File Ordering MQMFT Exit

Table of Contents

Change Log

Date	Version	Author	Comments
10 June 2010	1.00	Steve Parsons sparsons@us.ibm.com	Initial version
28 June 2010	1.01	Steve Parsons sparsons@us.ibm.com	Changed behavior when keys not in metadata so that transfer completes without any special ordering. Updated document accordingly and variously.
31 May 2013	2.00	Steve Parsons sparsons@us.ibm.com	Redesigned and added several features: First and Last positioning, sorting by Name and Path (as well as by the original parameters of Size and LastModified).
25 Sep 2014	2.10	Steve Parsons sparsons@us.ibm.com	Implemented fixes and increased the supported combos of parameters. Tested on both MQ 7.5.0.3 and MQ 8.0.0.0.
30 Sep 2014	2.20	Steve Parsons sparsons@us.ibm.com	Changed name of exit artifacts from FileListOrderingFteExit to FileOrderingMqmftExit. Qualified the names of the parameters with “mqmft.fileordering”. Implemented accommodations for message-to-file transfers.
01 July 2016	3.0	Steve Parsons sparsons@us.ibm.com	Public release to github.

About this Asset

File Ordering MQMFT Exit is an MQ Managed File Transfer (MQMFT) exit. In a transfer involving multiple files, MQMFT by default will transfer each of the files in a transfer set item in ascending order by path name. This exit allows the user to dynamically specify the order in which MQMFT transfers the files, based on the source files' names, paths, sizes, and last modified timestamps. Optionally, the user can also use a positioning capability to specify input files that should be transferred first and/or last, regardless of the sorting instructions; for example, a user might wish a summary file with an extension of ".sum" to arrive before companion files that contain the details. This exit is based on requirements from customers.

This document explains the exit and how to deploy and use it. Accompanying this document is an archive containing the Java classfiles for the exit.

The exit applies the specified sorting and positioning to both file-to-file and file-to-message transfers, expressly upon attributes of the source files. The exit has no effect with message-to-file transfers.

Version 3 of this exit has been built and tested with MQ 7.5.0.3, and 8.0.0.4 on Windows 7, RHEL 6.7, and AIX 7.1.0.0. The exit is expected to work on all supported MQMFT distributed platforms.

Many thanks to Rob Simons, Bobbee Broderick , Martin Phillips, Pete Verdon, and Enrique Relucio for their feedback on this exit and the requirements it addresses. Many thanks to Adrian Preston, John Hosie, Sally Whittingham, and Mark Taylor for their support in making it publicly available. Many thanks to Jared Spencer for providing test machines. And many thanks to my wife for her understanding in the times that this project absorbed me.

Direct questions and comments to the author via email: sparsons@us.ibm.com.

Materials Needed

- MQMFT with source and destination agents set up and working

Prerequisite Skills and Knowledge

- Familiar with MQ
- Able to set up MQMFT source and destination agents and to run transfers between them

Deployment

1) Stop the MQMFT source agent.
2) Put the accompanying jarfile (FileOrderingMqmftExit_v3.0.jar) in the exits directory of the source agent.
3) Update the agent.properties file of the source agent with this entry (one line): sourceTransferStartExitClasses=com.ibm.mqmft.exit.fileordering.FileOrderingSourceStartMqmftExit
4) Start the source agent.

Use

This section provides some examples of using the exit. We will use two MQMFT agents in our example. The Windows agent is named WINNIE and the Linux agent is named LINNIE. The exit is deployed on both agents. We will use a template in MQMFT's MQ Explorer plugin to submit transfer requests. We will view MQMFT log publications in rfhutil (from Support Pac IH03: http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24000637).

We will begin with a simple file-to-file transfer. On Linux we have the following five files in /XferDirs/ExitTest/src2:

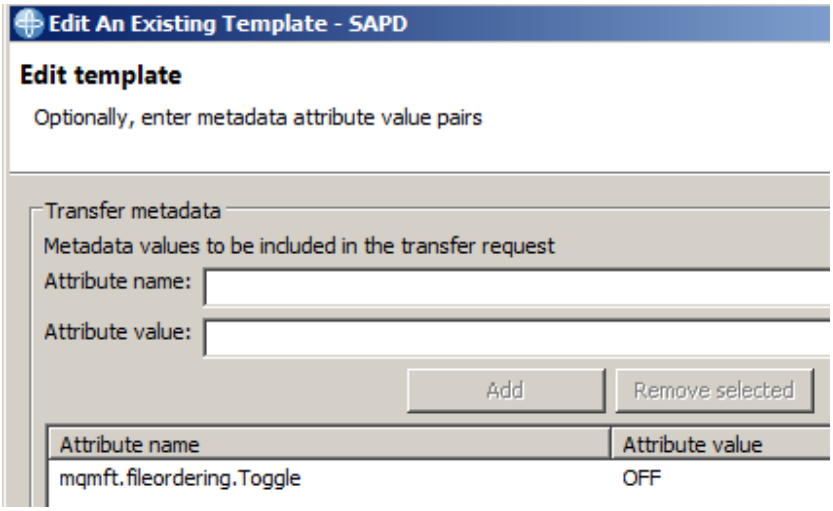
```
[sparsons@oc2518075255 src2]$ pwd
/XferDirs/ExitTest/src2
[sparsons@oc2518075255 src2]$ ls -l
total 5120
-rw-rw-r--. 1 sparsons sparsons 1048576 Jun 29 22:44 aOneMeg.dat
-rw-rw-r--. 1 sparsons sparsons 1048576 Jun 29 22:44 bOneMeg.dat
-rw-rw-r--. 1 sparsons sparsons 1048576 Jun 29 22:44 cOneMeg.dat
-rw-rw-r--. 1 sparsons sparsons 1048576 Jun 29 22:44 dOneMeg.dat
-rw-rw-r--. 1 sparsons sparsons 1048576 Jun 29 22:44 eOneMeg.dat
[sparsons@oc2518075255 src2]$
```

We will move those five files in one transfer from LINNIE to WINNIE:

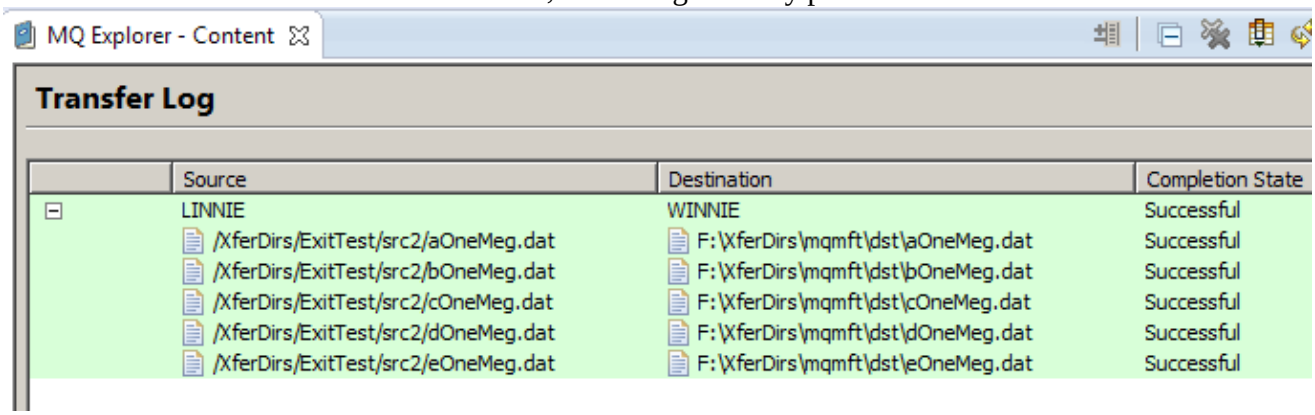
Transfer Templates

Name	Source	Destination	Scheduled Start
⊟ SAPD	LINNIE	WINNIE	
	/XferDirs/ExitTest/src2/*	F:/XferDirs/mqmft/dst	

Before submitting the transfer request, let's add some parameters for the exit. First, let's set mqmft.fileordering.Toggle to 'OFF':

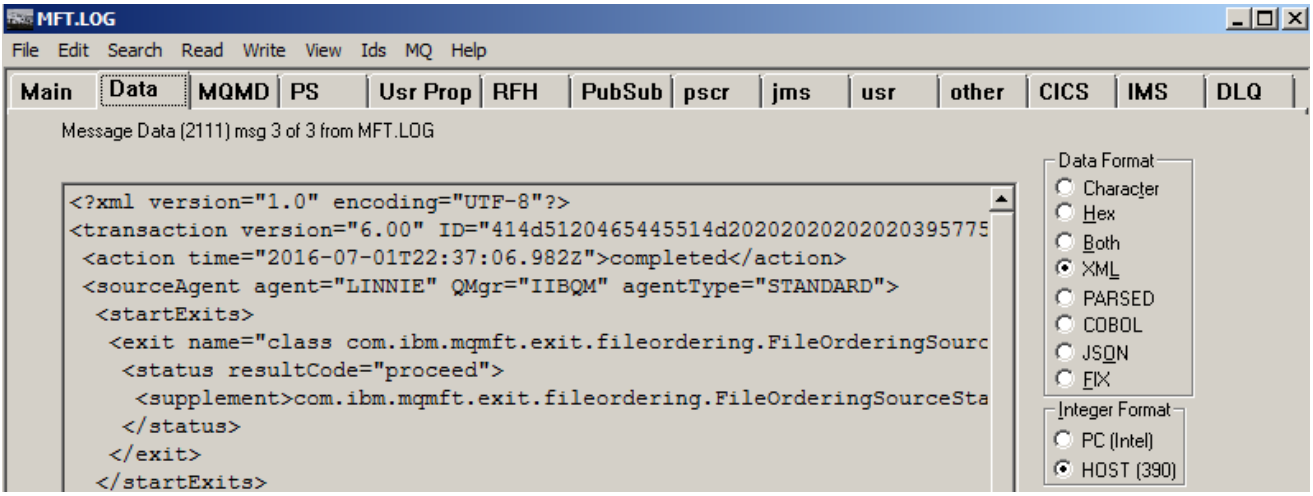


We submit the transfer request, and it succeeds. In MQ Explorer's Transfer Log, we see that MQMFT transferred the five files in the default order, ascending order by path:

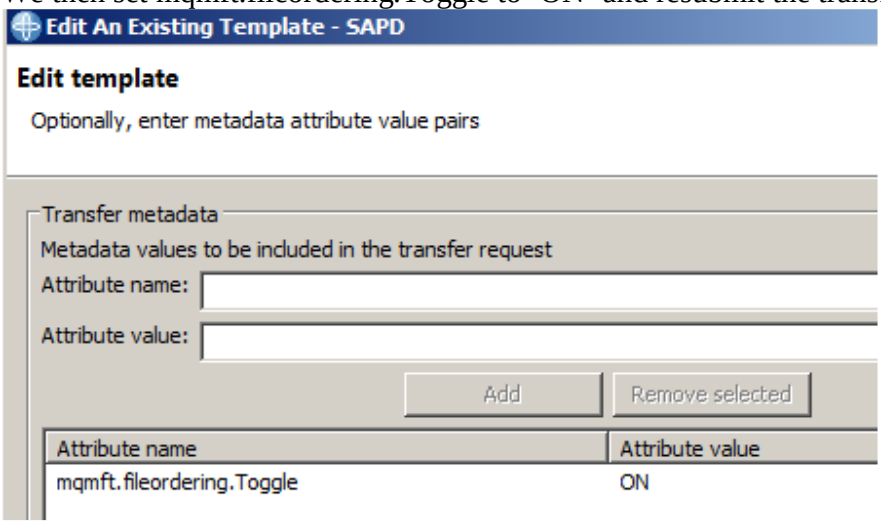


In the final log publication for the transfer, we see a message indicating that the transfer bypassed the exit because mqmft.fileordering.Toggle was not set to "On" (first we show the message text then show the relevant [part of the message in rfhutil):

```
<action time="2016-07-01T22:37:06.982Z">completed</action>
 <sourceAgent agent="LINNIE" QMgr="IIBQM" agentType="STANDARD">
  <startExits>
   <exit name="class
com.ibm.mqmft.exit.fileordering.FileOrderingSourceStartMqmftExit">
    <status resultCode="proceed">
     <supplement>com.ibm.mqmft.exit.fileordering.FileOrderingSourceStartMqmftExit:
Bypassing this exit because Toggle was not set to On in the user
metadata.</supplement>
    </status>
   </exit>
  </startExits>
```

