmqfte.AgentVersion=9.0.0.0,
com.ibm.wmqfte.ProductDirectory=/var/mqm/mqft/config/CDQMGR80/agents/AG901}
[12/05/2017 20:39:51:010 GMT-09:00] 00000042 StdOut      I   Transfer Meta
Data: {meta1=data1, meta2=data2, meta3=data3,
com.ibm.wmqfte.SourceAgent=AG901, com.ibm.wmqfte.DestinationAgent=AG801,
com.ibm.wmqfte.MqmdUser=mqm, com.ibm.wmqfte.OriginatingUser=mqm,
com.ibm.wmqfte.OriginatingHost=sagx01.makuhari.japan.ibm.com,
com.ibm.wmqfte.TransferId=414d5120514d4147393031202020202059167ec623cb6ab3
, com.ibm.wmqfte.Priority=0}
[12/05/2017 20:39:51:011 GMT-09:00] 00000042 StdOut      I   Source agent:
AG901
[12/05/2017 20:39:51:011 GMT-09:00] 00000042 StdOut      I   Destination agent:
AG801
[12/05/2017 20:39:51:012 GMT-09:00] 00000042 StdOut      I   File list:
[12/05/2017 20:39:51:012 GMT-09:00] 00000042 StdOut      I   Source file spec:
/home/mft/from/test1.txt, Destination file spec: /home/mft/to/test1.txt
[12/05/2017 20:39:51:013 GMT-09:00] 00000042 StdOut      I   Source file spec:
/home/mft/from/test2.txt, Destination file spec: /home/mft/to/test2.txt
[12/05/2017 20:39:51:013 GMT-09:00] 00000042 StdOut      I   Source file spec:
/home/mft/from/test3.txt, Destination file spec: /home/mft/to/test3.txt
[12/05/2017 20:39:51:014 GMT-09:00] 00000042 StdOut      I
SourceTransferEndExit Stop!

```

[Destination agent output0.log]

\$ cat output0.log

....

```

[12/05/2017 20:39:46:655 JST] 00000032 StdOut      I
DestinationTransferStartExit Start!
[12/05/2017 20:39:46:655 JST] 00000032 StdOut      I   VM Name :
7012592@villa
[12/05/2017 20:39:46:656 JST] 00000032 StdOut      I   PID : 7012592
[12/05/2017 20:39:46:656 JST] 00000032 StdOut      I

```

```

ReceiverEventListener414d5120514d4147393031202020202059167ec623cb6ab3
[12/05/2017 20:39:46:673 JST] 00000032 StdOut      I
DestinationTransferStartExit Stop!
[12/05/2017 20:39:47:054 JST] 00000036 StdOut      I
DestinationTransferEndExit Start!
[12/05/2017 20:39:47:054 JST] 00000036 StdOut      I   VM Name :
7012592@villa
[12/05/2017 20:39:47:055 JST] 00000036 StdOut      I   PID : 7012592
[12/05/2017 20:39:47:055 JST] 00000036 StdOut      I
TransferReceiver[414d5120514d4147393031202020202059167ec623cb6ab3]
*From the following, Environment metadata, Transfer metadata and File list output
of the Destination transfer exit.
[12/05/2017 20:39:47:056 JST] 00000036 StdOut      I   Environment Meta Data:
{com.ibm.wmqfte.AgentVersion=8.0.0.0,
com.ibm.wmqfte.ConfigurationDirectory=/usr/mqm-mq8000gm/usr/mqm/mqft,
com.ibm.wmqfte.ProductDirectory=/var/mqm/mqft/config/CDQMGR80/agents/AG801}
[12/05/2017 20:39:47:058 JST] 00000036 StdOut      I   Transfer Meta Data:
{meta1=data1, meta2=data2, meta3=data3, com.ibm.wmqfte.SourceAgent=AG901,
com.ibm.wmqfte.DestinationAgent=AG801, com.ibm.wmqfte.MqmdUser=mqm,
com.ibm.wmqfte.OriginatingUser=mqm,
com.ibm.wmqfte.OriginatingHost=sagx01.makuhari.japan.ibm.com,
com.ibm.wmqfte.TransferId=414d5120514d4147393031202020202059167ec623cb6ab3
, com.ibm.wmqfte.Priority=0}
[12/05/2017 20:39:47:059 JST] 00000036 StdOut      I   Source agent:
AG901
[12/05/2017 20:39:47:059 JST] 00000036 StdOut      I   Destination agent:
AG801
[12/05/2017 20:39:47:060 JST] 00000036 StdOut      I   File list:
[12/05/2017 20:39:47:060 JST] 00000036 StdOut      I   Source file spec:
/home/mft/from/test1.txt, Destination file spec: /home/mft/to/test1.txt
[12/05/2017 20:39:47:061 JST] 00000036 StdOut      I   Source file spec:
/home/mft/from/test2.txt, Destination file spec: /home/mft/to/test2.txt
[12/05/2017 20:39:47:062 JST] 00000036 StdOut      I   Source file spec:
/home/mft/from/test3.txt, Destination file spec: /home/mft/to/test3.txt
[12/05/2017 20:39:47:062 JST] 00000036 StdOut      I
DestinationTransferEndExit Stop!

```

[SourceTransferStartExit outputs]

Environment metadata, Transfer metadata and File metadata outputs from
SourceTransferStartExit.

