TIBCO® Managed File Transfer Platform Server for Windows
User's Guide

Software Release 7.2
June 2016

Two-Second Advantage®

Important Information

Contents

TIBCO Documentation and Support Services

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, please visit:

https://docs.tibco.com

Product-Specific Documentation

Documentation for TIBCO products is not bundled with the software. Instead, it is available on the TIBCO Documentation site at https://docs.tibco.com/products/tibco-managed-file-transfer-platform-server-for-windows.

The following documents for this product can be found on the TIBCO Documentation site:

- *TIBCO Managed File Transfer Platform Server for Windows Installation*
- *TIBCO Managed File Transfer Platform Server for Windows User's Guide*
- *TIBCO Managed File Transfer Platform Server for Windows Release Notes*

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

  http://www.tibco.com/services/support

- If you already have a valid maintenance or support contract, visit this site:

  https://support.tibco.com

  Entry to this site requires a user name and password. If you do not have a user name, you can request one.

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

https://www.tibcommunity.com

1

1. MFT Platform Server Administrator

MFT Platform Server Administrator provides an explorer-type interface that you can use to perform the following operations:

- Directly access MFT Platform Server
- Define SSL information
- View and administer the queue of transfers
- View past transactions
- Modify server settings
- Create, modify, and delete transfers, templates, and DNI entries

1.1 Transfer Properties

When you initialize MFT Platform Server Administrator for the first time, the panel automatically connects to TIBCO® Managed File Transfer Platform Server for Windows.

To go to the Transfer GUI panel, click the transfer icon under the server name or the transfer icon on the tool bar.

If you click the transfer icon under the server name, right-click and select a new transfer, and then select Advanced TCP transfer. This brings up the transfer panel.

If you click the transfer icon on the tool bar, you can select Advanced TCP transfer. This brings up the Transfer Properties panel.

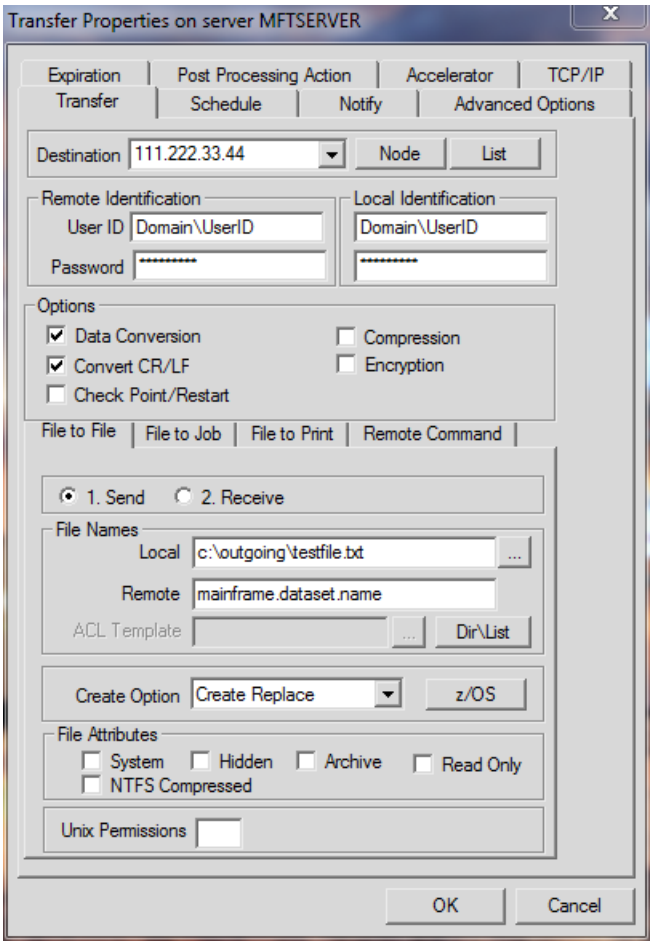
The transfer panel is divided into two sections. The upper section of the panel has several fields that are universal to all transfer types. The lower section of this panel consists of four tabs. Each tab represents a different type of transfer that is supported by TIBCO Managed File Transfer (MFT) Platform Server for Windows: File to File, File to Job, File to Print, and Remote Command.

The MFT Platform Server Administrator transfer panel changes dynamically in response to the setting of the File Transfer Type. For example, if you choose the File to File tab, a remote file name must be specified. However, if you select the File to Print tab, you must specify a remote printer name. Only fields that are pertinent to the transfer type selected are displayed on the panel.

Transfer fields and tabs default to the last value entered or last tab selected for any transaction. For example, if you have selected the File to File tab and then select OK to perform the transfer, you will be returned to the File to File tab the next time you initiate MFT Platform Server Administrator.

1.1.1 Transfer Tab

Several fields are common to all file transfers and are displayed on the upper section of the panel regardless of the transfer type selected. These fields are referred to as Universal Fields.



1.1.1.1 Universal Fields

The universal fields are the fields located on the upper section of the transfer panel.

- **Destination -** This is the address of the remote system.

  For TCP/IP transfers, this is the DNS name, WINS name, IPv4 or IPv6 address (for example, 251.250.41.5).

  The Destination field has a pull-down list that is designed to keep a list of the remote systems that are used in the past. A predefined Node can also be used in the Destination field.

- **Node -** This is the name of the remote system as defined by using the cfnode program provided with TIBCO MFT Platform Server for Windows. If a profile is associated with the node, the Remote Identification section is filled in with “Default from node”. If no profile is found, the fields are blank. You can type a Node in the Destination field and leave the User ID and Password fields blank. The information will be picked up from the profile definition if it is defined.

If any of the transfer settings conflicts with the node settings, you are notified with a pop-up message box. You can modify the transfer by clicking the OK button or you can stop the transfer by clicking the Cancel button.

- **List -** Displays a list of distribution lists available to choose from as defined in the cflist.cfg file located in the platform server installation directory.
  Note: The use of a distribution list is supported for SEND transactions only.
- **Remote Identification -** This section contains the information that is specific to the user on the remote system.

  User ID

  The User ID for the remote system, or the name by which the issuer is known to the remote system. The User ID can be up to 36 characters in length, including fifteen characters for a machine name or domain, a slash, and up to 20 characters for the User ID. The User ID is generally not case sensitive, unless going to a UNIX system. The User ID defaults to the last Issuer ID entered in this field. If a node is selected and a profile is associated with the node, this field is filled in with “Default from node”. If no profile is found, this field is blank.

  Password

  The remote password can be up to 20 characters in length and is case sensitive. For security reasons, this field is not saved in the registry as are other values. It remains in the panel for the duration of the Transfer Properties GUI execution, but must be reset at the next startup of the Transfer Properties GUI.

  With the feature called **cached passwords**, you can specify a password for a particular remote Windows User ID and store the password in the Windows registry on the remote system. In this way, you can perform MFT Platform transfers to that Windows system without having to specify the password. For more information on this feature, see Appendix B. Cached Passwords.

  Note: If your password on a remote z/OS system has expired, you cannot access a z/OS file from MFT Platform Server Administrator. To change the password, go to the password field on the main panel under the Transfer tab. Specify both the old password and the new password in the password field, separated by a slash (for example, old/new). This changes the z/OS password to the new one specified.
- **Local Identification -** In the Transfer Properties panel, you can also specify the local authentication credentials for transfers. The User ID can be up to 36 characters in length, including fifteen characters for a machine name or domain, a slash, and up to 20 characters for the User ID. The User ID in the Local Identification section is set to the user ID of the logon user. The default value for the password is "X:" which causes the platform server to read the cached password for this user. If you want to use this feature, you must first cache your password. You can override the default and enter a password or any of the other cached password keys: 'X:password', 'X:DELETE', 'X:DELETEALL', or 'X:'. For more information on cached passwords, see Appendix B. Cached Passwords.
- **Options -** You can specify Data Conversion, Convert CR/LF, Check Point/Restart, Compression, and Encryption. These options are defined as follows:

  Data Conversion - This is used to convert data between ASCII and EBCDIC. Transfers can be either binary or text. If you clear the box, the transfer is a binary transfer. If you select the box, the transfer is a text transfer. Additional parameters are available under the

Advance Options tab. If you want to use this feature, select the check box and give details under the Advanced Options tab.

Note: You have to select this box only when you are communicating with an EBCDIC system such as z/OS or IBMi. Otherwise, leave this box unselected.

Convert CR/LF - This option inserts an end-of-line character when you are receiving a file from a z/OS system. When you are sending a file to z/OS, this option removes those characters during the file transfer.

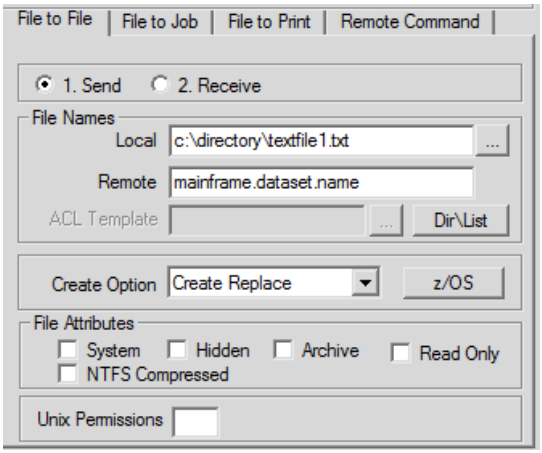
Check Point/Restart - With this parameter, packets of data can be sent periodically with the file transfer. These packets of data inform the receiver of the current point within the file. The receiver commits the latest data received to the file system and records the sender's checkpoint and its own checkpoint in the persistent queue. In the event of a failure, the initiator and the responder negotiate the saved checkpoint information and restart from the last known good checkpoint. Check Point is specified in minutes under the Advanced Options tab.

Compression - With this parameter, you can specify that compression is used for this transfer. Select the check box to turn compression on, and then go to the Advanced Options tab to select the type of compression to be used for the transfer. Compression compresses data on the sender side of the transfer and decompresses the data on the receiver side of the transfer. This results in fewer packets being sent between systems, and reduces network traffic. The compression provided by TIBCO MFT Platform Server for Windows is Smart compression because it removes a level of complexity from the user.

Encryption - With this parameter, you can turn encryption on and off. Select the check box to turn encryption on, and then go the Advanced Options tab to select the method of encryption to be used for the transfer.

1.1.1.2 File to File tab

Select the File to File tab to store the contents of the file transfer in a file.



- **1. Send -** Initiates a send request to the remote system.
- **2. Receive -** Initiates a receive request from the remote system.
- **File Names**

Local	The name by which a file is known at the local side. To Browse for the file, click the button with three dots (…). TIBCO MFT Platform Server for Windows supports standard 8.3 file names as well as UNC and long file names.
Remote	The name by which a file is known on the remote side.
ACL Template	The file name that the receiving partner uses as a template for its Access Control List (ACL). The ACL is a list that specifies users and groups and their access permissions on a file. The ACL of this file is copied to the ACL of the destination file. For this feature to function properly on Windows, the file specified must be readable by the partner which is receiving the File to File transfer and the file being created must be located on an NTFS drive. The ACL Template browse button (…) is available if the direction of the transfer is Receive.

- **Dir/List -** TIBCO MFT Platform Server for Windows supports transferring entire directories as well as sending to a distribution list. With the Dir/List button, you can set a directory transfer or transfer sent to a distribution list to stop on failure.

StopOnFailure	If the current file transfer fails, the platform server does not try to transfer the rest of the files.
ScanSubDir	The platform server scans both the directory from the file path and the subdirectories.
Test	The Local and Remote File Names are displayed to verify that the file names are correct.

- **Create Option -** You must choose one of the following options for the file being transferred:

Create	Creates a file with the same contents as the source file and with the same attributes and characteristics as specified in the source file. If the file already exists, the transaction is aborted.
Replace	Replaces the contents of the destination file with the contents of the source file.
Append	Appends the contents of the source file to the end of the destination file.
Create Replace	Creates the file if the file does not exist, and replaces its contents with the contents of the source file if the file does exist.
Create Append	Creates the file if the file does not exist, and appends the contents of the source file to the end of the destination file if the file does exist.
Create Replace New	Creates new files, replaces the existing files, and creates the path as part of the transfer if the path to the new file does not exist.

- **z/OS -** This button is only available in the File to File tab. Click this button to select the z/OS file creation options when sending files to z/OS partners.