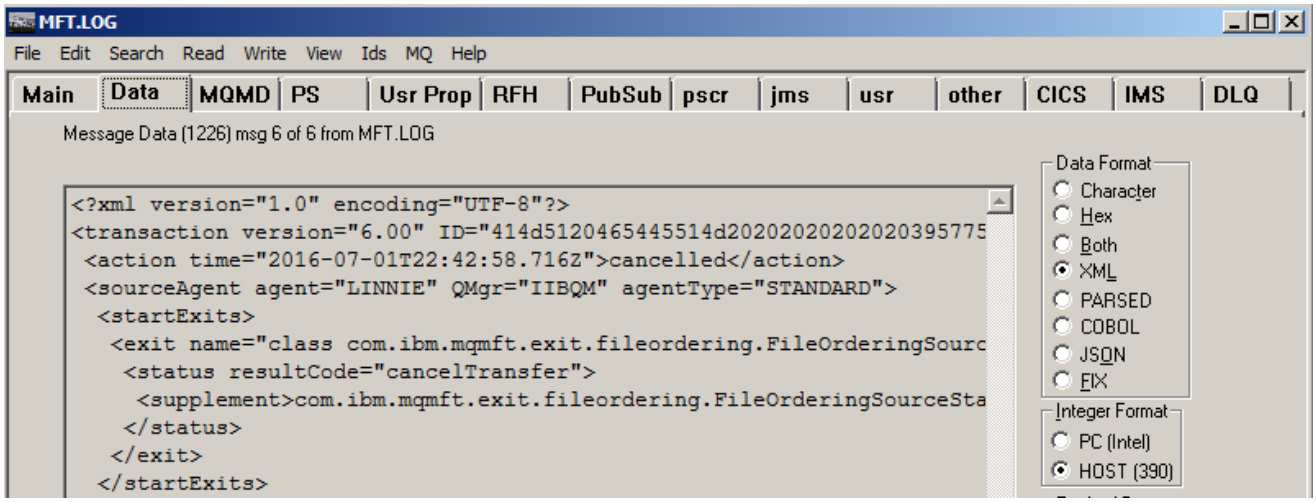


We then set mqmft.fileordering.Toggle to 'ON' and resubmit the transfer.

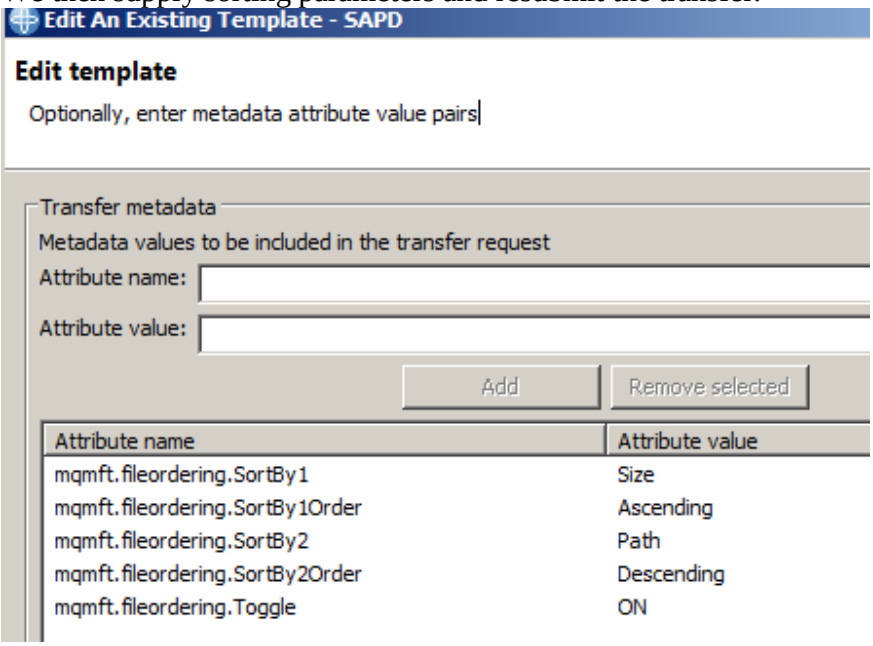


The transfer is cancelled, and we see in the log publications that the reason is that we set mqmft.fileordering.Toggle to "On" without specifying the required sorting parameters:

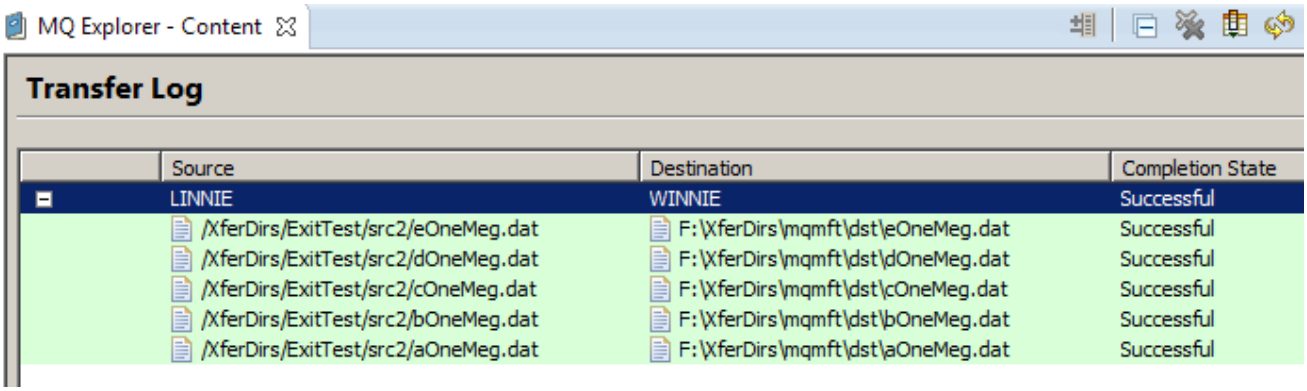
```
<action time="2016-07-01T22:42:58.716Z">cancelled</action>
 <sourceAgent agent="LINNIE" QMgr="IIBQM" agentType="STANDARD">
  <startExits>
   <exit name="class
com.ibm.mqmft.exit.fileordering.FileOrderingSourceStartMqmftExit">
    <status resultCode="cancelTransfer">

<supplement>com.ibm.mqmft.exit.fileordering.FileOrderingSourceStartMqmftExit:
Transfer cancelled.  The required keys were not found in the user metadata.
Refer to the exit documentation.  Prior log publications for this transfer
will show the metadata keys and values in use.</supplement>
    </status>
   </exit>
  </startExits>
```



We then supply sorting parameters and resubmit the transfer.