```
$ cat /tmp/Metadata.txt
```

```

SourceTransferStartExit Start
SourceAgentName : AG901
DestinationAgentName : AG801
EnvironmentMetaData :
  key: com.ibm.wmqfte.ConfigurationDirectory, value: /opt/mqm/v90/mqft
  key: com.ibm.wmqfte.AgentVersion, value: 9.0.0.0
  key: com.ibm.wmqfte.ProductDirectory, value:
/var/mqm/mqft/config/CDQMGR80/agents/AG901
TransferMetaData :
  key: meta1, value: data1
  key: meta2, value: data2
  key: meta3, value: data3
  key: com.ibm.wmqfte.SourceAgent, value: AG901
  key: com.ibm.wmqfte.DestinationAgent, value: AG801
  key: com.ibm.wmqfte.MqmdUser, value: mqm
  key: com.ibm.wmqfte.OriginatingUser, value: mqm
  key: com.ibm.wmqfte.OriginatingHost, value: sagx01.makuhari.japan.ibm.com
  key: com.ibm.wmqfte.TransferId, value:
414d5120514d4147393031202020202059167ec623cb6ab3
  key: com.ibm.wmqfte.Priority, value: 0
FileMetaData :
/home/mft/from/test1.txt
/home/mft/to/test1.txt
  key: com.ibm.wmqfte.SourceFileDisposition, value: leave
  key: com.ibm.wmqfte.ChecksumMethod, value: MD5
  key: com.ibm.wmqfte.FileConversion, value: binary
  key: com.ibm.wmqfte.FileType, value: file
/home/mft/from/test2.txt
/home/mft/to/test2.txt
  key: com.ibm.wmqfte.SourceFileDisposition, value: leave
  key: com.ibm.wmqfte.ChecksumMethod, value: MD5
  key: com.ibm.wmqfte.FileConversion, value: binary
  key: com.ibm.wmqfte.FileType, value: file
/home/mft/from/test3.txt
/home/mft/to/test3.txt
  key: com.ibm.wmqfte.SourceFileDisposition, value: leave
  key: com.ibm.wmqfte.ChecksumMethod, value: MD5
  key: com.ibm.wmqfte.FileConversion, value: binary
  key: com.ibm.wmqfte.FileType, value: file
----
```

[DestinationTransferStartExit outputs]

Environment metadata, Transfer metadata, File list outputs from DestinationTransferStartExit.

```

----
$ cat /tmp/MetadataD.txt
DestinationTransferStartExit Start
SourceAgentName : AG901
DestinationAgentName : AG801
EnvironmentMetaData :
  key: com.ibm.wmqfte.AgentVersion, value: 8.0.0.0
  key: com.ibm.wmqfte.ConfigurationDirectory, value: /usr/mqm-
mq8000gm/usr/mqm/mqft
  key: com.ibm.wmqfte.ProductDirectory, value:
/var/mqm/mqft/config/CDQMGR80/agents/AG801
TransferMetaData :
  key: meta1, value: data1
  key: meta2, value: data2
  key: meta3, value: data3
  key: com.ibm.wmqfte.SourceAgent, value: AG901
  key: com.ibm.wmqfte.DestinationAgent, value: AG801
  key: com.ibm.wmqfte.MqmdUser, value: mqm
  key: com.ibm.wmqfte.OriginatingUser, value: mqm
  key: com.ibm.wmqfte.OriginatingHost, value: sagx01.makuhari.japan.ibm.com
  key: com.ibm.wmqfte.TransferId, value:
414d5120514d4147393031202020202059167ec623cb6ab3
  key: com.ibm.wmqfte.Priority, value: 0
FileList :
/home/mft/to/test1.txt
/home/mft/to/test2.txt
/home/mft/to/test3.txt
----

```

Transfer Log :

*Transfer metadata provided by user, script and user exit invoked are also displayed.

```

----
....
  <action time="2017-05-13T05:39:49.823Z">started</action>
....
  <metaDataSet>
    <metaData key="meta1">data1</metaData>
    <metaData key="meta2">data2</metaData>
    <metaData key="meta3">data3</metaData>
....
  <preSourceCall>
    <command name="/home/mft/script/presrc.ksh" retryCount="0" retryWait="0"
successRC="0" type="executable"/>

```

```

    </preSourceCall>
    <postSourceCall>
      <command name="/home/mft/script/postsrc.ksh" retryCount="0" retryWait="0"
successRC="0" type="executable"/>
    </postSourceCall>
    <preDestinationCall>
      <command name="/home/mft/script/predst.ksh" retryCount="0" retryWait="0"
successRC="0" type="executable"/>
    </preDestinationCall>
    <postDestinationCall>
      <command name="/home/mft/script/postdst.ksh" retryCount="0" retryWait="0"
successRC="0" type="executable"/>
    </postDestinationCall>
  </transferSet>
</transaction>

....
<action time="2017-05-13T05:39:50.842Z">progress</action>
....