- **File Attributes**

System	The file is a system file and can only be viewed by the operating system and not by the user.
Hidden	The file cannot be seen by the user.
Archive	Select archive if you want to mark a file that has changed since it was last backed up.
Read Only	The file being accessed can only be viewed by the user. No changes can be made to the file.
NTFS Compressed	When this feature is selected from the dialog panel, batch interface, JCL, or TSO, the file is created and compressed on the remote system. This attribute is only available on NTFS partitions. If the receiving file system is not NTFS, the file transfer fails.

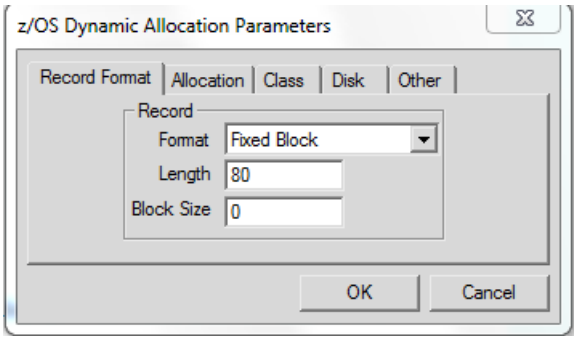
- **UNIX Permissions -** When a file is created on a UNIX system, you can set the UNIX permissions on the file. UNIX permissions are defined by a three digit number such as 777 (the same as the chmod command). The default value for this parameter is the file permissions of the file being sent or received.

  Note: Permissions are set under the file only if the file was created. In other words, UNIX permissions work only with Create, CreateReplace and CreateReplaceNew file options when the file is being created.

1.1.1.2.1 z/OS Options Panel

By clicking the z/OS button, you open the z/OS Dynamic Allocation Parameters window which provides four property pages that contain the fields that you must specify when sending files to a z/OS partner.

1.1.1.2.1.1 Record Format



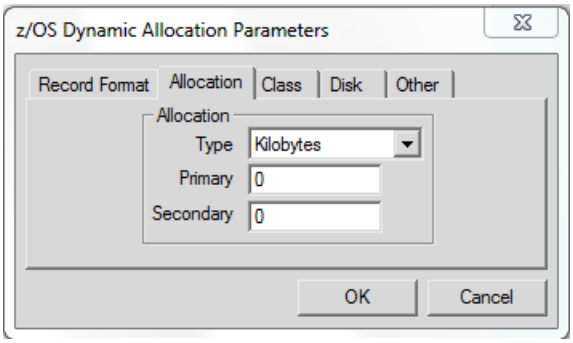
- **Format -** Determines the logical record length (LRECL). Choose one of the following formats:

Fixed	Records have a fixed length with one record per block.
Fixed ASA	Records have a fixed length with one record per block and use ASA control characters.
Fixed Block	Records have a fixed length. One or more logical records are included in each block.
Fixed Block ASA	Records have a fixed length. One or more logical records are included in each block and use ASA characters.
Fixed Block MACHINE	Records have a fixed length. One or more logical records are included in each block and use MACHINE characters.
Fixed MACHINE	Records have a fixed length with one record per block and use MACHINE characters.
Variable	Records have variable lengths. One record can be stored in a block.
Variable ASA	Records have variable lengths and use ASA control characters. One record can be stored in a block.
Variable Block	Records have variable lengths. Multiple records can be stored in a block.
Variable Block ASA	Records have variable lengths and use ASA control characters. Multiple records can be stored in a block.
Variable Block MACHINE	Records have variable lengths and use Machine control characters. Multiple records can be stored in a block.
Variable MACHINE	Records have variable lengths and use Machine control characters. One record can be stored in a block.
Undefined	Blocks are of variable sizes. One record is stored in each block. No logical record is included. The logical record length is displayed as zero. This record format is usually only used in load libraries. Block size must be used if you are specifying Undefined.

- **Length** - Record length is the maximum number of characters in a string or record of the file. The maximum number is 32760.

- **Block Size** - Specifies the size of the block. For FB, the block size must be a multiple of the record length; and for VB, the record length can be any size up to the block size minus four. The maximum number is 32760.

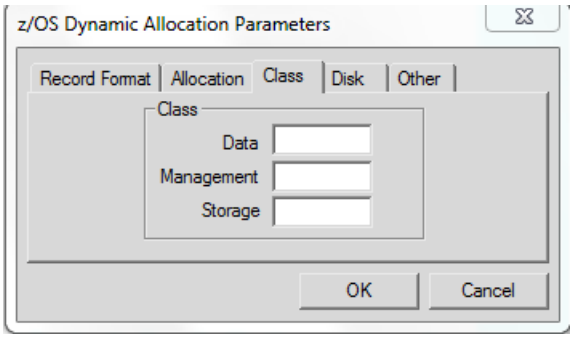
1.1.1.2.1.2 Allocation



- **Type -** The valid values are as follows: Tracks, Cylinders, Megabytes, and Kilobytes.
- **Primary -** Allocation Primary is used by the z/OS partner when creating data sets.
- **Secondary -** The secondary allocation quantity is used by the z/OS partner when creating data sets and is used when the initial space in the data set is exhausted.

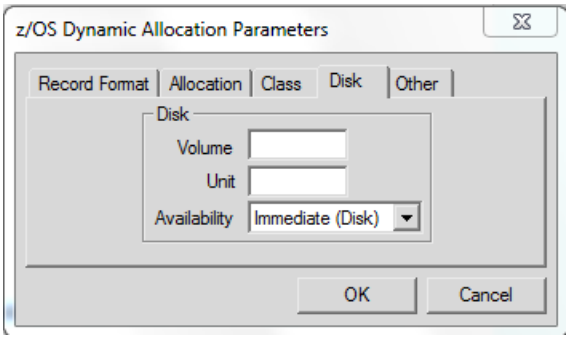
The default is Kilobytes with zero primary and zero secondary space. This default configuration picks up the size of the file being sent to the z/OS system and allocates the appropriate space.

1.1.1.2.1.3 Class



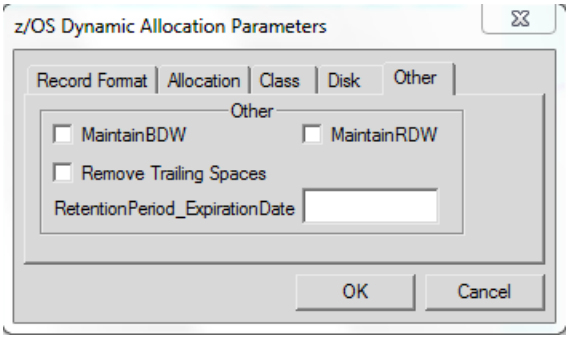
- **Data -** This represents the z/OS Data Class as defined to the Data Facility/System Managed Storage. In addition, you can use it to indirectly select file attributes such as Record Format and Logical Record Length. The value can contain 1 to 8 characters, including numeric, alphabetic, or national characters ($, #, or @). The first character must be an alphabetic or national character.
- **Management -** This represents the z/OS Management Class as defined to the Data Facility/System Managed Storage. The value can contain 1 to 8 characters, including numeric, alphabetic, or national characters ($, #, or @). The first character must be an alphabetic or national character.
- **Storage -** This represents the z/OS Storage Class as defined to the Data Facility/System Managed Storage, which is used to indicate the media type of the host file and the backup, restore, and archive policies of installation. The value can contain 1 to 8 characters. Consult your mainframe administrator for more information.

1.1.1.2.1.4 Disk



- **Volume -** This is the volume name of the disk drive on which the z/OS data set is to be allocated. It can contain 1 to 6 characters.
- **Unit -** This is the name of the type of unit where the host data set is to be allocated. It can contain 1 to 8 characters.
- **Availability** - This indicates when the remote file is available to users. The two valid values are Immediate (Disk) and Deferred (Tape).

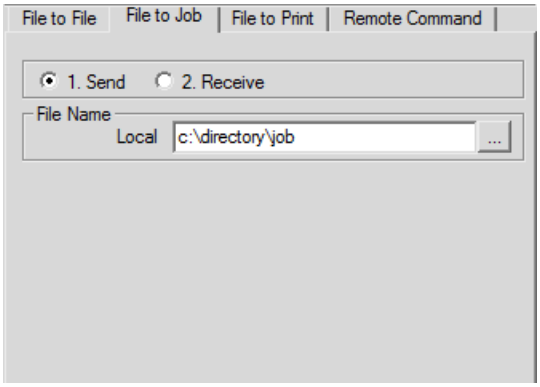
1.1.1.2.1.5 Other



- **MaintainBDW** - This option maintains the Block Descriptor Word (BDW) when sending or receiving variable block binary files to or from a z/OS system. If the data being sent or received is not in the proper BDW format, the transfer will fail.
- **MaintainRDW** - This option maintains the Record Descriptor Word (RDW) when sending or receiving variable block binary files to or from a z/OS system. If the data being sent or received is not in the proper RDW format, the transfer will fail.
- **RetentionPeriod_ExpirationDate** - This option sets the expiration date or retention period for the z/OS data sets.
- **Remove Trailing Spaces** - This option removes all spaces or binary zeros at the end of a record when transferred from the z/OS platform. This parameter is valid only when Windows is receiving a text file from z/OS.

1.1.1.3 File to Job Tab

Select the File to Job tab to send a local file to a remote system. The partner executes it as a batch job.

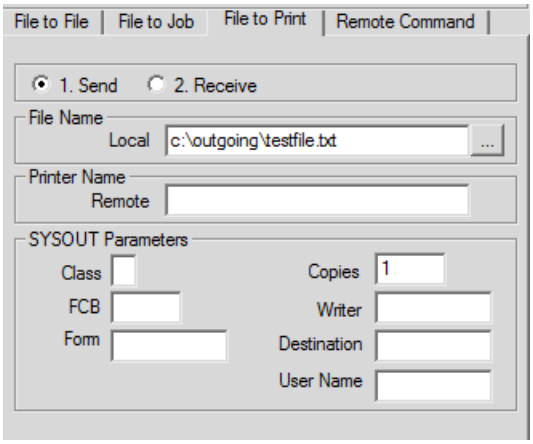


- **1. Send** - This initiates a send request to the remote system.
- **2. Receive** - This initiates a receive request from the remote system.
- **File Name** - The File Name field changes dynamically according to the direction of the transfer. If you specify Send, the Local field is displayed in the File Name section. If you specify Receive, the Remote field is displayed in the File Name section.

Local	The name by which a file is known at the local side. To browse for the file, click the button with three dots (…). TIBCO MFT Platform Server for Windows supports standard 8.3 file names as well as UNC and long file names.
Remote	The name by which a file is known on the remote side.

1.1.1.4 File to Print Tab

Select the File to Print tab to send a local file to a remote system. The partner executes it as a print job.



- **1. Send** - This initiates a send request from the local system to the remote system.
- **2. Receive** - This initiates a receive request from the remote system.
- **File Name** - The File Name field in the File to Print tab changes according to the direction of the transfer. If you specify Send, the Local field is displayed in the File Name section. If you specify Receive, the Remote field is displayed in the File Name section.

Local	The name by which a file is known at the local side. To browse for the file, click the button with three dots (…). TIBCO MFT Platform Server for Windows supports standard 8.3 file names as well as UNC and long file names.
Remote	The name by which a file is known on the remote side.

- **Printer Name** - The name of the printer to which the file is to be sent. By setting this parameter, you can send the file that is being transferred directly to the print queue or spool on the remote or local side.

  To specify a network printer, you must use the UNC name for that device. To specify a printer name by using UNC, type two backslashes (\\) before the computer name and separate the computer name from the name of the shared printer with a single backslash (\).

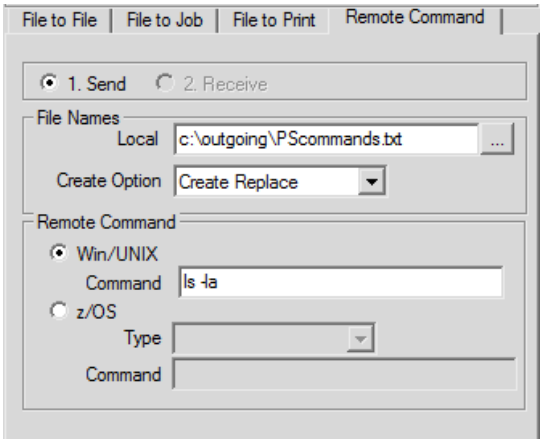
  To specify a z/OS printer, type SYSOUT@, where @ is the class to which you want to send the output. When specifying a z/OS printer, you can specify the SYSOUT parameters as described in the following table.