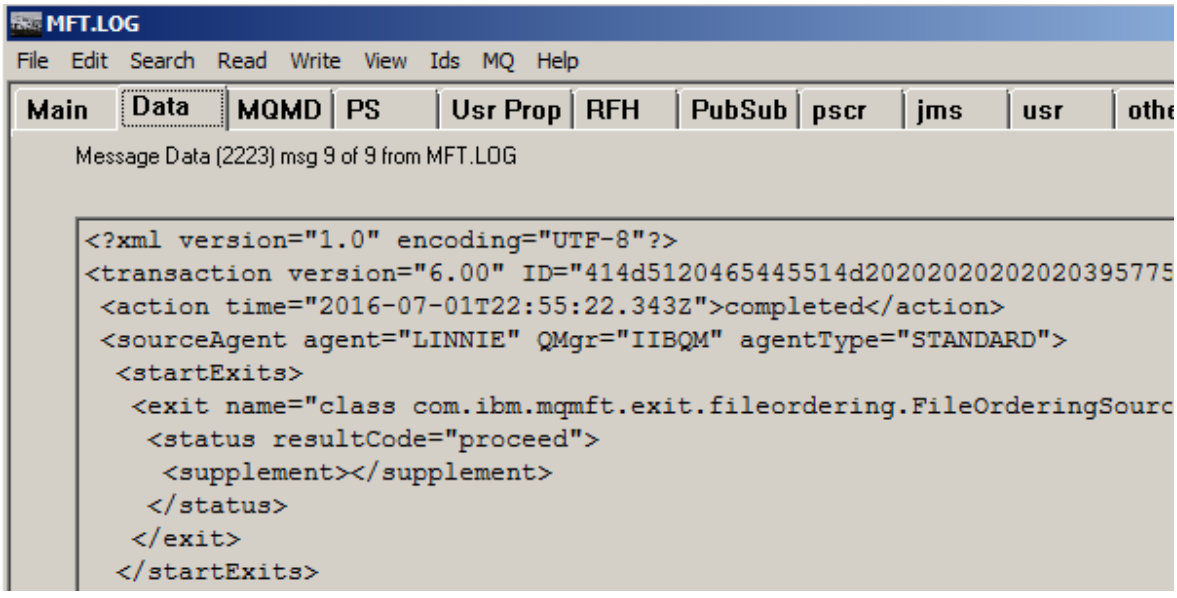


The transfer succeeds, and we see in MQ Explorer's Transfer Log that MQMFT has transferred the five files according the specified parameters: firstly, in ascending order by size, and, since all five source files are the same size, secondly in descending order by path:

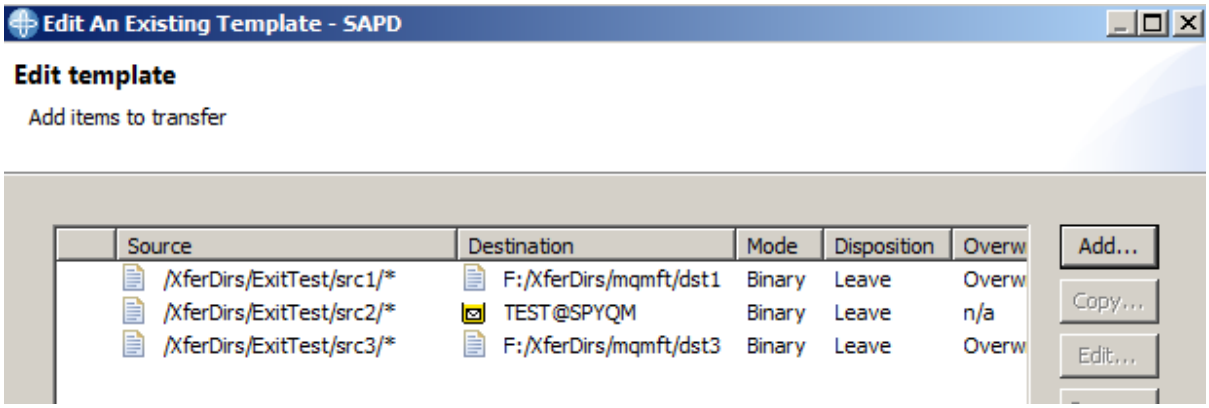


The log publications contain a corresponding notification of successful completion (simply the resultCode “proceed”):

```
<action time="2016-07-01T22:55:22.343Z">completed</action>
 <sourceAgent agent="LINNIE" QMgr="IIBQM" agentType="STANDARD">
  <startExits>
   <exit name="class
com.ibm.mqmft.exit.fileordering.FileOrderingSourceStartMqmftExit">
    <status resultCode="proceed">
     <supplement></supplement>
    </status>
   </exit>
  </startExits>
```



Let's try a second, more complex example. We alter the transfer definition so that the transfer set contains three items:



The first item transfers all of the files in the src1 directory to the dst1 directory. The second item transfers all five of the files in our first example to a queue named TEST on the SPYQM queue manager. The third item transfers all of the files in the src3 directory to the dst3 directory.

On Linux, the sixteen files in /XferDirs/ExitTest/src1 have these attributes:

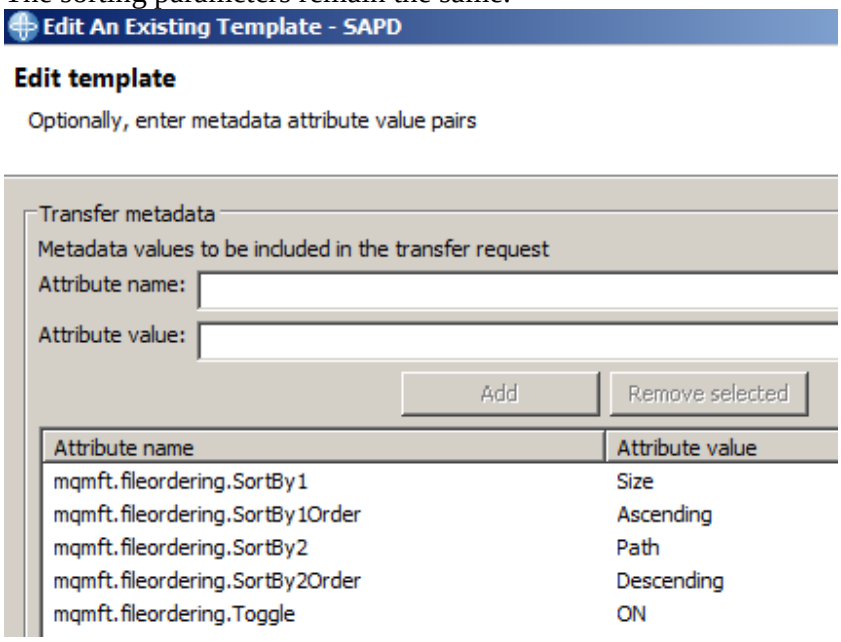
```
[sparsons@oc2518075255 src1]$ ls -l
total 450560
-rw-rw-r--. 1 sparsons avdefs 41943040 Jun 29 22:51 aFortyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 20971520 Jun 29 22:10 aSendMeFirst.sum
-rw-rw-r--. 1 sparsons avdefs 20971520 Jun 29 22:10 aSendMeLast.sum
-rw-rw-r--. 1 sparsons avdefs 31457280 Jun 29 22:51 aThirtyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 20971520 Jun 29 22:51 aTwentyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 41943040 Jun 30 11:59 bFortyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 31457280 Jun 30 11:59 bThirtyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 20971520 Jun 30 11:59 bTwentyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 41943040 Jun 29 22:53 cFortyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 31457280 Jun 29 22:53 cThirtyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 20971520 Jun 29 22:53 cTwentyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 41943040 Jun 29 22:10 dFortyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 31457280 Jun 29 22:10 dThirtyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 20971520 Jun 29 22:10 dTwentyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 20971520 Jun 29 22:10 zSendMeFirst.sum
-rw-rw-r--. 1 sparsons avdefs 20971520 Jun 29 22:10 zSendMeLast.sum
[sparsons@oc2518075255 src1]$
```

On Linux, the five files in /XferDirs/ExitTest/src2 remain the same as they were in the first example.

On Linux, the sixteen files in /XferDirs/ExitTest/src3 have the same attributes, except differing timestamps, as the files in / XferDirs/ExitTest/src1:

```
[sparsons@oc2518075255 src3]$ pwd
/XferDirs/ExitTest/src3
[sparsons@oc2518075255 src3]$ ls -l
total 450560
-rw-rw-r--. 1 sparsons avdefs 41943040 Jul  1 16:13 aFortyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 20971520 Jul  1 16:13 aSendMeFirst.sum
-rw-rw-r--. 1 sparsons avdefs 20971520 Jul  1 16:13 aSendMeLast.sum
-rw-rw-r--. 1 sparsons avdefs 31457280 Jul  1 16:13 aThirtyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 20971520 Jul  1 16:13 aTwentyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 41943040 Jul  1 16:19 bFortyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 31457280 Jul  1 16:19 bThirtyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 20971520 Jul  1 16:19 bTwentyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 41943040 Jul  1 16:13 cFortyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 31457280 Jul  1 16:13 cThirtyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 20971520 Jul  1 16:13 cTwentyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 41943040 Jul  1 16:13 dFortyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 31457280 Jul  1 16:13 dThirtyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 20971520 Jul  1 16:13 dTwentyMeg.dat
-rw-rw-r--. 1 sparsons avdefs 20971520 Jul  1 16:13 zSendMeFirst.sum
-rw-rw-r--. 1 sparsons avdefs 20971520 Jul  1 16:13 zSendMeLast.sum
[sparsons@oc2518075255 src3]$
```

The sorting parameters remain the same:



We submit the transfer, and it succeeds. In MQ Explorer's Transfer Log, we check the order in which MQMFT transferred the files:

Transfer Log

Source	Destination	Completion Sta
LINNIE	WINNIE	Successful
/XferDirs/ExitTest/src2/eOneMeg.dat	TEST@SPYQM	Successful
/XferDirs/ExitTest/src2/dOneMeg.dat	TEST@SPYQM	Successful
/XferDirs/ExitTest/src2/cOneMeg.dat	TEST@SPYQM	Successful
/XferDirs/ExitTest/src2/bOneMeg.dat	TEST@SPYQM	Successful
/XferDirs/ExitTest/src2/aOneMeg.dat	TEST@SPYQM	Successful
/XferDirs/ExitTest/src3/zSendMeLast.sum	F:\XferDirs\mqmft\dst3\zSendMeLast.sum	Successful
/XferDirs/ExitTest/src3/zSendMeFirst.sum	F:\XferDirs\mqmft\dst3\zSendMeFirst.sum	Successful
/XferDirs/ExitTest/src3/dTwentyMeg.dat	F:\XferDirs\mqmft\dst3\dTwentyMeg.dat	Successful
/XferDirs/ExitTest/src3/cTwentyMeg.dat	F:\XferDirs\mqmft\dst3\cTwentyMeg.dat	Successful
/XferDirs/ExitTest/src3/bTwentyMeg.dat	F:\XferDirs\mqmft\dst3\bTwentyMeg.dat	Successful
/XferDirs/ExitTest/src3/aTwentyMeg.dat	F:\XferDirs\mqmft\dst3\aTwentyMeg.dat	Successful
/XferDirs/ExitTest/src3/aSendMeLast.sum	F:\XferDirs\mqmft\dst3\aSendMeLast.sum	Successful
/XferDirs/ExitTest/src3/aSendMeFirst.sum	F:\XferDirs\mqmft\dst3\aSendMeFirst.sum	Successful
/XferDirs/ExitTest/src1/zSendMeLast.sum	F:\XferDirs\mqmft\dst1\zSendMeLast.sum	Successful
/XferDirs/ExitTest/src1/zSendMeFirst.sum	F:\XferDirs\mqmft\dst1\zSendMeFirst.sum	Successful
/XferDirs/ExitTest/src1/dTwentyMeg.dat	F:\XferDirs\mqmft\dst1\dTwentyMeg.dat	Successful
/XferDirs/ExitTest/src1/cTwentyMeg.dat	F:\XferDirs\mqmft\dst1\cTwentyMeg.dat	Successful
/XferDirs/ExitTest/src1/bTwentyMeg.dat	F:\XferDirs\mqmft\dst1\bTwentyMeg.dat	Successful
/XferDirs/ExitTest/src1/aTwentyMeg.dat	F:\XferDirs\mqmft\dst1\aTwentyMeg.dat	Successful
/XferDirs/ExitTest/src1/aSendMeLast.sum	F:\XferDirs\mqmft\dst1\aSendMeLast.sum	Successful
/XferDirs/ExitTest/src1/aSendMeFirst.sum	F:\XferDirs\mqmft\dst1\aSendMeFirst.sum	Successful
/XferDirs/ExitTest/src3/dThirtyMeg.dat	F:\XferDirs\mqmft\dst3\dThirtyMeg.dat	Successful
/XferDirs/ExitTest/src3/cThirtyMeg.dat	F:\XferDirs\mqmft\dst3\cThirtyMeg.dat	Successful
/XferDirs/ExitTest/src3/bThirtyMeg.dat	F:\XferDirs\mqmft\dst3\bThirtyMeg.dat	Successful
/XferDirs/ExitTest/src3/aThirtyMeg.dat	F:\XferDirs\mqmft\dst3\aThirtyMeg.dat	Successful
/XferDirs/ExitTest/src1/dThirtyMeg.dat	F:\XferDirs\mqmft\dst1\dThirtyMeg.dat	Successful
/XferDirs/ExitTest/src1/cThirtyMeg.dat	F:\XferDirs\mqmft\dst1\cThirtyMeg.dat	Successful
/XferDirs/ExitTest/src1/bThirtyMeg.dat	F:\XferDirs\mqmft\dst1\bThirtyMeg.dat	Successful
/XferDirs/ExitTest/src1/aThirtyMeg.dat	F:\XferDirs\mqmft\dst1\aThirtyMeg.dat	Successful
/XferDirs/ExitTest/src3/dFortyMeg.dat	F:\XferDirs\mqmft\dst3\dFortyMeg.dat	Successful
/XferDirs/ExitTest/src3/cFortyMeg.dat	F:\XferDirs\mqmft\dst3\cFortyMeg.dat	Successful
/XferDirs/ExitTest/src3/bFortyMeg.dat	F:\XferDirs\mqmft\dst3\bFortyMeg.dat	Successful
/XferDirs/ExitTest/src3/aFortyMeg.dat	F:\XferDirs\mqmft\dst3\aFortyMeg.dat	Successful
/XferDirs/ExitTest/src1/dFortyMeg.dat	F:\XferDirs\mqmft\dst1\dFortyMeg.dat	Successful
/XferDirs/ExitTest/src1/cFortyMeg.dat	F:\XferDirs\mqmft\dst1\cFortyMeg.dat	Successful
/XferDirs/ExitTest/src1/bFortyMeg.dat	F:\XferDirs\mqmft\dst1\bFortyMeg.dat	Successful
/XferDirs/ExitTest/src1/aFortyMeg.dat	F:\XferDirs\mqmft\dst1\aFortyMeg.dat	Successful