....
<action time="2017-05-13T05:39:51.048Z">completed</action>
....
  <startExits>
    <exit name="class SampleStartExit">
      <status resultCode="proceed">
        <supplement/>
      </status>
    </exit>
  </startExits>
  <endExits>
    <exit name="class SampleEndExit">
      <status>
        <supplement>Done</supplement>
      </status>
    </exit>
  </endExits>
....
  <startExits>
    <exit name="class SampleDstStartExit">
      <status resultCode="proceed">
        <supplement/>
      </status>
    </exit>
  </startExits>

```



```

<endExits>
  <exit name="class SampleDstEndExit">
    <status>
      <supplement>Done</supplement>
    </status>
  </exit>
</endExits>
....
  <metaData key="meta1">data1</metaData>
  <metaData key="meta2">data2</metaData>
  <metaData key="meta3">data3</metaData>
....
  <preSourceCall>
    <command name="/home/mft/script/presrc.ksh" retryCount="0" retryWait="0"
successRC="0" type="executable"/>
....
  <postSourceCall>
    <command name="/home/mft/script/postsrc.ksh" retryCount="0" retryWait="0"
successRC="0" type="executable"/>
....
  <preDestinationCall>
    <command name="/home/mft/script/predst.ksh" retryCount="0" retryWait="0"
successRC="0" type="executable"/>
....
  <postDestinationCall>
    <command name="/home/mft/script/postdst.ksh" retryCount="0" retryWait="0"
successRC="0" type="executable"/>
....
----

```

Ex. 8: Using transfer definition file (Case 1) (Specify only source and destination)

Specify the transfer definition file that specifies only the source and destination.

Command Sample :

```
fteCreateTransfer -de overwrite -p CDQMGR80 -sa AG901 -da AG801 -td
TransferDefinition1.xml -sm QMAG901 -dm QMAG801
```

Parameter description:

-de : Destination file behavior : overwrite
-p : Set of configuration options: CDQMGR80

```
-sa : Source Agent Name : AG901
-da : Destination Agent Name : AG801
-td : Transfer definition file path: TransferDefinitionFile1.xml
-sm : Source agent qmgr name : QMAG901
-dm : Destination agent qmgr name : QMAG801
```

*In the parameter XML("Save Parameter"), the transfer definition file path is set to this program's the unique attribute 'transferDefinitionFile' of the ManagedTransfer tag. Transfer XML is created with the transfer definition file expanded.

Setup XML: Example2Setup.xml

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameter XML: Example8.xml

```
----
$ cat Example8.xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer
transferDefinitionFile="/home/mft/TransferDefinitionFile1.xml">
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QMAG901" agent="AG901"/>
    <destinationAgent QMgr="QMAG801" agent="AG801"/>
    <transferSet/>
  </managedTransfer>
</request>
----
```

Transfer Definition File: TransferDefinitionFile1.xml

*Example of specifying two transfers

```
----
$ cat TransferDefinitionFile1.xml
<?xml version="1.0" encoding="UTF-8"?>
<transferSpecifications xsi:noNamespaceSchemaLocation="FileTransfer.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <item mode="text" checksumMethod="MD5">
    <source disposition="leave" recursive="false">
      <file>/home/mft/from/test1.txt</file>
    </source>
  </item>
</transferSpecifications>
```

```

    </source>
    <destination exist="overwrite" type="directory">
      <file>/home/mft/to</file>
    </destination>
  </item>
  <item mode="text" checksumMethod="MD5">
    <source disposition="leave" recursive="false">
      <file>/home/mft/from/test2.txt</file>
    </source>
    <destination exist="overwrite" type="directory">
      <file>/home/mft/to</file>
    </destination>
  </item>
</transferSpecifications>
----
```

Transfer XML: Example8out.xml

This is the format when Transfer Template File Path is specified and saved. When using the Transfer definition file, Transfer XML is different from the format saved by "File" → "Save Parameter". The XML actually written to the Command queue is a format in which the 'reply' tag is added to the XML file outputted by setting the output file name to "Transfer Template File Path:" of [Generating Transfer Templates]. The 'reply' tag is required to receive a response from the command queue manager. (See "Ex. 3. How to create a transfer" - "Generate transfer request XML file")

```

$ cat Example8out.xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QMAG901" agent="AG901"/>
    <destinationAgent QMgr="QMAG801" agent="AG801"/>
    <transferSet>
      <item checksumMethod="MD5" mode="text">
        <source disposition="leave" recursive="false">
          <file>/home/mqm/mft/from/test1.txt</file>
        </source>
        <destination exist="overwrite" type="directory">
          <file>/home/mqm/mft/to</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

```

</item>
<item checksumMethod="MD5" mode="text">
  <source disposition="leave" recursive="false">
    <file>/home/mqm/mft/from/test2.txt</file>
  </source>
  <destination exist="overwrite" type="directory">
    <file>/home/mqm/mft/to</file>
  </destination>
</item>
</transferSet>
</managedTransfer>
</request>
----
```

Transfer Parameters :

- **Parameters for connection to the command queue manager**

*See "Ex. 2: Scheduling Transfer (Case 1)"

- **Parameters for connection to the coordination queue manager**

[File Transfer Log] Tab

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameters for transfer :

- **Agent Properties**

[Agent Properties] Tab

Source Agent

agentQMGr: QMAG901

Destination Agent

agentQMGr: QMAG801

- **Transfer Parameters**

[Create Transfer] Tab

Agent Specification

Source Agent: AG9011

Destination Agent: AG801

Specifying the Destination

Destination Type: Transfer Definition File

Transfer Definition File: /home/mft/TransferDefinitionFile1.xml

Transfer Log :