- **SYSOUT Parameters** - If you have specified that a z/OS printer is the destination of a file, you can use the File to Print tab to specify SYSOUT parameters. The parameters are as shown in the following table:

Class (Required)	SYSOUT Class describes to which class the JES output is routed. On a z/OS system, the printer queues are organized around a printer class. Contact the z/OS staff for the one-character alphabetic class to supply.
FCB	SYSOUT FCB is the form control block for the JES output. This symbolic name on the z/OS system is essentially a "font profile". The FCB name is defined by an administrator (or system programmer) on a z/OS system and indicates character size, and so on.
Form	SYSOUT Form indicates the name of the form on which this output is printed. The host operator receives a message to load the correct type of paper into the printer to print this report. For example, if you are printing shipping labels, the operator is prompted to put labels into the printer before the printing starts. Do not supply a value for this unless your application requires. If you do want to use this parameter, coordinate the printing with the operator at the z/OS printer site, so that the operator knows which paper form to mount when seeing this name.
Copies	The amount of copies you want to print of this item. (The default is 1.)
Writer	SYSOUT Writer indicates the external writer name that is used to process this printer file on the z/OS system. Do not specify a value for this parameter unless directed to by the system analyst on the z/OS system.
Destination	SYSOUT Destination indicates the JES print destination name. This is a symbolic name with 1 to 8 characters that identifies routing information for this print file on the z/OS system. If you do not specify this value, most z/OS systems apply a default value of "LOCAL".
User Name	SYSOUT User Name indicates the host user name (such as the TSO or RACF user name) with which the output is tagged.

1.1.1.5 Remote Command

Select the Remote Command tab to execute a command on a remote system. Output can be received in the file indicated under local file name except when the remote system is z/OS.



- **Send -** This initiates a send request from the local system to the remote system.
- **Local -** The name by which a file is known at the local side. To browse for the file, click the button with three dots (…). **Note**: The z/OS platform does not send the output back. TIBCO MFT Platform Server for Windows supports standard 8.3 file names as well as UNC and long file names.
- **Create Option -** You must choose one of the following options to save the output returned from the remote system.

Create	Creates a file on the local system. If the file already exists, the transaction is terminated.
Replace	Replaces the file on the local system. If the file does not exist, the transfer is terminated.
Append	Appends the file on the local system. If the file does not exist, the transfer is terminated.
Create Replace	Creates the file if the file does not exist on the system, and replaces its contents with the contents of the source file if the file does exist.
Create Append	Creates the file if the file does not exist on the system, and appends the contents of the source file to the end of the destination file if the file does exist.
Create Replace New	Creates new files, replaces existing files, and creates the path as part of the transfer if the path to the new file does not exist.

- **Remote Command -** In this section, fill in the command that you want to execute remotely.
- **Win/UNIX -** Select this radio button if you want the command to be executed on a Windows or UNIX platform.

  Command - This is the command that you want to execute on the remote system.

- **z/OS -** Select this radio button if you want the command to be executed on the z/OS platform.

  Type - This is the type of z/OS command that you would like to execute.

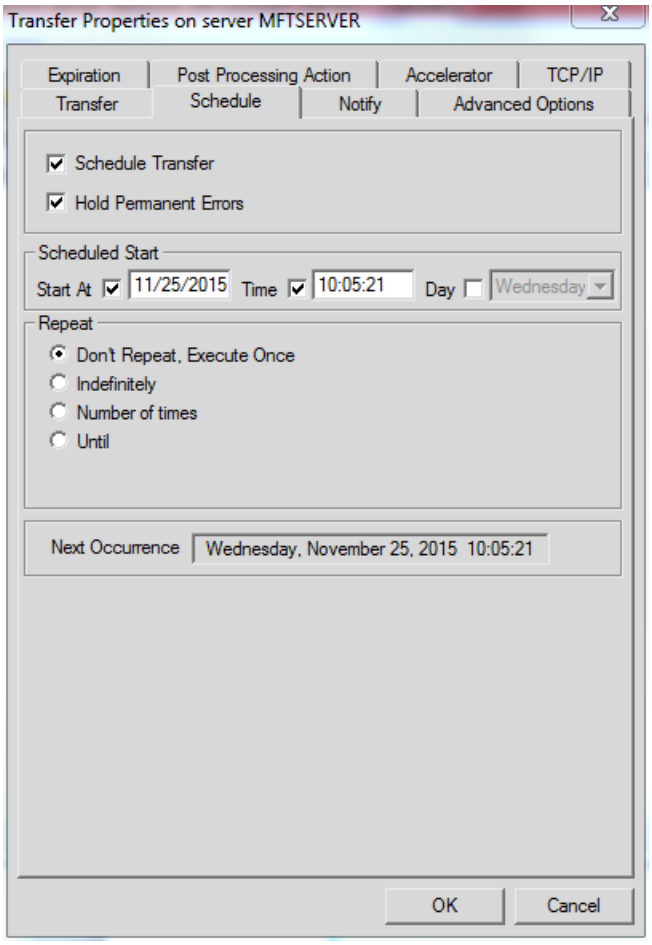
Execute/REXX Exec	You can specify an exec command or a REXX exec for execution on the remote z/OS system.
Submit JCL	You can submit a job on the remote z/OS system. This differs

	from File to Job because the JCL to run actually sits on the remote system.
Call JCL	You can call a user program defined on the remote z/OS system using JCL linkage.
Call Program	You can call a user program defined on the remote z/OS system using Program linkage.

Command - This is the command that you want to execute on the remote system.

1.1.2 Schedule Tab

You can use the Schedule property page to schedule transfer activity.



- **Schedule Transfer** - Add (select the check box) or delete (clear the check box) schedules from the transfer. If a transfer is scheduled, that takes precedence over the Check Point/Restart option and the settings under the Expiration tab.
- **Hold Permanent Errors -** This option puts a scheduled transfer on hold if a permanent error occurs. If this option is not selected, the system continues to try the transfer even after a permanent error occurred. Examples of permanent errors include the remote file not existing, bad user ID or password.
- **Scheduled Start** - In the Scheduled Start section, you can indicate when you want a file transfer to execute in the future.

Start At - This field specifies the date in which the transfer is eligible. This defaults to the current date. This entry is mutually exclusive with the value in the Day (day of week) field.

Time - This field specifies a particular time at which the transfer is eligible. This defaults to the current time.

Day - This field specifies a particular day of the week on which the transfer is eligible. This entry is mutually exclusive with the value in the Start At (date) field.

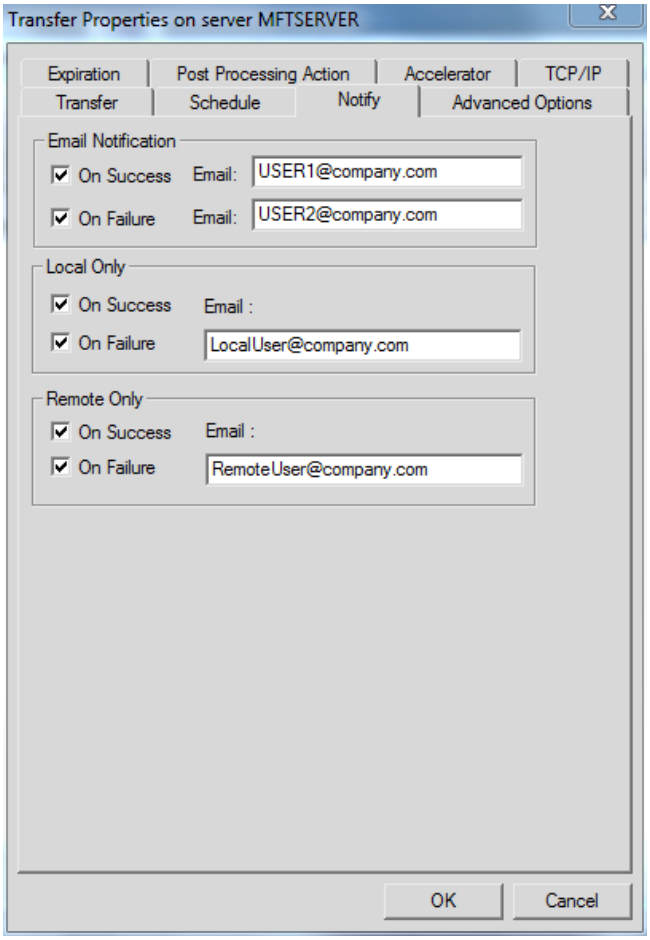
- **Repeat -** Provides information relative to the future execution (if any) of a particular file transfer after it has been executed once.

Don't Repeat, Execute Once	When this option is selected, the system executes the file transfer once, and then no longer tries it.
Indefinitely	When this option is selected, the Interval field is displayed on the panel. The system executes the transfer indefinitely (or until the current user or administrator deletes the job) and in accordance with the information specified in the Start At field and in the Interval field.
Number of times	This option specifies the number of times the file transfer can be executed before it is removed from the queue. The range for this field is from 2 to 32767. The default is 2. Similar to the Indefinitely option, when this option is selected, the Interval field is displayed.
Until	You can use this option to specify the date, time, and the day of the week until which you want to execute the file transfer. When this option is selected, fields similar to the Start At field and the Interval field where you can specify the required information are displayed.
Interval	If you specify a repeat option (with the exception of Don't Repeat, Execute Once), you can select this parameter. The drop-down list provides the following options: Daily 7 (Sunday to Saturday), Weekly, Bi-Weekly, Monthly, Bi-Monthly, Quarterly, Semi-Annually, Annually, Bi-Annually, and Every. When the option Every is selected, two additional fields that you can use to indicate the frequency with which you want to repeat the transfer are displayed. In the first field, you can insert a number. The second field contains a drop-down list which contains seconds, minutes, hour(s), day(s), week(s), month(s), and year(s).

If Scheduling is selected along with Check Point/Restart and Try Count, the system sends the transfer at the next scheduled date and time rather than sends the transfer as soon as the problem that causes the failure is resolved if for any reason your scheduled transfer fails during transmission.

1.1.3 Notify Tab

Use this property page to indicate the type of notification that you want to receive at the end of a transaction. You can specify the recipient and the method of the notification.



Select either On Success or On Failure or both for each section to receive Remote, Local, or Email Notification.

- **Email Notification**

  Email - This is the name of the user to notify when a transaction is completed. The notification informs the user whether the transaction is successful or not. If using email, ensure you have completed the SMTP field under the General tab in the MFT Platform Server Properties panel.

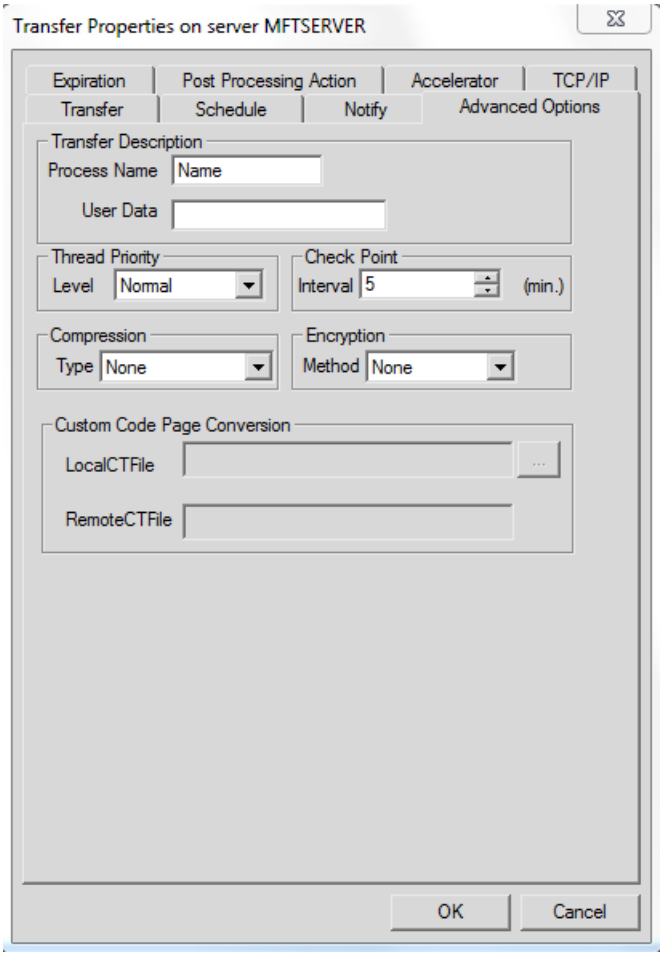
- **Local Only**

  Email - This is the name of the local user to notify when a transaction is completed. The notification informs the local user whether the transaction is successful or not. Ensure you have completed the SMTP field under the General tab in the MFT Platform Server Properties panel.