The instructions specified to transfer firstly in ascending order by size, then, when the size matches, in descending order by path. The above screen capture shows that MQMFT transferred the files according to these instructions. Notice that the exit applied the sorting parameters across all of the source files defined in the transfer set, not to individual items (e.g. only the source files in the /XferDirs/ExitTest/src1 directory) in the transfer set.

Let’s try a third, more complex example. We will alter the transfer definition to the converse: We will transfer from WINNIE to LINNIE, and move the same items from the prior example back.

Edit template

Add items to transfer

Source	Destination	Mode	Disposition	Overwrite
F:\XferDirs\mqmft\dst1*	/XferDirs/ExitTest/src1	Binary	Leave	Overwrite
TEST@SPYQM	/XferDirs/ExitTest/src2/FromTESTQueue.dat	Binary	n/a	Overwrite
F:\XferDirs\mqmft\dst3*	/XferDirs/ExitTest/src3	Binary	Leave	Overwrite

The first item transfers the files in the dst1 directory back to the src1 directory. The second item is a message-to-file transfer that reads the messages written to the TEST queue in the prior example to a file named FromTESTQueue.dat. The third item transfers the files in the dst3 directory back to the src3 directory.

Let’s alter the exit parameters, adding positioning parameters.

Edit An Existing Template - SAPD

Edit template

Optionally, enter metadata attribute value pairs

Transfer metadata

Metadata values to be included in the transfer request

Attribute name:

Attribute value:

Add | Remove selected

Attribute name	Attribute value
mqmft.fileordering.First	*SendMeF*.sum
mqmft.fileordering.FirstSortBy1	Size
mqmft.fileordering.FirstSortBy1Order	Ascending
mqmft.fileordering.FirstSortBy2	Path
mqmft.fileordering.FirstSortBy2Order	Descending
mqmft.fileordering.Last	*SendMeL*.sum
mqmft.fileordering.LastSortBy1	Size
mqmft.fileordering.LastSortBy1Order	Ascending
mqmft.fileordering.LastSortBy2	Path
mqmft.fileordering.LastSortBy2Order	Descending
mqmft.fileordering.SortBy1	Size
mqmft.fileordering.SortBy1Order	Ascending
mqmft.fileordering.SortBy2	Path
mqmft.fileordering.SortBy2Order	Descending
mqmft.fileordering.Toggle	ON

We submit the transfer, and it succeeds. In MQ Explorer's Transfer Log, we see the order in which MQMFT transferred the items.