*Two file transfer statuses are generated in "progress".

```
----
....
    <action time="2017-05-16T05:03:14.899Z">started</action>
....
    <action time="2017-05-16T05:03:17.099Z">progress</action>
....
    <transferSet bytesSent="111" index="0" size="2" startTime="2017-05-
16T05:03:14.901Z" total="2">
        <item mode="text">
            <source disposition="leave" type="file">
                <file EOL="LF" encoding="eucJP-open" last-modified="2017-04-
13T05:05:15.000Z" size="75">/home/mft/from/test1.txt</file>
                <checksum
method="MD5">595f5e5ab58c86c70e7a315d76e23301</checksum>
                </source>
                <destination exist="overwrite" type="file">
                    <file EOL="LF" encoding="IBM-943C" last-modified="2017-05-
15T11:03:11.000Z" size="75">/home/mft/to/test1.txt</file>
                    <checksum
method="MD5">c34bf54934da575ab0c65b4088836639</checksum>
                    </destination>
                    <status resultCode="0"/>
                </item>
                <item mode="text">
                    <source disposition="leave" type="file">
                        <file EOL="LF" encoding="eucJP-open" last-modified="2017-05-
09T06:24:41.000Z" size="36">/home/mft/from/test2.txt</file>
                        <checksum
method="MD5">449507ca589bac9a73a8f0310b4db7c3</checksum>
                        </source>
                        <destination exist="overwrite" type="file">
                            <file EOL="LF" encoding="IBM-943C" last-modified="2017-05-
15T11:03:11.000Z" size="33">/home/mft/to/test2.txt</file>
                            <checksum method="MD5">5ca74738bfe4c13986afaf906c23d6df</checksum>
                            </destination>
                            <status resultCode="0"/>
                        </item>
                    </transferSet>
....
    <action time="2017-05-16T05:03:17.220Z">completed</action>
....
----
```

Ex. 9: Using transfer definition file (Case 2) (Specify all required information)

Specify all required information in the Transfer definition file.

Command Sample :

```
fteCreateTransfer -p CDQMGR80 -td /home/mft/TransferDefinitionFile2.xml
```

Parameter description:

-p : Set of configuration options: CDQMGR80

-td : Transfer definition file path: TransferDefinitionFile1.xml

*In the parameter XML("Save Parameter"), the transfer definition file path is set to this program's the unique attribute 'transferDefinitionFile' of the ManagedTransfer tag. Transfer XML is created with the transfer definition file expanded.

Setup XML: Example2Setup.xml

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameter XML: Example9.xml

```
$ cat Example9.xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer
transferDefinitionFile="/home/mft/TransferDefinitionFile2.xml">
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent/>
    <destinationAgent/>
    <transferSet/>
  </managedTransfer>
</request>
```

Transfer Definition File: TransferDefinitionFile2.xml

*Example of specifying separate transfer destinations, directory and queue, for two

items.

```
$ cat TransferDefinitionFile2.xml
<?xml version="1.0" encoding="UTF-8"?>
<request version="3.00" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QMAG901" agent="AG901"/>
    <destinationAgent QMgr="QMAG801" agent="AG801"/>
    <transferSet>
      <item mode="text" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>/home/mft/from/test1.txt</file>
        </source>
        <destination type="directory" exist="overwrite">
          <file>/home/mft/to</file>
        </destination>
      </item>
      <item checksumMethod="none" mode="binary">
        <source disposition="leave" recursive="false">
          <file>/home/mft/from/test2.txt</file>
        </source>
        <destination type="queue">
          <queue>MFTTESTQ</queue>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Transfer XML: Example9out.xml

This is the format when Transfer Template File Path is specified and saved.

When using the Transfer definition file, Transfer XML is different from the format saved by "File" → "Save Parameter". The XML actually written to the Command queue is a format in which the 'reply' tag is added to the XML file outputted by setting the output file name to "Transfer Template File Path:" of [Generating Transfer Templates]. The 'reply' tag is required to receive a response from the command queue manager. (See "Ex. 3. How to create a transfer" - "Generate transfer request XML file") When specifying all necessary information for the transfer definition file (having the

managedTransfer tag), it is the same as the transfer definition file TransferDefinitionFile2.xml.

Other parameters are ignored even if they are specified.

```
$ cat Example9out.xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QMAG901" agent="AG901"/>
    <destinationAgent QMgr="QMAG801" agent="AG801"/>
    <transferSet>
      <item checksumMethod="MD5" mode="text">
        <source disposition="leave" recursive="false">
          <file>/home/mft/from/test1.txt</file>
        </source>
        <destination exist="overwrite" type="directory">
          <file>/home/mft/to</file>
        </destination>
      </item>
      <item checksumMethod="none" mode="binary">
        <source disposition="leave" recursive="false">
          <file>/home/mft/from/test2.txt</file>
        </source>
        <destination type="queue">
          <queue>MFTTESTQ</queue>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Transfer Parameters :

- **Parameters for connection to the command queue manager**

*See "Ex. 2: Scheduling Transfer (Case 1)"

- **Parameters for connection to the coordination queue manager**

[File Transfer Log] Tab

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameters for transfer :

· Agent Properties
[Agent Properties] Tab
Source Agent
(unspecified)

Destination Agent
(unspecified)

· Transfer Parameters
[Create Transfer] Tab

Agent Specification
(unspecified)

Specifying the Destination

Destination Type: Transfer Definition File
Transfer Definition File: /home/mft/TransferDefinitionFile2.xml

Transfer Log :

*Two file transfer statuses are generated in "progress".

....

<action time="2017-05-16T06:11:31.081Z">**started**</action>

....

<action time="2017-05-16T06:11:31.729Z">**progress**</action>

....