- **Remote Only**

  Email - This is the name of the remote user to notify when a transaction is completed. The notification informs the remote user whether the transaction is successful or not. Ensure you have completed the SMTP field under the General tab in the MFT Platform Server Properties panel.

1.1.4 Advanced Options Tab



- **Transfer Description**

  Process Name - This eight-character field describes the application which initiates the transfer. As an alternative to an 8-character description, the parameter $(TIME) can be used in this field to give an 8-digit time for the Process Name.

  This field can be used for automating transactions from the Host. See Appendix C Automated Operations of *TIBCO Managed File Transfer Platform Server for z/OS User's Guide*.

  User Data - This entry is logged into the history files that contain information describing the transfer on the local and remote systems. It can contain any alpha, numeric, or national characters of up to 25 characters. You can omit this parameter.

  This field can be used for automating transactions from the Host. See Appendix C Automated Operations of *TIBCO Managed File Transfer Platform Server for z/OS User's Guide.*

- **Thread Priority -** Assigns priority to transactions that are executing simultaneously and are therefore competing for resources. This priority is assigned when creating the transfer thread. This is not the priority that defines when the transfer will be executed.

  Note: Because network bandwidth is typically the limiting factor in transfer speeds, there is limited benefit in setting this parameter.

Level - The levels of priority that can be assigned are: Highest, Above Normal, Normal, Below Normal, Lowest, and Idle.

- **Check Point -** Check Point periodically sends packets of data with the file transfer that inform the receiver of the current point of the file transfer. The receiver takes the latest data received to the file system and records the sender's checkpoint and its own checkpoint in the persistent queue. In the event of a failure, the initiator and the responder negotiate with the saved checkpoint information and restart from the last known good checkpoint.

  Interval (in minutes) - The TIBCO MFT Platform Server for Windows checkpoint uses a time interval to determine when to send a checkpoint. Because checkpoint is time-based, it always occurs at a regular interval.

  Check Point Interval is specified in minutes and has a valid range of 1 - 90 minutes. The default value is 5 minutes.

- **Compression -** Compression compresses data on the sender side of the transfer and decompresses the data on the receiver side of the transfer. This results in fewer packets being sent between systems, and reduces network traffic.

  Type - TIBCO MFT Platform Server for Windows provides three different compression algorithms: Limpel-Zev (LZ), Run Length Encoding (RLE), and ZLIB1 – ZLIB9. You can select the algorithm which best suits your network. The default is None.

  TIBCO MFT Platform Server for Windows reports to the Windows Event Log when an initiator specifies Compression and communicates with a responder that does not support compression.

  The following table lists description of each algorithm:

LZ	LZ provides better compression ratios and compresses a wider variety of data types than RLE. Choose LZ if you want better compression ratios and can spare CPU cycles. Generally, ZLIB2 compresses data better and uses less CPU cycles than LZ.
RLE	RLE is more data-dependent than LZ or ZLIB. RLE compresses repeated characters such as spaces of nulls. Choose RLE if your network bandwidth is not a critical bottleneck for your network and you must save CPU cycles.
ZLIB1 through ZLIB9	ZLIB1 through ZLIB9 refer to levels of ZLIB compression. The higher the level, the greater degree of compression, but the more CPU cycles that are used. ZLIB2 typically offers the best compromise between compression and CPU usage.
None	No compression is used for this transfer.
Default from Node	If Default from Node is chosen, the type of compression is taken from the Node setting or it is set to None for non-Node transfers.

- **Encryption** - With this parameter, you can turn encryption on and off. The method of encryption can be DES, Triple DES (3DES), Blowfish, Blowfish Long, AES (Rijndael), Default, or None.

  When encryption is required, it is good practice for you to use AES encryption. It is the most secure encryption method, and offers better performance compared to DES or 3DES. When communicating with z/OS, AES encryption allows the z/OS mainframe to use hardware compression to save CPU cycles.

DES (56 bit encryption)	DES (Data Encryption Standard) is a symmetric cryptographic algorithm, in which one secret key is used for encryption and decryption of the data being sent. DES uses a 56 bit encryption key.
3DES (112 bit encryption)	3DES is DES performed three times with two secret keys applied in a particular order giving you 112 bit encryption.
Blowfish (56 bit encryption)	Blowfish is a block encryption algorithm that can use keys from 40 to 448 bits long. The platform server implementation of Blowfish uses a 56 bit encryption key.
Blowfish Long (448 bit encryption)	This Blowfish block encryption algorithm uses a key of 448 bits long (also known as Blowfish Long encryption). It is very fast, about six times faster than DES, and about fifteen times faster than 3DES.
AES(Rijndael) (256 bit encryption)	AES is a symmetric block encryption algorithm that uses a key length of 256 bits. It was selected as the Advanced Encryption Standard (AES) by the US Government.
Default	If Default is chosen, the type of encryption is taken from a Node that has been configured or it is set to None for non-Node transfers.
None	No encryption is used for this transfer.

- **Custom Code Page Conversion**

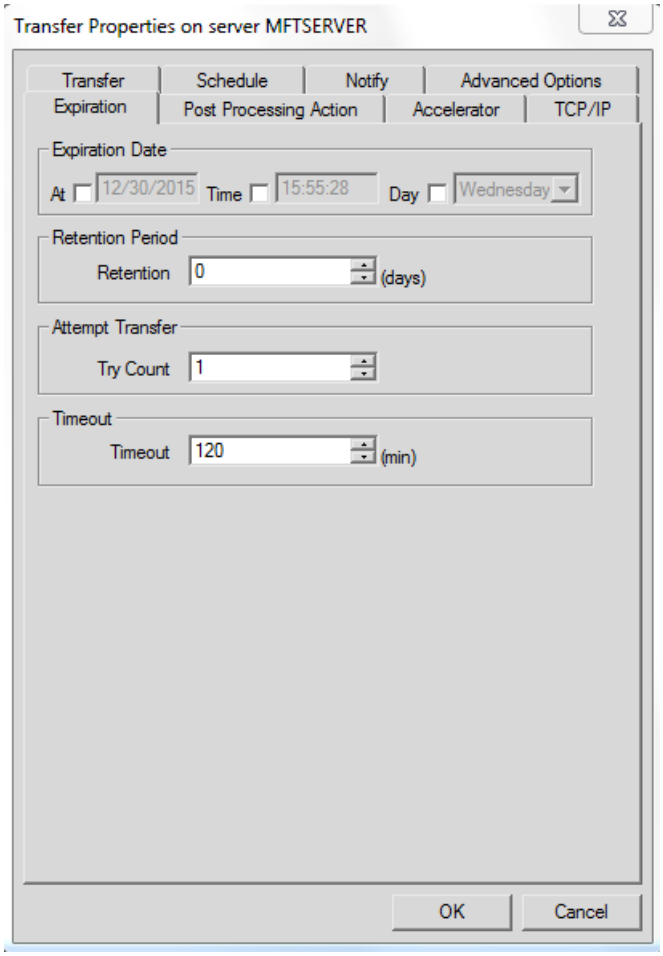
  LocalCTFile - This parameter contains the name of the file, which is used to translate on the local side.

  Note: When defining LocalCTFile, you must also define RemoteCTFile:NULL so that no translation takes place remotely.

  RemoteCTFile - This parameter contains the name of the file, which is used to translate on the remote side.

  Note: When defining RemoteCTFile, you must also define LocalCTFile:NULL so that no translation takes place locally.

1.1.5 Expiration Tab



- **Expiration Date** - Specifies the exact date and time when the system no longer tries a transfer. However, if this transfer is scheduled, that takes precedence over expiration. If expiration and retention are used, whichever value occurs first takes precedence.

At	This field specifies the date on which you want the transfer to expire. This defaults to approximately one month from the current date. This entry is mutually exclusive with the value in the Day (day of week) field.
Time	This field specifies a particular time at which you want the transfer to expire. This defaults to the current time.
Day	This field specifies a particular day of the week on which you want the transfer to expire. This entry is mutually exclusive with the value in the start At (date) field.

- **Retention Period** - Specifies the number of days that pass from the start of the transfer to the point it is no longer attempted. If expiration and retention are used, whichever value occurs first takes precedence.
- **Attempt Transfer: Try Count** - Specifies the number of times that TIBCO MFT Platform Server for Windows attempts the transfer. When the try count is reached, TIBCO MFT Platform Server for Windows no longer attempts the transfer. The default value for the try count is 1 when the panel is first opened. The max number is 9998. The value 0 represents “Unlimited” feature, which is actually 9999 tries.

Note: If a transfer fails because of a severe error, it will not be retried. See the following information regarding the return codes:

0 = Success
4 = Network error or retry-able file error that has exceeded try count.
8 = Permanent error. The transfer will not be retried.
15 = Directory transfer: no files to send or receive.
255 = Invalid Command Line Interface (CLI) parameter.

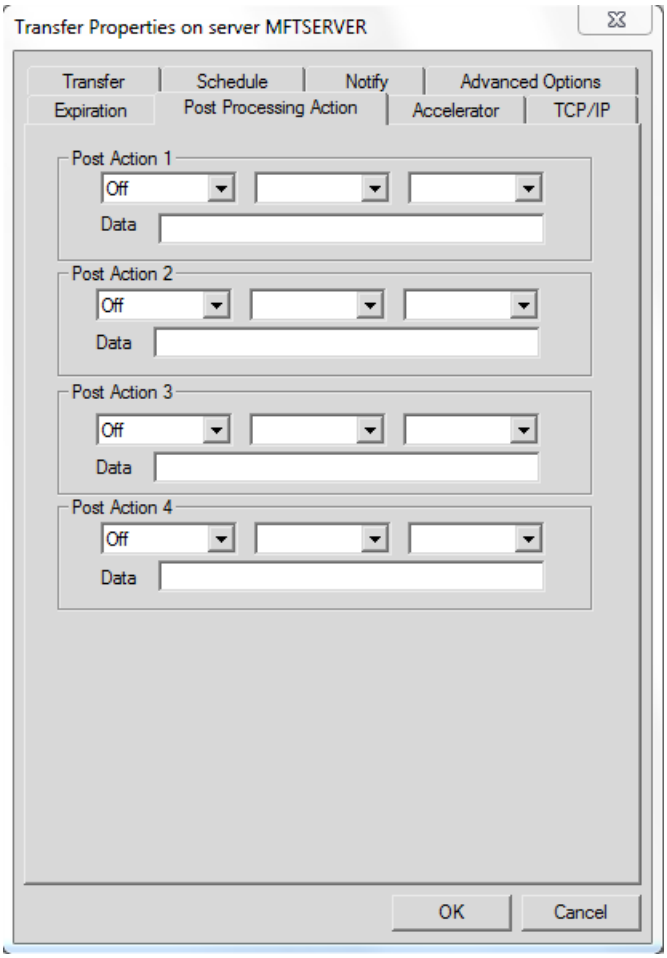
For any other return codes, check the system code on the operating system where the transfer is executed.

- **Timeout -** Specifies the amount of time (minutes) a connection stays open while waiting for a response from the remote side. When the time is reached, the connection is ended.

  Note: This parameter takes precedence over the **Initiator Timeout** field on the Server Properties page.

1.1.6 Post Processing Action Tab

Post Processing Actions are commands that are executed upon the completion of a transfer. This command can be defined up to four times. If the remote system is a z/OS mainframe, CALLJCL, CALLPGM, and SUBMIT are also supported in place of COMMAND. For more information on the CALLJCL, CALLPGM, and SUBMIT commands, see the TIBCO Managed File Transfer Platform Server for z/OS documentation.



- **Post Action 1** - This is a command (.bat, .com, .exe, and so on) that is executed upon the completion of the transfer.

  Field 1 - The values for this field are Off, Success, or Failure. Post Action 1 is executed based on the completion status of the transfer.

  Field 2 - The values for this field are Local or Remote. Post Action 1 is executed based on the source of the file transfer.

  Field 3 - The values for this field are Command, Call Program, Call JCL, and Submit. This is the type of action that is executed.

  Data - Defines the file that is executed. Up to 2565 bytes of data can be entered.

  Append a number sign (#) to the end of the data entered to have TIBCO MFT Platform Server for Windows launch the PPA and wait for the return code of the action. Append an ampersand (&) to the end of the data entered to have TIBCO MFT Platform Server for

Windows launch the PPA and not wait for the action to finish. The default behavior is the same as appending an ampersand (&) sign to the data entered.

Note: Do not insert number sign (#) or ampersand (&) at the end of a PPA executed on z/OS because z/OS does not support these parameters.