Transfer Log

	Source	Destination	Completion State
⊟	WINNIE	LINNIE	Successful
	F:\XferDirs\mqmft\dst3\zSendMeFirst.sum	/XferDirs/ExitTest/src3/zSendMeFirst.sum	Successful
	F:\XferDirs\mqmft\dst3\aSendMeFirst.sum	/XferDirs/ExitTest/src3/aSendMeFirst.sum	Successful
	F:\XferDirs\mqmft\dst1\zSendMeFirst.sum	/XferDirs/ExitTest/src1/zSendMeFirst.sum	Successful
	F:\XferDirs\mqmft\dst1\aSendMeFirst.sum	/XferDirs/ExitTest/src1/aSendMeFirst.sum	Successful
	F:\XferDirs\mqmft\dst3\dTwentyMeg.dat	/XferDirs/ExitTest/src3/dTwentyMeg.dat	Successful
	F:\XferDirs\mqmft\dst3\cTwentyMeg.dat	/XferDirs/ExitTest/src3/cTwentyMeg.dat	Successful
	F:\XferDirs\mqmft\dst3\bTwentyMeg.dat	/XferDirs/ExitTest/src3/bTwentyMeg.dat	Successful
	F:\XferDirs\mqmft\dst3\aTwentyMeg.dat	/XferDirs/ExitTest/src3/aTwentyMeg.dat	Successful
	F:\XferDirs\mqmft\dst1\dTwentyMeg.dat	/XferDirs/ExitTest/src1/dTwentyMeg.dat	Successful
	F:\XferDirs\mqmft\dst1\cTwentyMeg.dat	/XferDirs/ExitTest/src1/cTwentyMeg.dat	Successful
	F:\XferDirs\mqmft\dst1\bTwentyMeg.dat	/XferDirs/ExitTest/src1/bTwentyMeg.dat	Successful
	F:\XferDirs\mqmft\dst1\aTwentyMeg.dat	/XferDirs/ExitTest/src1/aTwentyMeg.dat	Successful
	F:\XferDirs\mqmft\dst3\dThirtyMeg.dat	/XferDirs/ExitTest/src3/dThirtyMeg.dat	Successful
	F:\XferDirs\mqmft\dst3\cThirtyMeg.dat	/XferDirs/ExitTest/src3/cThirtyMeg.dat	Successful
	F:\XferDirs\mqmft\dst3\bThirtyMeg.dat	/XferDirs/ExitTest/src3/bThirtyMeg.dat	Successful
	F:\XferDirs\mqmft\dst3\aThirtyMeg.dat	/XferDirs/ExitTest/src3/aThirtyMeg.dat	Successful
	F:\XferDirs\mqmft\dst1\dThirtyMeg.dat	/XferDirs/ExitTest/src1/dThirtyMeg.dat	Successful
	F:\XferDirs\mqmft\dst1\cThirtyMeg.dat	/XferDirs/ExitTest/src1/cThirtyMeg.dat	Successful
	F:\XferDirs\mqmft\dst1\bThirtyMeg.dat	/XferDirs/ExitTest/src1/bThirtyMeg.dat	Successful
	F:\XferDirs\mqmft\dst1\aThirtyMeg.dat	/XferDirs/ExitTest/src1/aThirtyMeg.dat	Successful
	F:\XferDirs\mqmft\dst3\dFortyMeg.dat	/XferDirs/ExitTest/src3/dFortyMeg.dat	Successful
	F:\XferDirs\mqmft\dst3\cFortyMeg.dat	/XferDirs/ExitTest/src3/cFortyMeg.dat	Successful
	F:\XferDirs\mqmft\dst3\bFortyMeg.dat	/XferDirs/ExitTest/src3/bFortyMeg.dat	Successful
	F:\XferDirs\mqmft\dst3\aFortyMeg.dat	/XferDirs/ExitTest/src3/aFortyMeg.dat	Successful
	F:\XferDirs\mqmft\dst1\dFortyMeg.dat	/XferDirs/ExitTest/src1/dFortyMeg.dat	Successful
	F:\XferDirs\mqmft\dst1\cFortyMeg.dat	/XferDirs/ExitTest/src1/cFortyMeg.dat	Successful
	F:\XferDirs\mqmft\dst1\bFortyMeg.dat	/XferDirs/ExitTest/src1/bFortyMeg.dat	Successful
	F:\XferDirs\mqmft\dst1\aFortyMeg.dat	/XferDirs/ExitTest/src1/aFortyMeg.dat	Successful
	TEST@SPYQM	/XferDirs/ExitTest/src2/FromTESTQueue.dat/TEST@SPYQM	Successful
	F:\XferDirs\mqmft\dst3\zSendMeLast.sum	/XferDirs/ExitTest/src3/zSendMeLast.sum	Successful
	F:\XferDirs\mqmft\dst3\aSendMeLast.sum	/XferDirs/ExitTest/src3/aSendMeLast.sum	Successful
	F:\XferDirs\mqmft\dst1\zSendMeLast.sum	/XferDirs/ExitTest/src1/zSendMeLast.sum	Successful
	F:\XferDirs\mqmft\dst1\aSendMeLast.sum	/XferDirs/ExitTest/src1/aSendMeLast.sum	Successful

MQMFT first transferred the files matching the regular expression for mqmft.fileordering.First, firstly in ascending order by size, then in descending order by path. Then MQMFT transferred the remaining files that did not match the regular expression for mqmft.fileordering.Last, firstly in ascending order by size, then in descending order by path. Then MQMFT performed the message-to-file transfer, which is not subject to the sorting and positioning parameters; the exit will move message-to-file transfers either last, or, if the transfer set contains files that match the Last positioning parameters, penultimate. Lastly MQMFT transferred the files matching the regular expression for mqmft.fileordering.Last, firstly in ascending order by size, then in descending order by path. The exit correctly applied source file ordering as specified in the transfer's metadata.

Configuration

The exit allows the user to set the transfer order of files based on the input files' name, path, last modified timestamp, and size. The user supplies these sorting and positioning instructions in the transfer's user metadata. The following table describes the use of the sorting and positioning parameters.

Parameter Name (case-sensitive: type exactly as shown)	**Description**	**Required**	**Permitted Values (case-insensitive)**	**Comments**
mqmft.fileordering.Toggle	Unless it is set to "On", the transfer will skirt this exit.	No	{any}	If Toggle is not specified, the transfer will skirt this exit.
mqmft.fileordering.SortBy1	Sorting: The parameter that this exit will first sort files on.	Yes	LastModified, Size, Name	Name means the name of the input file, excluding its path.
mqmft.fileordering.SortBy2	Sorting: The parameter that this exit will use to break a tie when two files have the same value for the SortBy1 parameter.	Yes	LastModified, Size, Path	Path means the input file's full absolute path, including the filename. SortBy1 and SortBy2 may not have matching values.
mqmft.fileordering.SortBy1Order	Sorting: The ordering for files sorted on the SortBy1 parameter.	Yes	Ascending, Descending	• LastModified, Ascending: From oldest to newest • Size, Ascending: From smallest to biggest • Name, Ascending: From A to Z • LastModified, Descending: From newest to

				oldest • Size, Descending: From biggest to smallest • Name, Descending: From Z to A
mqmft.fileordering.SortBy2Order	Sorting: The ordering for files sorted on the SortBy2 parameter.	Yes	Ascending, Descending	• LastModified, Ascending: From oldest to newest • Size, Ascending: From smallest to biggest • Path, Ascending: From A to Z • LastModified, Descending: From newest to oldest • Size, Descending: From biggest to smallest • Path, Descending: From Z to A
mqmft.fileordering.First	Positioning: Files that match the given regular expression will be transferred first.	No, but if any of the First parameters are present, all of the First parameters must be present.	A regular expression. A "*" matches any multiple characters, a "?" matches any single character.	Matches on an input file's full path name, not only on the file name. If an input file's path matches both of the First and Last regular expressions, then the First parameters win, and the file will be positioned among the First files.
mqmft.fileordering.FirstSortBy1	The parameter that this exit will use to sort when multiple files match	No, but if any of the First parameters are present, all of the First	LastModified, Size, Name	Name means the name of the input file, excluding its path.