<transferSet bytesSent="111" index="0" size="2" startTime="2017-05-16T06:11:31.081Z" total="2">

<item mode="text">

<source disposition="leave" type="file">

<file EOL="LF" encoding="eucJP-open" last-modified="2017-04-13T05:05:15.000Z" size="75">/home/mft/from/test1.txt</file>

<checksum

method="MD5">595f5e5ab58c86c70e7a315d76e23301</checksum>

</source>

<destination exist="overwrite" type="file">

<file EOL="LF" encoding="IBM-943C" last-modified="2017-05-15T12:11:26.000Z" size="75">/home/mft/to/test1.txt</file>

<checksum

method="MD5">c34bf54934da575ab0c65b4088836639</checksum>

</destination>

```

        <status resultCode="0"/>
    </item>
    <item mode="binary">
        <source disposition="leave" type="file">
            <file last-modified="2017-05-09T06:24:41.000Z"
size="36">/home/mft/from/test2.txt</file>
            <checksum method="none"/>
        </source>
        <destination type="queue">
            <queue messageCount="1"
messageId="414d5120514d414738303120202020205919884720002b05"
persistent="true">MFTTESTQ@QMAG801</queue>
            <checksum method="none"/>
        </destination>
        <status resultCode="0"/>
    </item>
</transferSet>
....
<action time="2017-05-16T06:11:31.803Z">completed</action>
....
----
```

Ex.10: Specify code conversion

Specify to convert the code at the time of transfer.

Command Sample :

```
fteCreateTransfer -p CDQMGR80 -sa AG901 -da AG801 -de overwrite -df
/home/mft/to/test3.txt /home/mft/from/test3.txt -t text -sce IBM-943 -dce UTF-8 -dle LF
-sm QMAG901 -dm QMAG801
```

Parameter description:

```
-p : Set of configuration options: CDQMGR80
-sa : Source Agent Name : AG901
-da : Destination Agent Name : AG801
-de : Destination file behavior : overwrite
-df : Destination file : /home/mft/from/test3.txt
: Source specification : /home/mft/from/test3.txt
-t : The type of file transfer : text
-sce : Source Character Encoding : IBM-943(ShiftJIS)
-dce : Destination Character Encoding : UTF-8
-dle : Destination End-Of-Line Characters : LF(0x0a)
-sm : Source agent qmgr name : QMAG901
```

-dm : Destination agent qmgr name : QMAG801

Setup XML: Example2Setup.xml

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameter XML: Example10.xml

```
----
$ cat Example10.xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QMAG901" agent="AG901"/>
    <destinationAgent QMgr="QMAG801" agent="AG801"/>
    <transferSet>
      <item checksumMethod="MD5" mode="text">
        <source disposition="leave" recursive="false">
          <file encoding="IBM-943">/home/mft/from/test3.txt</file>
        </source>
        <destination exist="overwrite" type="file">
          <file EOL="LF" encoding="UTF-8">/home/mft/to/test3.txt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
----
```

Transfer Parameters :

· **Parameters for connection to the command queue manager**

*See "Ex. 2: Scheduling Transfer (Case 1)"

· **Parameters for connection to the coordination queue manager**

[File Transfer Log] Tab

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameters for transfer :

・ Agent Properties
[Agent Properties] Tab
Source Agent
agentQMgr: QMAG901

Destination Agent
agentQMgr: QMAG801

・ Transfer Parameters
[Create Transfer] タブ
Source Specification: /home/mft/from/test3.txt

Specifying Transfer Option
Checksum Algorithm: MD5

Specifying the Destination
Destination Type: file
Destination File: /home/mft/to/test3.txt
Destination File Behavior: overwrite
Type of File Transfer: text
Destination Character Encoding: UTF-8
Destination Line Ending: LF

Specifying the Source
Source File Disposition: leave
Source Character Encoding: IBM-943

Confirm transfer source message :

\$ od -x mft/from/test3.txt
0000000 82a0 82a2 82a4 0d0a 8260 8261 8262 0d0a

*IBM-943(ShiftJIS), return code : CRLF(0x0d, 0x0a)

Confirm transfer destination message :

\$ od -x mft/to/test3.txt
0000000 e381 82e3 8184 e381 860a efbc a1ef bca2
0000020 efbc a30a

*UTF-8, return code : LF(0x0a)

Transfer Log :
(abridgement)

Ex.11: Recursively transfer files in subdirectories

Also transfer all files that exist in subdirectories.

Command Sample :

```
fteCreateTransfer -p CDQMGR80 -sa AG901 -da AG801 -dq MFTTESTQ -dqp false  
/home/mft/rec/* -r -t text -sce EUC-JP -dce IBM-943 -dqdt "AA*" -mquserid mqm -  
mqpassword mqm -sm QMAG901 -dm QMAG801
```

Parameter description:

- p : Set of configuration options: CDQMGR80
- sa : Source Agent Name : AG901
- da : Destination Agent Name : AG801
- dq : Destination Queue : MFTTESTQ
- dqp : persistent : false(non persistent)
- : Source specification : /home/mft/rec/* (Wildcards need to be used)
- r : Recursively transfer files in subdirectories when source_specification contains wildcard characters.:
- t : The type of file transfer : text
- sce : Source Character Encoding : EUC-JP
- dce : Destination Character Encoding : IBM-943(ShiftJIS)
- dqdt : The Java™ regular expression to use when splitting a text file into multiple messages. : "AA*"
- mquserid : The user ID to authenticate with the command queue manager. : mqm
- mqpassword : The password to authenticate with the command queue manager. : mqm
- sm : Source agent qmgr name : QMAG901
- dm : Destination agent qmgr name : QMAG801

*- mquserid / - mqpassword was added from the MFT of MQ 80, it is set to MQCSP for CONNAUTH. In parameter XML("Save Parameter"), it is set to the this program unique attribute mqUserId / mqPassword of the ManagedTransfer tag.