For example: C:\MyAction1.exe arg1=true #

C:\MyAction2.exe arg1=false &

The definitions listed above for Post Action 1 are the same for all four Post Actions.

1.1.6.1 Substitutable Parameters

MFT Platform Server supports substitutable parameters with which you can take full advantage of the 256 character maximum on the command data, and not have to copy the file name from the LocalFileName or RemoteFileName parameter. Note that file name tokens within PPA are not supported, because they are relatively long and the substitutable parameters conserve as many bytes as possible within the PPA action data field.

The PPA Substitutable fields use the percent sign (%) as the escape character instead of the dollar sign ($) that tokens use. The following table lists the substitutable parameters that are supported.

For this example, assume that the file is called: C:\a\b\c\d\config.txt.

Substitutable Parameter	Description	Resolved Name Example
%DIR	Remote file name directory without file name or drive	a\b\c\d
%DRIVE	Remote file name drive	C
%NODRIVE	File name without drive	a\b\c\d\config.txt
%SDIR	The lowest level directory	d
%HDIR	The high level directory	a
%NOSDIR	Directory name without the lowest directory	a\b\c
%NOHDIR	Directory name without the high level directory	b\c\d
%FILE	File name without directory	config.txt
%LFILE	File name with directory	C:\a\b\c\d\config.txt
%LLQ	Low level qualifier of file (data after last period (.))	txt
%HLQ	High level qualifier of file	config
%TRN	Transaction number	I824500001
%PROC	Process name	ABC123
%UDATA	User data	USRDATAABC123
%JDATE	Julian date (*YYDDD*)	05236
%JDATEC	Julian date with Century (*CCYYDDD*)	2005236
%TIME	Time (*HHMMSS*)	165030
%GDATE	Gregorian date (*YYMMDD*)	050824
%GDATEC	Gregorian date with century (*CCYYMMDD*)	20050824

There can be multiple PPA parameters within a single PPA data field. Each substitutable parameter must be processed one at a time before going onto the next byte of PPA data.

Some fields do not make sense such as %DRIVE in a UNIX environment. If a field does not make sense in the environment where PPA is being used, then the substitutable data is the text in the

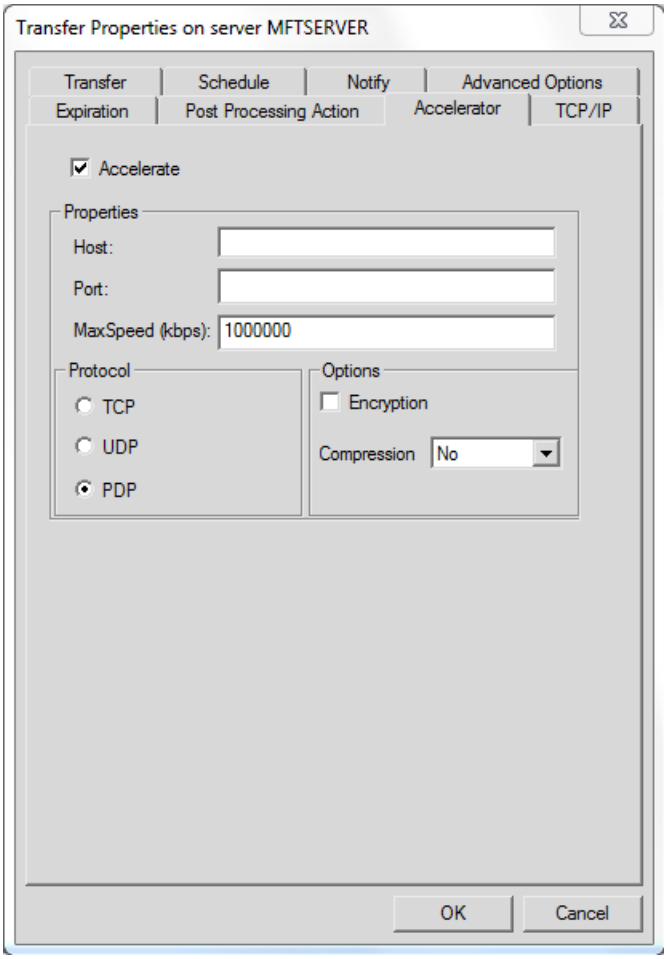
name of the parameter without the percent sign (%). If UNIX detects the %DRIVE parameter, the value DRIVE is used as substitution. Similarly, %PROC becomes PROC and %UDATA becomes UDATA if not interacting with a z/OS system.

When a parameter, such as a file name or file path, contains embedded spaces, you have to enclose the parameter in double or single quotation marks.

1.1.7 Accelerator Tab

If you want to set transfer requests to be sent by using the TIBCO Accelerator protocols of UDP (User Datagram Protocol), PDP (Parallel Delivery Protocol), or TCP, you can enable it by selecting the Accelerate check box. The properties section becomes available for you to configure the TIBCO Accelerator host and port your transfer request is sent to.

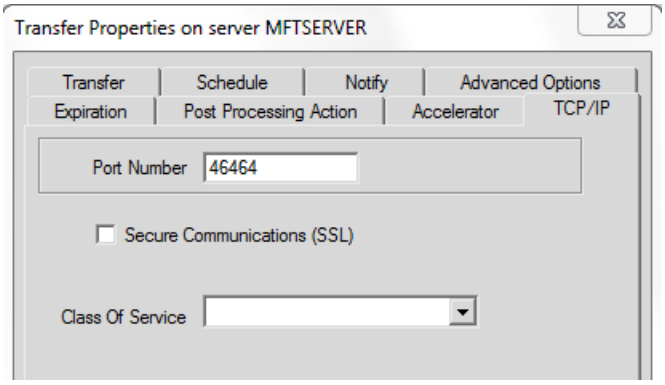
Note: The standard Accelerator port to use is 9099. Do not use another port unless instructed by your local administrator.



You can also configure the Accelerator host to use, Encryption (Blowfish), Compression (Best, Default, Fast) (This is a proprietary compression compatible with ZLIB), or a Max Speed in kilobytes per second your transfer request use.

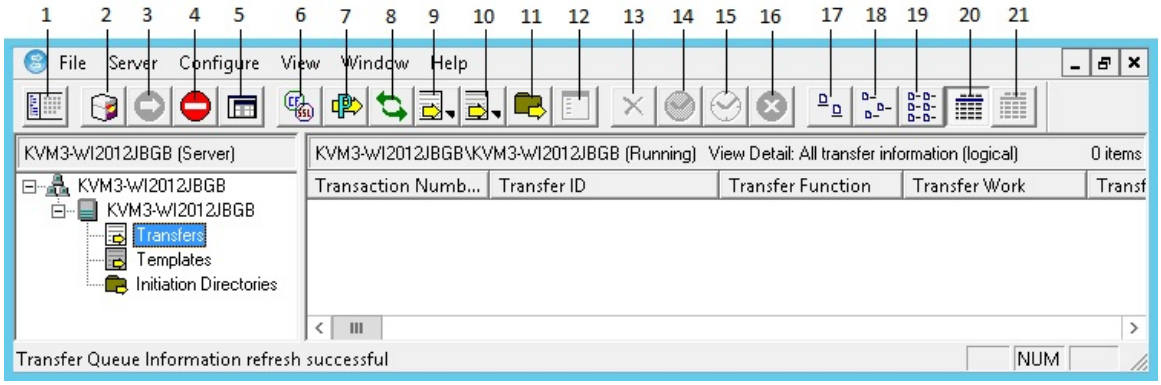
Note: Do not use MFT Platform Server compression or encryption with the TIBCO Accelerator compression or encryption. Use one or the other.

1.1.8 TCP/IP Tab



- **Port Number** - This is the TCP/IP port that the destination platform server is listening on. In TCP/IP networks, applications choose a specific port number for transactions so they do not conflict with other applications at the same TCP/IP address. By default, TIBCO MFT Platform Server for Windows uses 46464.
- **Secure Communications (SSL) -** This check box is selected when SSL is to be used. You must also enter the SSL port number in the Port Number field if SSL communication is used. For more information, see SSL.
- **Class Of Service -** Select the sockets buffer size used for the initiator in the TCP/IP transfer. The valid values are the different levels of buffer sizes specified in the configuration file named cfcos.cfg. The transfer performance can be improved as responder and initiator using different level of classes. For more information on setting the class of service for the responder, see Responder Property Page.

1.2 The Network View



Use the buttons along the top row to perform your tasks. From left to right, you can use the buttons to perform the following operations:

1. Create a new network view
2. Add a server to the list
3. Start an MFT Platform Server
4. Stop an MFT Platform Server
5. View/Change Server Properties
6. SSL Settings
7. View/Change Configured Post Processing
8. Refresh view
9. Create a new transfer
10. Create a new transfer template
11. Create a new directory named initiation entry
12. View/Change selected object properties (transfers, templates, and DNI)
13. Delete selected objects (transfers, templates, and DNI)
14. Hold (transfers and DNI)
15. Release (transfers and DNI)
16. Abort (transfers)
17. View items in large icons
18. View items in small icons
19. View items in a list
20. View items in detail
21. Change the detail view fields

You can use the menu to perform the same tasks as the buttons. This Guide describes the Administrator's functionality in terms of the buttons.

1.2.1 Buttons

1.2.1.1 Create a New Network View Button

 Click this button to create a new window to view server and transfer information.

1.2.1.2 Add a New Server Button

 Click this button to add a server to the Network window.

1.2.1.3 Start an MFT Platform Server Button

 Click this button to start a server.

1.2.1.4 Stop an MFT Platform Server Button

 Click this button to stop a server.

1.2.1.5 View/Change Server Properties Button

 Click this button to display the MFT Platform Server Properties panel, which displays configuration information about the selected server. When the panel is invoked, a query is issued to the server for the current settings which are returned and displayed in the panel. From this panel, you can modify the information. If you do not have permission to start and stop the MFT Platform Server service, you cannot modify the information on the Server Properties (the panel is displayed as Read Only).

1.2.1.6 SSL Properties Button

 Click this button to display the MFT Platform Server SSL Settings panel. From this panel you can also modify the current SSL settings.

1.2.1.7 Configure Post Processing Button

 This allows you to view and change the Configure Post Processing feature. Select the check box to turn on this feature and enter the name of the file to be used for the post processing.

1.2.1.8 Refresh View Button

 Click this button to view the current server and transfer information.

1.2.1.9 New Transfer Button

 Click this button to add a new transfer to the queue of the server that you are viewing. When selected, the Transfer Properties panel is displayed. On this panel, you specify all of the particulars of the file transfer that you want to add to the queue.

1.2.1.10 New Template Button

 Click this button to create a new transfer template. See the Transfer Templates section for details.

1.2.1.11 New Initiation Directory Button

 Click this button to create a new Directory Named Initiation entry. See the DNI section for details.

1.2.1.12 Update Properties Button

 Click this button to view or change the parameters of a specific transfer, template, or Directory Named Initiation entry. The Properties panel is displayed. Modify the fields and select OK.

If the job is active at the time of modification and it is scheduled to execute only one time, your modifications are denied. If the job is active and scheduled to execute more than once, your modifications takes effect the next time the transfer becomes eligible. If the job is scheduled and it has not yet been executed, your modifications become effective immediately.

Any significant changes made to the platform server queue view are logged to the event log (see the The Event Log appendix).

1.2.1.13 Delete Selected Objects Button

 Click this button to remove a non-active transfer, template, or Directory Named Initiation template.

1.2.1.14 Hold Button

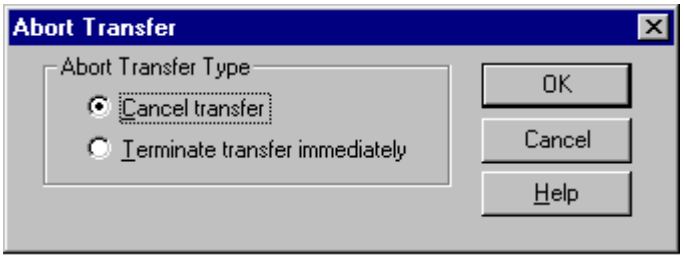
 Click this button to put a hold request on a transfer or Directory Named Initiation entry so that it cannot be dispatched. This action prevents the Schedule Dispatch Service from initiating the transfer until otherwise notified.

1.2.1.15 Release Button

 Click this button to release a held transfer or Directory Named Initiation entry.

1.2.1.16 Abort Button

 Click this button to abort a transfer. The Abort Transfer panel is displayed.



Select one of the two options. TIBCO MFT Platform Server for Windows prompts you to confirm your selection. Upon confirmation, the program issues the abort command for each of the transfers selected.