	the regular expression given for First.	parameters must be present.		
mqmft.fileordering.FirstSortBy2	The parameter that this exit will use to break a tie when two files have the same value for the FirstSortBy1 parameter.	No, but if any of the First parameters are present, all of the First parameters must be present.	LastModified, Size, Path	Path means the input file's full absolute path, including the filename. FirstSortBy1 and FirstSortBy2 may not have matching values.
mqmft.fileordering.FirstSortBy1Order	The ordering for files sorted on the FirstSortBy1 parameter.	No, but if any of the First parameters are present, all of the First parameters must be present.	Ascending, Descending	• LastModified, Ascending: From oldest to newest • Size, Ascending: From smallest to biggest • Name, Ascending: From A to Z • LastModified, Descending: From newest to oldest • Size, Descending: From biggest to smallest • Name, Descending: From Z to A
mqmft.fileordering.FirstSortBy2Order	The ordering for files sorted on the FirstSortBy2 parameter.	No, but if any of the First parameters are present, all of the First parameters must be present.	Ascending, Descending	• LastModified, Ascending: From oldest to newest • Size, Ascending: From smallest to biggest • Path, Ascending: From A to Z • LastModified, Descending: From newest to oldest

				• Size, Descending: From biggest to smallest • Path, Descending: From Z to A
mqmft.fileordering.Last	Positioning: Files that match the given regular expression will be transferred last.	No, but if any of the Last parameters are present, all of the Last parameters must be present.	A regular expression. A "*" matches any multiple characters, a "?" matches any single character.	Matches on an input file's full path name, not only on the file name. If an input file's path matches both of the First and Last regular expressions, then the First parameters win, and the file will be positioned among the First files.
mqmft.fileordering.LastSortBy1	The parameter that this exit will use to sort when multiple files match the regular expression given for Last.	No, but if any of the Last parameters are present, all of the Last parameters must be present.	LastModified, Size, Name	Name means the name of the input file, excluding its path.
mqmft.fileordering.LastSortBy2	The parameter that this exit will use to break a tie when two files have the same value for the LastSortBy1 parameter.	No, but if any of the Last parameters are present, all of the Last parameters must be present.	LastModified, Size, Path	Path means the input file's full absolute path, including the filename. LastSortBy1 and LastSortBy2 may not have matching values.
mqmft.fileordering.LastSortBy1Order	The ordering for files sorted on the LastSortBy1 parameter.	No, but if any of the Last parameters are present, all	Ascending, Descending	• LastModified, Ascending: From oldest to newest • Size, Ascending: From smallest to biggest

		of the Last parameters must be present.		• Name, Ascending: From A to Z • LastModified, Descending: From newest to oldest • Size, Descending: From biggest to smallest • Name, Descending: From Z to A
mqmft.fileordering.LastSortBy2Order	The ordering for files sorted on the LastSortBy2 parameter.	No, but if any of the Last parameters are present, all of the Last parameters must be present.	Ascending, Descending	• LastModified, Ascending: From oldest to newest • Size, Ascending: From smallest to biggest • Path, Ascending: From A to Z • LastModified, Descending: From newest to oldest • Size, Descending: From biggest to smallest • Path, Descending: From Z to A

Important to Understand

- The exit applies sorting and positioning only to files at the source of a transfer. It can not and does not apply sorting and positioning to destination files, because they likely do not exist at the time of the transfer. It also can not and does not apply sorting and positioning to messages on MQ queues at the source of a transfer.
- The exit has no effect on message-to-file transfers. If the transfer set includes only message-to-file transfers, the transfer will bypass the exit, with a corresponding message written to a log publication. If the transfer set includes a mixture of file-to-file, message-to-file, and/or file-to-message transfers, then the exit will position the message-to-file transfers either last or, if files match the regular expression for mqmft.fileordering.Last, penultimate before those files.
- The exit has an immaterial impact upon performance. Testing indicates that users should expect a 0 to 6% performance cost from using the exit, with the possibility of exceptional results above and below that range.
- The scope of the sorting and positioning is limited to the current transfer. The exit does not allow for sorting or positioning files across multiple transfers.
- The exit applies the sorting and positioning across all of the files in a transfer. The exit cannot limit the scope of the sorting and positioning to one item within a transfer set.
- If the same source file appears more than once in the transfer set, then the exit will include only its first occurrence in the files transferred. For example, with the exit deployed and mqmft.fileordering.Toggle set to “On”, if you first specify to transfer a file to a directory, then subsequently in the same transfer specify to transfer that same file to a queue, only the file-to-file transfer will happen.
- File timestamps are usually more fine-grained than their values in command outputs or graphical panels; therefore, files that appear to have the same timestamp likely may not. Please bear this fact in mind when viewing the results of files transferred with a sorting or positioning parameter set to LastModified.
- As mentioned already, the values for the First and Last positioning parameters must accommodate files’ full paths and not merely their names.
- The exit can not cause files to transfer in the same order that they are written to a directory, regardless of whether the transfer is started by an MQMFT resource monitor.

Supported Combinations of Parameters

This version of the exit supports the following twenty-eight combinations of sorting and positioning parameters, each of which I list in the following format:
<*SortBy1>, <*SortBy1Order>, <*SortBy2>, <*SortBy2Order>

1. LastModified, Ascending, Size, Ascending
2. LastModified, Ascending, Size, Descending
3. LastModified, Ascending, Path, Ascending
4. LastModified, Ascending, Path, Descending
5. LastModified, Descending, Size, Ascending
6. LastModified, Descending, Size, Descending
7. LastModified, Descending, Path, Ascending
8. LastModified, Descending, Path, Descending
9. Size, Ascending, LastModified, Ascending
10. Size, Ascending, LastModified, Descending
11. Size, Ascending, Path, Ascending
12. Size, Ascending, Path, Descending
13. Size, Descending, LastModified, Ascending
14. Size, Descending, LastModified, Descending
15. Size, Descending, Path, Ascending
16. Size, Descending, Path, Descending
17. Name, Ascending, LastModified, Ascending
18. Name, Ascending, LastModified, Descending
19. Name, Ascending, Path, Ascending
20. Name, Ascending, Path. Descending
21. Name, Ascending, Size, Ascending
22. Name, Ascending, Size, Descending
23. Name, Descending, LastModified, Ascending
24. Name, Descending, LastModified, Descending
25. Name, Descending, Path, Ascending
26. Name, Descending, Path, Descending
27. Name, Descending, Size, Ascending
28. Name, Descending, Size, Descending

I hope you find this exit useful.