*The string length can be matched by regular expressions is limited to 5 characters by default in the destination agent. To change its size, specify the maximum size with maxDelimiterMatchLength property in agent.properties.

Ex)

maxDelimiterMatchLength=20

Setup XML: Example2Setup.xml

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameter XML: Example11.xml

\$ cat Example11.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer mqUserId="mqm" mqPassword="mqm">
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QMAG901" agent="AG901"/>
    <destinationAgent QMgr="QMAG801" agent="AG801"/>
    <transferSet>
      <item checksumMethod="MD5" mode="text">
        <source disposition="leave" recursive="true">
          <file encoding="EUC-JP">/home/mft/rec/*</file>
        </source>
        <destination type="queue">
          <queue delimiter="AA*" delimiterType="text" encoding="IBM-943"
persistent="false">MFTTESTQ</queue>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Transfer Parameters :

· **Parameters for connection to the command queue manager**

*See "Ex. 2: Scheduling Transfer (Case 1)"

· **Parameters for connection to the coordination queue manager**

[File Transfer Log] Tab

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameters for transfer :

· Agent Properties

[Agent Properties] Tab

Source Agent

agentQMgr: QMAG901

Destination Agent

agentQMgr: QMAG801

· Transfer Parameters

[Create Transfer] Tab

Source Specification: /home/mft/rec/*

Agent Specification

Source Agent: AG901

Destination Agent: AG801

Specifying Transfer Option

Checksum Algorithm: MD5

Specifying the Destination

Destination Type: queue

Destination Queue: MFTTESTQ

Persistent: false

Pattern: AA*

Type of File Transfer: text

Destination Character Encoding: IBM-943

Parameters for Security

MQ UserId: mqm

MQ Password: mqm

Specifying the Source

Source File Disposition: leave

Recursively transfer files in subdirectories: check

Source Character Encoding: EUC-JP

Confirm transfer source message(EUC-JP) :

\$ cat testr1.txt

あAAAいいAAA う う う AAA

\$ od -x testr1.txt

```
00000000 a4a2 4141 41a4 a4a4 a441 4141 a4a6 a4a6
00000200 a4a6 4141 4100
```

```
$ cat child/testr2.txt
ええええAAAおおおおAAAかかかかかかAAA
```

```
$ od -x child/testr2.txt
00000000 a4a8 a4a8 a4a8 a4a8 4141 41a4 aaa4 aaa4
00000200 aaa4 aaa4 aa41 4141 a4ab a4ab a4ab a4ab
00000400 a4ab a4ab 4141 4100
----
```

Confirm transfer destination message(IBM-943) :

*The order of message transfer is irregular.

```
$ mqpqgf -qm QMAG801 -q MFTTESTQ -brv -r
message number: 1
```

```
...
CodedCharSetId[943] Format[MQSTR    ] Priority[0] Persistence[0]
```

....

data length: 8

```
00000000:  82A6 82A6 82A6 82A6                                'ええええ'
```

message number: 2

....

```
CodedCharSetId[943] Format[MQSTR    ] Priority[0] Persistence[0]
```

....

data length: 10

```
00000000:  82A8 82A8 82A8 82A8 82A8                                'おおおおお'
```

message number: 3

....

```
CodedCharSetId[943] Format[MQSTR    ] Priority[0] Persistence[0]
```

....

data length: 2

```
00000000:  82A0                                'あ'
```

message number: 4

....

```
CodedCharSetId[943] Format[MQSTR    ] Priority[0] Persistence[0]
```

....

data length: 4

```
00000000:  82A2 82A2                                'い い'
```



```

message number: 5
....
CodedCharSetId[943] Format[MQSTR    ] Priority[0] Persistence[0]
....
data length: 12
00000000:  82A9 82A9 82A9 82A9 82A9 82A9          'かゝかゝかゝかゝかゝ'

```

```

message number: 6
....
CodedCharSetId[943] Format[MQSTR    ] Priority[0]
.....
data length: 6
00000000:  82A4 82A4 82A4          'う う う'

```

```

no message available : MFTTESTQ CompCd=02 ReasonCd=2033
----
```

Transfer Log :

*Two file transfer situations searched recursively for "progress" are generated.

```

----
....
    <action time="2017-05-18T05:03:17.373Z">started</action>
....
    <action time="2017-05-18T05:03:17.880Z">progress</action>
....
    <transferSet bytesSent="60" index="0" size="2" startTime="2017-05-
18T05:03:17.373Z" total="2">
        <item mode="text">
            <source disposition="leave" type="file">
                <file EOL="LF" encoding="EUC-JP" last-modified="2017-05-
18T03:52:34.000Z" size="39">/home/mft/rec/child/testr2.txt</file>
                <checksum method="MD5">818a807cfeec031b4b55dc6f9c1923ca</checksum>
            </source>
            <destination type="queue">
                <queue delimiter="AA*" delimiterPosition="postfix" delimiterType="text"
groupId="414d5120514d41473830312020202020591b901b20003103"
includeDelimiterInMessage="false" messageCount="3" messageLength="1048576"
persistent="false">MFTTESTQ@QMAG801</queue>
                <checksum
method="MD5">0838b5881600032a7306a1c5cd301be4</checksum>
                </destination>
                <status resultCode="0"/>
            </item>