“Cancel transfer” notifies the remote system that the transfer has been terminated.

“Terminate transfer immediately” terminates the transfer and does not notify the partner. In certain instances, this selection can stop a transfer that “Cancel transfer” cannot.

1.2.1.17 View Items as Large Icons Button

 This button causes a single large icon for each file transfer to display with its Transfer ID directly below it. The icon’s appearance depends on the file transfer type selected.

1.2.1.18 View Items as Small Icons Button

 This button causes a single small icon for each file transfer to display with its Transfer ID directly next to it. The icon's appearance depends on the file transfer type selected.

1.2.1.19 View Items in a List Button

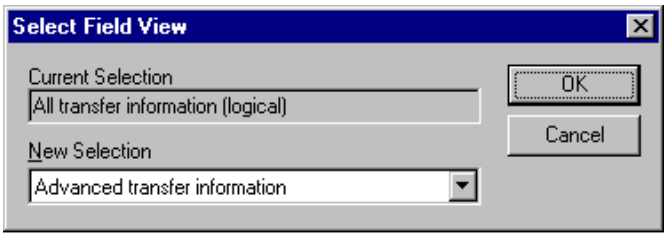
 This button causes a single small icon for each file transfer to display with its Transfer ID directly next to it. The way the icon is displayed differs depending upon the file transfer type selected.

1.2.1.20 View Items in Detail Button

 Click this button to view detailed information about the transfers in the Queue view. The fields are displayed according to your selection in the Select Field View panel (see Select Field View Button). By default, all fields on the queue are displayed.

1.2.1.21 Select Field View Button

 Click this button to select which fields you want to view from a predefined group. The panel is displayed as shown in the following figure.



- **Current Selection**

  This field names the predefined group you are viewing.

- **New Selection**

  Use this field to change the predefined group. These groups are defined as shown in the following table.

All (alphabetical)	All fields on the queue are arranged alphabetically.
All (logical)	All fields are displayed in a logical sequence.
General	A short list of fields that are common to all file transfers are displayed.
File to File	Only those fields relative to File to File transfers are displayed.
File to Print	Only those fields relative to File to Print transfers are displayed.
File to Job	Only those fields relative to File to Job transfers are displayed.
Remote Command	Only those fields relative to Remote Command executions are displayed.
Advanced	Only those fields that are displayed on the Advanced panel are displayed.
Status	Only those fields that are seen in the Initiated Transfers window are

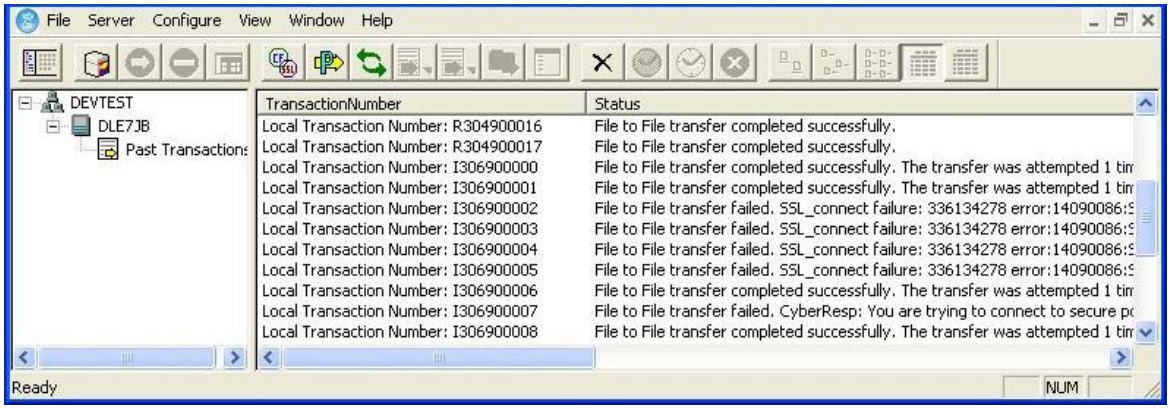
	displayed.
Schedule	Only those fields that are displayed on the Schedule panel are displayed.
z/OS Parameters	Only those fields that are displayed on the z/OS options panel are displayed.
TCP/IP Transfer	Only those fields that are TCP/IP specific are displayed.

1.2.2 Past Transactions

The MFT Platform Server Administrator Past Transactions feature supports you to see the status of previously completed transfers.

You can select a server, click View from the menu, and then select Past Transactions. You must add the server to view status of the previously completed transfers on that server. This is exactly the same as adding a server in a network view in the Administrator.

The transfers in this particular window cannot be viewed by double-clicking the transfer. The status of these transfers is pulled from the Event log of the respective server. Therefore, if you clear your event log, the past transactions are also deleted.



A backup event log on the server is created every time you open or refresh the Past Transactions panel. This backup is in a file called c:\temp\tmp.evt.

Note: The C:\temp directory must already exist on the system. If the directory does not exist, the backup event file will not be created.

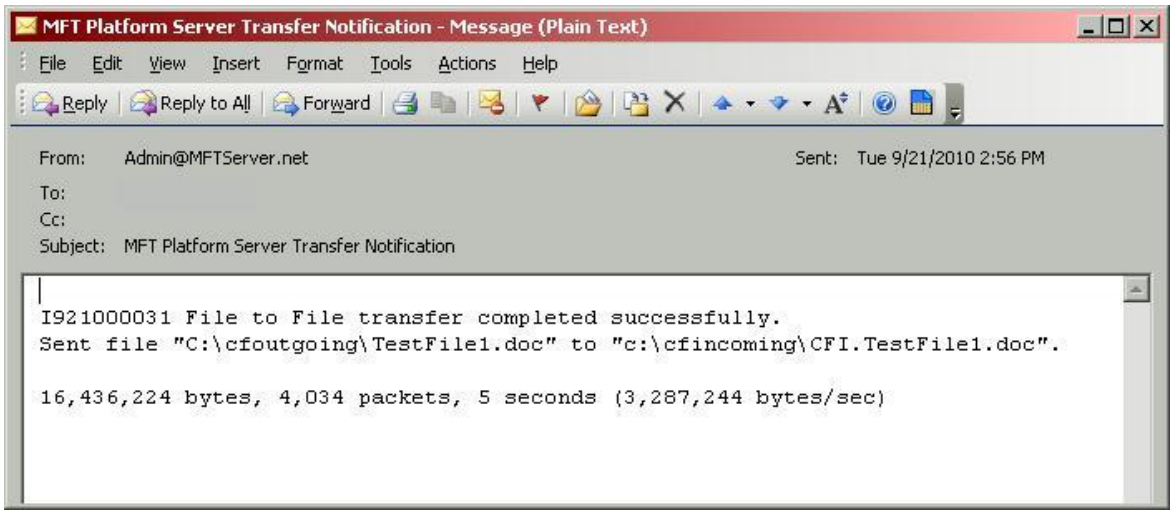
You can read from the backup event log by selecting **Open Backup Eventlog** from the **File** drop-down menu in the Administrator. You must select the server to enable this menu item. You can delete this file but in that case no transaction is available when you open the backup event log. You can sort the transactions by clicking any column header.

1.2.3 Notification

After a file transfer has been submitted and executed, an email can be received about the transfer's success or failure on both the remote and local systems.

1.2.3.1 MFT Platform Server Email Notification

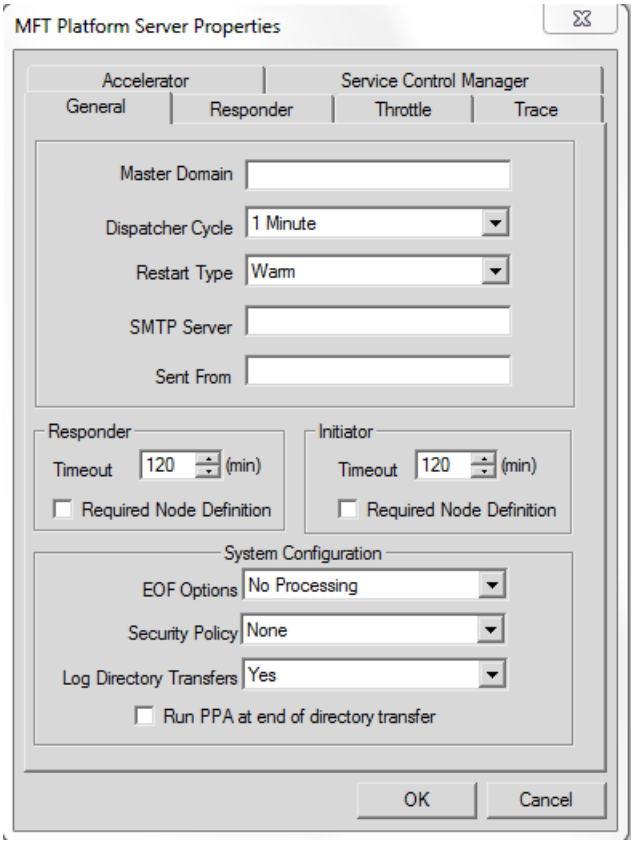
Upon completion of the file transfer, the platform server sends an email to the address specified in the Notify tab of the Transfer Properties page. For more information, see Notify Tab. The following figure shows a sample email:



1.3 Server Properties

You can go to the Server Properties page by selecting a server and then selecting Properties from the Server menu.

1.3.1 General Properties Page



Note: If you can view the server status but do not have the authority to start or stop the server, the connection status is set to "ServerName - Connect Query Only." You cannot change the server settings on this panel. The MFT Platform Server Properties panel is displayed as Read-Only. The OK button and the Cancel button are replaced by a Close button.

Parameter	Description
Master Domain	In this field, enter the name of the domain that you want to use as the default domain for verifying security rights when your server is acting as a responder. This means if you have defined the master domain, a remote user only have to define the user ID without a domain name preceding it in the remote identification transfer information.
Dispatcher Cycle	This parameter specifies the time that the scheduled dispatcher service waits before it checks for transfers that must be started or restarted. The selectable values in this field are 10 seconds, 30 seconds, 1minute, 2 minutes, 3 minutes, 4 minutes, 5 minutes, 7 minutes, 10 minutes, 15 minutes, 30 minutes, 45 minutes, 1hr., 2hr., 4hr., 8hr., 12hr., and 24hr. The scheduled dispatcher service writes the date and time to the MFT

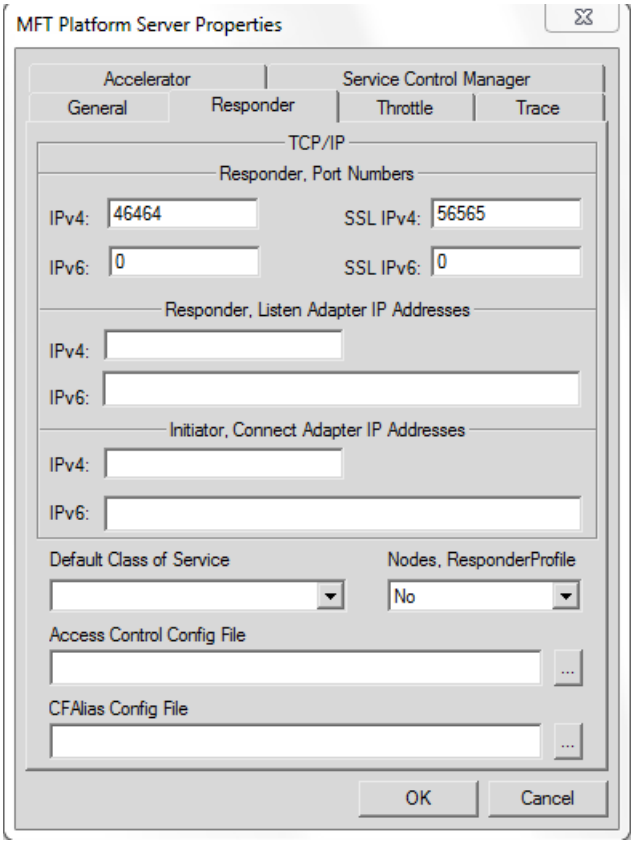
	Platform Server Monitor when it checks schedules for eligibility.
Restart Type	**Warm** - All transfers that exist in the persistent work queue are retained when MFT Platform Server is restarted. **Cold** - All transfers that exist in the persistent work queue are deleted when MFT Platform Server is restarted. The old PQF is overwritten by a new PQF. **Note**: A cold start deletes your DNI definitions as well as any queued or active transfers. After performing a cold restart, the restart type will be reset to Warm. The cold option will not be saved at the next restart.