```

```

    <item mode="text">
      <source disposition="leave" type="file">
        <file EOL="LF" encoding="EUC-JP" last-modified="2017-05-
18T03:52:23.000Z" size="21">/home/mft/rec/testr1.txt</file>
        <checksum
method="MD5">ca88fd41b48731f001c0a9ad4e38509c</checksum>
      </source>
      <destination type="queue">
        <queue delimiter="AA*" delimiterPosition="postfix" delimiterType="text"
groupId="414d5120514d41473830312020202020591b901b20003106"
includeDelimiterInMessage="false" messageCount="3" messageLength="1048576"
persistent="false">MFTTESTQ@QMAG801</queue>
        <checksum
method="MD5">0a3983433b82bb29df16c6bbb348f8ad</checksum>
      </destination>
      <status resultCode="0"/>
    </item>
  </transferSet>
....
  <action time="2017-05-18T05:03:17.944Z">completed</action>
....
----
```

Ex.12: Handle group messages when the source specification is queue (Case 1)

Messages on the source queue are grouped by group ID and transferred as one file.

*It is necessary to add 'enableQueueInputOutput=true' to agent.properties of source agent.

Command Sample :

```
fteCreateTransfer -p CDQMGR80 -sa AG901 -da AG801 -de overwrite -sq -df
/home/mft/test.txt MFTTESTQ -t text -sqgi -sqwt 30 -sqdt '¥r¥n' -sqdp prefix -sm
QMAG901 -dm QMAG801
```

Parameter description:

```
-p : Set of configuration options: CDQMGR80
-sa : Source Agent Name : AG901
-da : Destination Agent Name : AG801
-de : Destination file behavior : overwrite
-sq : The source of a transfer is a queue. :
```

-df : Destination file : /home/mft/test.txt
 : Source specification : MFTTESTQ
 -t : The type of file transfer : text
 -sqgi : The messages are grouped by IBM MQ group ID. :
 -sqwt : The time to wait for the arrival of a message (if -sqgi is specified then the time to wait until the group message appears completely) : 30(second)
 -sqdt : A sequence of text to insert as the delimiter when appending multiple messages to a text file.: ¥r¥n(CRLF)
 -sqdp : The position of insertion of source text and binary delimiters. : prefix(The start of each message)
 -sm : Source agent qmgr name : QMAG901
 -dm : Destination agent qmgr name : QMAG801

*Even if there is no message in the source queue, the transfer succeeds on the transfer log, and a 0 byte output file is created.

*Both source and destination can not be specified as queue. ("-sq" and "-dq" can not be specified at the same time)

Setup XML: Example2Setup.xml

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameter XML: Example12.xml

```

$ cat Example12.xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QMAG901" agent="AG901"/>
    <destinationAgent QMgr="QMAG801" agent="AG801"/>
    <transferSet>
      <item checksumMethod="MD5" mode="text">
        <source>
          <queue delimiter="¥r¥n" delimiterPosition="prefix" delimiterType="text"
groupId="{GROUPID}" useGroups="true" waitTime="30">MFTTESTQ</queue>
        </source>
        <destination exist="overwrite" type="file">
          <file>/home/mft/to/test.txt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

```
        </item>
      </transferSet>
    </managedTransfer>
  </request>
  ----
```

Transfer Parameters :

- **Parameters for connection to the command queue manager**

*See "Ex. 2: Scheduling Transfer (Case 1)"

- **Parameters for connection to the coordination queue manager**

[File Transfer Log] Tab

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameters for transfer :

- **Agent Properties**

[Agent Properties] Tab

Source Agent

agentQMgr: QMAG901

Destination Agent

agentQMgr: QMAG801

- **Transfer Parameters**

[Create Transfer] Tab

Source Specification: MFTTESTQ

Agent Specification

Source Agent: AG901

Destination Agent: AG801

Specifying Transfer Option

Checksum Algorithm: MD5

Specifying the Destination

Destination Type: file

Destination file: /home/mft/to/test.txt

Destination Behavior: overwrite

Type of File Transfer: text

Specifying the Source

The source of a transfer is a queue: check

The message are grouped by WMQ group ID: check

Text Delimiter: ¥r¥n

Position of source text and binary delimiters: prefix

Wait time: 30 *(by second)

Write a group message to the source queue :

*Because wait time is specified for 30 seconds, you can also write to queue after instructing transfer with "Execute" button.