SMTP Server	The IP name or address of the email server that is used to send out email notification. If you change the value in this field, you must also stop and start the MFT Platform Server service to pick up the new value.
	You can also specify the port to be used in the following format:
	IPv4:
	`hostname:port`
	`IP_address:port`
	IPv6:
	`[host]:port`
	If no port is specified, the default SMPT port 25 is used.

Sent From	This field identifies the name displayed in the email notification. This value must be an email address and cannot contain any spaces.
Responder: Timeout	This field specifies the amount of time (minutes) a connection stays open while waiting for a response from either the Initiator or the Responder. When the time is reached, the connection is ended. The valid values are from 0 to 1440. 0 indicates using the default value 120 minutes.
Responder: Required Node Definition	Required Node Definition supports restricting the remote systems that the responder accepts transfer requests from. To be accepted, incoming requests must come from remote systems that are defined in the cfnode.cfg file. All others are refused. See Nodes, Profiles, and Distribution Lists for details.
Initiator: Timeout	This field specifies the amount of time (minutes) a connection stays open while waiting for a response from either the Initiator or the Responder. When the time is reached, the connection is ended. The valid values are from 0 to 1440. 0 indicates using the default value 120 minutes.
Initiator: Required Node Definition	Required Node Definition supports restricting the remote systems that the initiator can initiate transfer requests to. To be accepted, outgoing requests must go to remote systems that are defined in the cfnode.cfg file. All others are refused. See Nodes, Profiles, and Distribution Lists for details.
System Configurations: EOF Options	This field defines whether a Carriage Return Line Feed (CRLF), an End of File (EOF), or both will be added to the end of the transferred binary file. This parameter only works when CRLF=YES is specified in the transfer. If you define CRLF=NO, or set no permitted actions (no processing) along with CRLF=YES in the transfer, this field is ignored.
System Configurations: Security Policy	This parameter defines whether this platform server enforces HIPAA or FIPS-140 regulations on initiated and responding transfers.

	HIPAA – This setting requires the platform server to comply with HIPAA standards. At this time, the standards require that all files are transferred by using encryption key length that is 128 bits or greater. **FIPS-140** – This setting requires the platform server to comply with FIPS (Federal Information Processing Standard). This is a Government standard that certifies cryptographic modules used for the protection of information and communications in electronic commerce within a security system protecting sensitive but unclassified information. This requires that all files are transferred by using SSL with an encryption type of Rijndael (AES) which uses a key length of 256 bits. For more information on configuring SSL, see SSL. To comply with the security policies of HIPAA or FIPS-140, for transfer requests configured incorrectly, for example a transfer using an encryption type of DES which is not allowed for either HIPAA or FIPS-140, the encryption is overridden. And to comply with HIPAA, if you set your encryption to DES, a pop-up message is displayed informing you the encryption will be changed to 3DES; if you set the encryption to Blowfish, a pop-up message is displayed informing you the encryption will be changed to Blowfish Long. If you use FIPS-140, you receive a pop-up message informing you the encryption will be changed to Rijndael (AES) when a transfer is initiated.
Log Directory Transfers	This parameter defines whether to log cfdir requests when doing directory transfers. The cfdir program is the internal directory command to scan the remote folder. The cfdir program will read a directory to determine the files in that directory that can be transferred. The valid values are Y, N, or Errors. The default value is Y. Errors means the directory list request is logged only when an error occurs.
Run PPA at end of directory transfers	This parameter defines when doing a directory transfer or doing a transfer using a distribution list, if Post Processing Action(s) are configured, the PPA will only be run once at the end of the entire transaction instead of after every file that is transferred from the directory.

1.3.2 Responder Property Page



This page provides configuration of the responder port number and responder IP address.

- **TCP/IP Responder, Port Numbers**

  IPv4 and IPv6 - TIBCO MFT Platform Server for Windows responds to transfers by using TCP/IP which are routed to the IP address of the system where MFT Platform Server is installed. Subordinate to that address is the port number. The port number allows different applications to locate at the same IP address on the same machine, but makes them unique so that they can coexist.

  The default IP port number for the platform server IPv4 address is 46464, but you can change it to any number from 1025 to 65535.

  SSL IPv4 and IPv6 - This is the port number on which SSL is listening. The default for the SSL IPv4 IP port number for the platform server is 56565, but you can change it to any number from 1025 to 65535. For more information on SSL, see SSL.

- **Responder, Listen Adapter IP Addresses**

  IPv4 and IPv6 - If a machine has more than one network adapter, you can bind the connection to a particular adapter. It can guarantee that all transfers go only through this particular adapter. If this parameter is defined, the responder receives data for incoming connections only through this network adapter.

- **Initiator, Connect Adapter IP Addresses**

  IPv4 and IPv6 - If a machine has more than one network adapter, you can send all initiated transfer requests out on one particular adapter. It can guarantee that all transfers go out only

through this particular adapter. If this parameter is defined, the initiator sends data for outgoing connections only through this network adapter.

- **Default Class of Service**

  This defines the default Class of Service for Responder transfers. A list of all Class of Service entries is available in the drop-down box. By leaving this parameter blank, Class of Service will not be used and default TCP buffer sizes will be used for Responder transfers.

  For more information on setting the Class of Service for Initiator, see TCP/IP Tab.

- **Nodes, Responder Profiles**

  To have a Responder Profile, you must have a node defined. Responder Profiles define a local user name and password that are used in place of the incoming user name and password. By using responder profiles, a remote MFT Platform Server installation does not require an actual user name and password on your local machine to initiate a transfer. Setting the value to Yes means all connections to/from this installation require a node to be defined with a responder profile. If the setting is Dual, a responder profile is searched for first; if none is found, the system accepts the transaction and processes it with the IP/host name and the user ID associated with the transaction. The node responder parameter overrides this parameter unless the node is configured to use the Server Responder Profiles default setting.

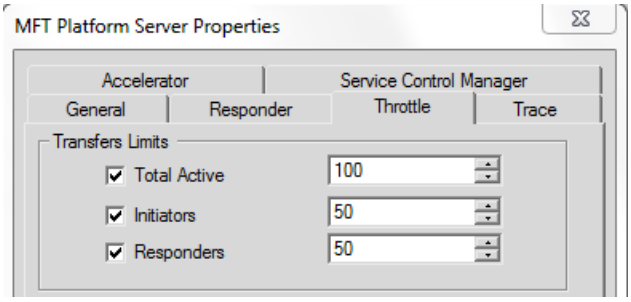
- **Access Control Config File**

  This allows you to select the Access Control Config file. Access Control allows the admin to force transfers to go to a predefined directory based on user-defined criteria. The default file name for the Access Control configuration is AccessControl.cfg. See Access Control for more information. This is used by the MFT Platform Server Responder only.

- **CFAlias Config File**

  This allows you to select the CFAlias Config file. CFAlias allows you to define aliases for transfer file names. The default file name for the CFAlias configuration is CfAlias.cfg. See CfAlias for more information. This is used by the MFT Platform Server Responder only.

1.3.3 Throttle Properties Page

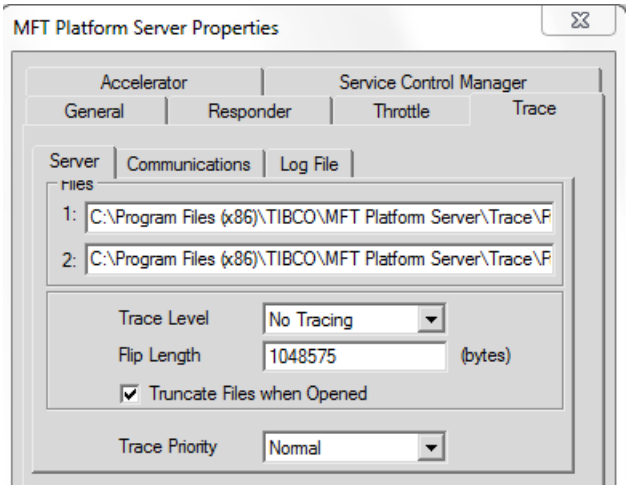


Server throttling limits user activity. You can use this property sheet to limit total active transfers, initiators, and responders.

- **Total Active** - This field indicates how many active transfers are allowed at any given time.
- **Initiators -** This field indicates how many Initiators only are allowed at any given time.
- **Responders -** This field indicates how many Responders only are allowed at any given time.

To limit DNI directories to be managed by the server, there is one setting for the entire server.

1.3.4 Trace Property Page



Use this property page to configure the tracing for the platform server.

Note: You should only turn on Tracing at the request of TIBCO Technical Support.

Each tab contains information for each trace file generated by the platform server.

Use the Server tab to trace the activities of the server, including actions related to performing file transfers, and managing transfers, DNI, and Template objects.

Use the Communications tab to trace the communications layer which is activated during transfers. The information contained in this trace file shows exactly what is being transmitted and received across the network during a transfer. Performing traces of large high speed transfers can cause severe performance degradation.

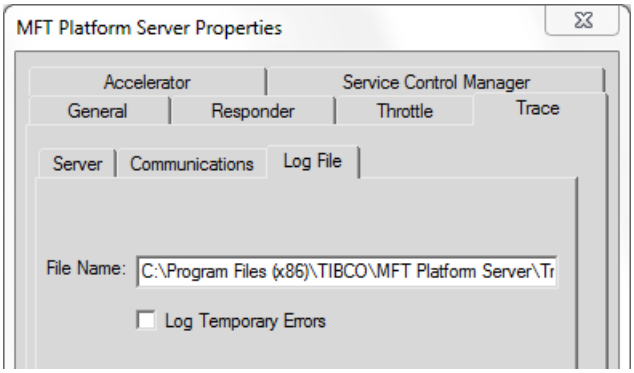
Use the Log File tab for web administration. With this, you can view past transactions through MFT Command Center.

- **Trace File 1** - This field indicates which file to use for the first flip file.
- **Trace File 2** - This field indicates which file to use for the second flip file.
- **Trace Level** - This field indicates the amount of information that is reported to the trace file. The value is directly proportional to the amount of information written to the trace files. You can only use Tracing to troubleshoot a problem and turn on Diagnostic Level 3 at the request of TIBCO Technical Support.
- **Flip Length** - This is the maximum amount of information (in bytes) that is written before the trace files flip. This value cannot be less than 1024.
- **Truncate Files when Opened** - When the application or server starts, it can clear out (truncate) the trace files before it begins to write information. If this option is TRUE, the trace files are truncated when the program starts. Otherwise, the application or server opens the existing files and appends the information to the end.
- **Trace Priority** - While the fields described above apply separately to each trace file, this field applies to all of the trace files at the same time.

  This field indicates the priority given to the thread that is responsible for receiving and formatting the trace information from the system. Increase this value if the trace information

generated by the system exceeds the system's ability to write the information to the trace files. You can only turn on tracing at the request of TIBCO Technical Support.

1.3.4.1 Log File Tab



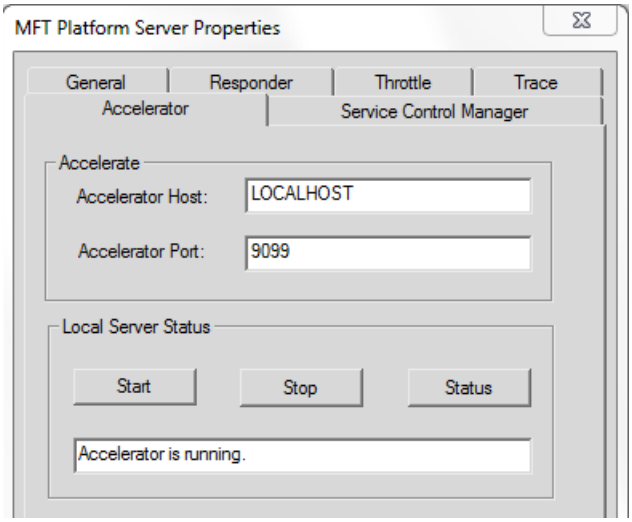
- **File Name**

  Enter the path name for the Log file to which the information is written. This file can be accessed when inquiring on transactions by using the cfinq utility as well as by MFT Command Center.

- **Log Temporary Errors**

  Select this check box to set Log All Transfer Attempts to on. Setting this to off (clearing the check box) causes the platform server to log only the final transfer attempt in a restart situation.