```
mqpgf -qm QMAG901 -q MFTTESTQ -m "test3" -gi "mftgroup1" -ms 3
MQMD_VERSION_2 MQMF_MSG_IN_GROUP MQMF_LAST_MSG_IN_GROUP
mqpgf -qm QMAG901 -q MFTTESTQ -m "test1" -gi "mftgroup1" -ms 1
MQMD_VERSION_2 MQMF_MSG_IN_GROUP
mqpgf -qm QMAG901 -q MFTTESTQ -m "test2" -gi "mftgroup1" -ms 2
MQMD_VERSION_2 MQMF_MSG_IN_GROUP
```

*When using the group ID, the arrival order on the queue does not need to be in the order of MsgSeqNumber. The transfer destination file is created by combining them in the order of MsgSeqNumber.

Confirm transfer destination message :

```
$ cat mft/to/test.txt
```

```
test1
test2
test3
```

```
$ od -x mft/to/test.txt
00000000 0a74 6573 7431 0a74 6573 7432 0a74 6573
00000020 7433
```

*In this test, only 0x0A(¥n) was appended to the beginning of each message, 0x0D(¥r) was not appended.

Transfer Log :

(abridgement)

Ex.13: Handle group messages when the source specification is queue (Case 2)

Messages on the source queue are grouped by group ID and transferred as one file.

*It is necessary to add 'enableQueueInputOutput=true' to agent.properties of source agent.

Command Sample :

```
fteCreateTransfer -p CDQMGR80 -sa AG901 -da AG801 -de overwrite -sq -df
/home/mft/to/test.txt MFTTESTQ -t binary -sqgi -sqwt 30 -sqdb x0d,x0a -sqdp postfix -
sm QMAG901 -dm QMAG801
```

Parameter description:

-p : Set of configuration options: CDQMGR80
-sa : Source Agent Name : AG901
-da : Destination Agent Name : AG801
-de : Destination file behavior : overwrite
-sq : The source of a transfer is a queue. :
-df : Destination file : /home/mft/test.txt
: Source specification : MFTTESTQ
-t : The type of file transfer : binary
-sqgi : The messages are grouped by IBM MQ group ID. :
-sqwt : The time to wait for the arrival of a message (if -sqgi is specified then the time to wait until the group message appears completely) : 30(second)
-sqdb : One or more byte values to insert as the delimiter when appending multiple messages to a binary file. : x0d,x0a(CRLF)
-sqdp : The position of insertion of source text and binary delimiters. : postfix (The end of each message)
-sm : Source agent qmgr name : QMAG901
-dm : Destination agent qmgr name : QMAG801

*Even if there is no message in the source queue, the transfer succeeds on the transfer log, and a 0 byte output file is created.

*Both source and destination can not be specified as queue. ("-sq" and "-dq" can not be specified at the same time)

Setup XML: Example2Setup.xml

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameter XML: Example13.xml

```
$ cat Example13.xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```

<request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="6.00"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>AG901_host</hostName>
      <userID>mqm</userID>
    </originator>
    <sourceAgent QMgr="QMAG901" agent="AG901"/>
    <destinationAgent QMgr="QMAG801" agent="AG801"/>
    <transferSet>
      <item checksumMethod="MD5" mode="binary">
        <source>
          <queue delimiter="x0d,x0a" delimiterPosition="postfix"
delimiterType="binary" groupId="{GROUPID}" useGroups="true"
waitTime="30">MFTTESTQ</queue>
        </source>
        <destination exist="overwrite" type="file">
          <file>/home/mft/to/test.txt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

Transfer Parameters :

- **Parameters for connection to the command queue manager**

*See "Ex. 2: Scheduling Transfer (Case 1)"

- **Parameters for connection to the coordination queue manager**

[File Transfer Log] Tab

*See "Ex. 2: Scheduling Transfer (Case 1)"

Parameters for transfer :

- **Agent Properties**

[Agent Properties] Tab

Source Agent

agentQMgr: QMAG901

Destination Agent

agentQMgr: QMAG801

- **Transfer Parameters**

[Create Transfer] Tab

Source Specification: MFTTESTQ

Agent Specification

Source Agent: AG901

Destination Agent: AG801

Specifying Transfer Option

Checksum Algorithm: MD5

Specifying the Destination

Destination Type: file

Destination file: /home/mft/to/test.txt

Destination Behavior: overwrite

Type of File Transfer: binary

Specifying the Source

The source of a transfer is a queue: check

The message are grouped by WMQ group ID: check

Hexadecimal Delimiter: x0d,x0a

Position of source text and binary delimiters: postfix

Wait time: 30 *(by second)

Write a group message to the source queue :

*Because wait time is specified for 30 seconds, you can also write to queue after instructing transfer with "Execute" button.

```
mqpqgf -qm QMAG901 -q MFTTESTQ -m "test3" -gi "mftgroup1" -ms 3
MQMD_VERSION_2 MQMF_MSG_IN_GROUP MQMF_LAST_MSG_IN_GROUP
mqpqgf -qm QMAG901 -q MFTTESTQ -m "test1" -gi "mftgroup1" -ms 1
MQMD_VERSION_2 MQMF_MSG_IN_GROUP
mqpqgf -qm QMAG901 -q MFTTESTQ -m "test2" -gi "mftgroup1" -ms 2
MQMD_VERSION_2 MQMF_MSG_IN_GROUP
```

*When using the group ID, the arrival order on the queue does not need to be in the order of MsgSeqNumber. The transfer destination file is created by combining them in the order of MsgSeqNumber.

Confirm transfer destination message :

```
$ cat mft/to/test.txt
test1
```


test2

test3

```
$ od -x mft/to/test.txt
```

```
00000000  7465 7374 310d 0a74 6573 7432 0d0a 7465
```

```
00000020  7374 330d 0a00
```

*0x0D, 0x0A(CRLF) is appended to the end of each message.

Transfer Log :

(abridgement)

Conclusion

If you find any defects in this program, or if you have any questions and requests about this program, please contact us.

**S.N.Software Inc. - <https://www.pulsarintegration.com>
e-mail: support@pulsarintegration.com**