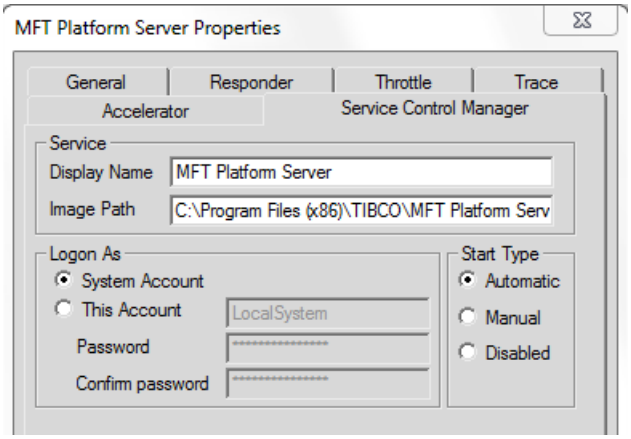
1.3.5 Accelerator



The Accelerator property page is used to maintain the configuration of the Accelerator service (RsTunnel.exe). You can stop and start the Accelerator service from this location. If you edit the Accelerator Host or Port, you must restart the Accelerator service for the new settings to be taken.

- **Accelerator Host** - This is the host name or IP of the host.
- **Accelerator Port** - This is the port number the Accelerator is listening on. The default port is 9099.
- **Local Server Status** - Through this section, you can start and stop the RSTunnel service as well as display the current status of the service.

1.3.6 Service Control Manager Property Page



Use the Service Control Manager property page to maintain the configuration of the TIBCO MFT Platform Server for Windows service in the Windows Service Control Manager. TIBCO MFT Platform Server for Windows operates as a Windows server (on Windows); therefore, through this panel, you can maintain both types of service.

- **Display Name** -This describes the service shown in the Windows Service Control utilities.
- **Image Path -** This is the full name to the executable file for the service. For MFT Platform Servers, "….\ftmssvr.exe" is displayed.
- **Logon As -** This section defines the user ID that is used by the Windows service. You can choose one of the following two options:

  System Account: the Windows service will use the local systems account. When this is selected, you might not be able to use UNC paths as part of the DNI Initiation Directories definition.

  This Account: you can define credentials that will be used by the Windows Service.

- **Start Type** - This section indicates how the service is started.

  Automatic: starts when the system reboots (suggested setting).

  Manual: starts when the administrator tells it to start.

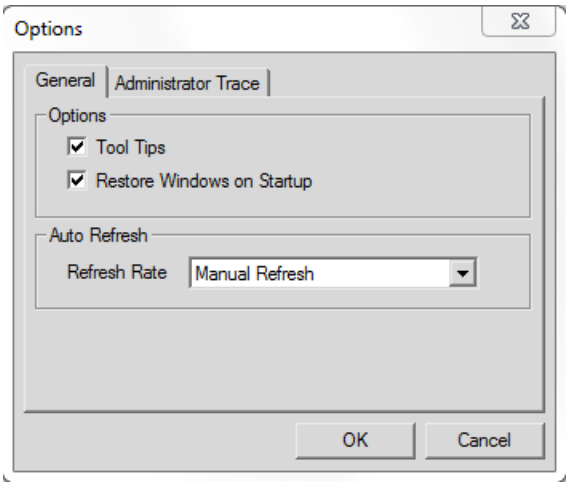
  Disabled: prevents the service from ever starting.

  After you modify any of the MFT Platform Server Properties panels, click OK. A dialog panel indicates whether or not the change is effective. Upon receipt of the change request, the server makes the parameter changes and then writes the parameters to the registry entries of the server.

1.4 View Menu: Options Property Sheet

The Options property sheet is available from the MFT Platform Server Administrator View menu. You can also access this property sheet by right-clicking in an un-written window space.

1.4.1 General Property Page



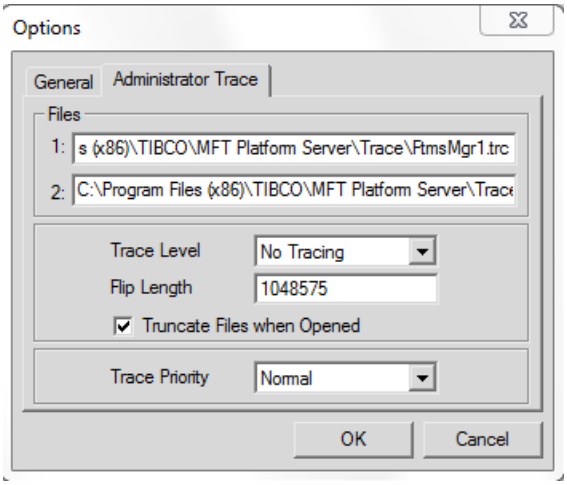
- **Tool Tips** - Select this check box to view the TIBCO MFT Platform Server for Windows tool tips when you start TIBCO MFT Platform Server for Windows.
- **Restore Windows on Startup -** Select this check box to restore the windows settings when you start TIBCO MFT Platform Server for Windows. This check box is selected by default.
- **Refresh Rate -** The administrator can automatically refresh the information it displays. This field indicates how often the refresh occurs.

  The available options are as follows:

  - Manual Refresh - you must select the Refresh command to update the view.
  - 5 Seconds - the refresh occurs every 5 seconds.
  - 10 Seconds - the refresh occurs every 10 seconds.
  - 20 Seconds - the refresh occurs every 20 seconds.
  - 30 Seconds - the refresh occurs every 30 seconds.
  - 60 Seconds - the refresh occurs every 60 seconds.
  - 2 Minutes - the refresh occurs every 2 minutes.
  - 5 Minutes - the refresh occurs every 5 minutes.
  - 10 Minutes - the refresh occurs every 10 minutes.
  - 30 Minutes - the refresh occurs every 30 minutes.
  - 60 Minutes - the refresh occurs every 60 minutes.

When MFT Platform Server Administrator is opened, a network view with the local server is added automatically.

1.4.2 Administrator Trace Property Page



Use this property page to configure the tracing for the MFT Platform Server Administrator application.

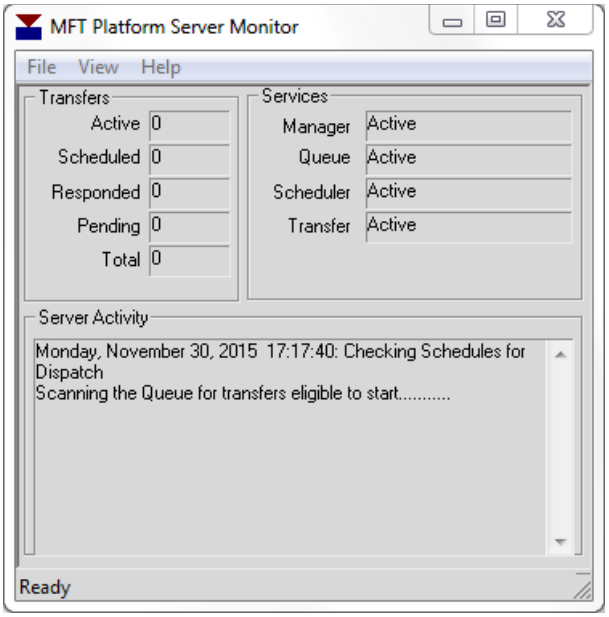
Note: This panel is for configuring MFT Platform Server Administrator locally. You can only turn on Tracing at the request of TIBCO Technical Support.

- **Trace File 1** - This field indicates which file to use for the first flip file.
- **Trace File 2 -** This field indicates which file to use for the second flip file.
- **Trace Level -** This field indicates the amount of information that is reported to the trace file. The value is directly proportional to the amount of information written to the trace files. You can only use Tracing to troubleshoot a problem and turn on Diagnostic Level 3 at the request of TIBCO Technical Support.
- **Flip Length -** This is the maximum amount of information (in bytes) that is written before the trace files flip. This value cannot be less than 1024.
- **Truncate Files when Opened -** When the application opens, it can clear out (truncate) the trace files before it begins to write information. If this option is TRUE, the trace files are truncated when the program starts. Otherwise, the application opens the existing files and appends the information to the end.
- **Trace Priority -** This field indicates the priority given to the thread responsible for receiving and formatting the trace information from the system. Increase this value if the trace information generated by the system exceeds the system's ability to write the information to the trace files. You can only turn on Tracing at the request of TIBCO Technical Support.

2

2. MFT Platform Server Monitor

Use MFT Platform Server Monitor to view all activities that MFT Platform Server is performing on the server on which it is running. Here, you cannot enter any information or change any values. The MFT Platform Server Monitor panel contains the following three sections: Transfers, Services, and Server Activity.



- **Transfers -** Displays the number of transfers that are present on a particular server queue.
- **Services -** Displays the status of each service available on a selected server.
- **Server Activity -** Displays all of the actions that the selected server performs.

2.1 Functions

From the View menu of the MFT Platform Server Monitor panel, you can choose the following options:

- **Status Bar** - Show or hide the system status bar at the bottom of the window. Hide the status bar to provide more desktop area for viewing information in the Server Activity display.
- **Always On Top** - Indicates that the window is always on top of the desktop. With the window always on top, you can view the status of the local MFT Platform Server at a glance while continuing to work in other applications.
- **Hide When Minimized** - Directs the program to hide itself and remove its icon from the task bar when you minimize the window. You can save space on the task bar when the window is not being viewed. To restore the window, double-click the Monitor icon on the system tray.
- **Clear Display** - Clear the information from the Server Activity display.

3

3. Command Line Interface

Through the Command Line Interface, you can produce clear and readable batch programs by using parameters created for all of the MFT Platform Server functions.

Therefore, the following methods for specifying parameters on the command line are supported:

- Environment variables
- Short (1 or 2 characters) command line parameters
- Long command line parameters
- Environment variables on the command line

In the GUI panels, the values of the previous transfer are saved in the Registry and used as the default values for the next transaction. However, values that are used for a transaction in the command line program are not saved in the Registry.

Note: The environment variable setting stays active until you change it or remove it by using the SET command with no value specified.

3.1 Command Line Format

The following example shows the format of a simple transfer from the command line.

```
FTMSCMD /SEND /FILE [parameters] “c:\local\file\name.txt”
“remote.file.name”
```

In this example, no environment variables are used. Therefore, you must specify the mandatory parameters in the parameter section.

The following example shows the format of a simple transfer from the command line that uses environment variables.

```
SET HOST_NAME=hostname

SET CR_LF=no

SET REMOTE_USER_ID=userid

FTMSCMD /SEND /FILE [parameters] “c:\local\file\name.txt”
“remote.file.name”
```

Here, the mandatory parameters are specified in the environment variables. You do not have to specify parameters in the parameter section; however, you can still specify any of the additional parameters in the parameter section or in the environment variables.

3.1.1 Specifying Command Line Parameters

To set a command line argument, use the following syntax:

```
FTMSCMD /send | /recv /DS:dest [options] local_file remote_file
```

Options can include any number of the following forms:

1. Options are indicated by a forward slash (/) or hyphen (-) followed by the option. Some options need arguments, some do not. Forward slash (/) is provided for users who like the DOS standard. Hyphen (-) is provided for users who like the UNIX standard.

   `/option` (DOS Standard)
   `-option` (UNIX Standard)

2. When an option requires an argument, it is separated from the option name by a colon (:) or an equal sign (=), as the following examples illustrate.

   ```
   /option_name:option_value
   -option_name:option_value
   /option_name=option_value
   -option_name=option_value
   ```

Use `FTMSCMD /?` command to get a list of all arguments. For more information on the available parameters, see Command Line Parameters.

3.2 File to File Transfers

To have the output of the transfer written to a file in the destination, you must specify the positional parameter /F.

```
FTMSCMD /send | /recv /DS:dest /F /[parameters]  “local_file_name” “remote_file_name”
```

File to file transfers can be performed in either direction: you can receive a file from a remote system or send a file to a remote system. The file name can be either local or remote, depending on transfer directions.

File Name	Description
Local File Name	The name of the file on the local system that is involved in the transfer.
Remote File Name	The remote file name on the remote system. It can be any combination of up to 255 characters. If the name contains embedded spaces or commas, specify the name in single quotation marks. If the remote system is z/OS, only the first 54 characters are significant.

Example of Sending a File to a Remote System

```
FTMSCMD /S /F /DS:HOSTNAME  /PORT:46464 /DT=BINARY /RI=USERID  /RW=pswd “F:\JOHN\QA\ONEX1.BIN”  “JTPLM.QAL.BATCHB.ONEX1”
```

Example of Receiving a File from a Remote System

```
FTMSCMD /R /F /DS:HOSTNAME /PORT:46464 /DT=ASCII /RL=1  /RI=USERID /RW=PSWD “F:\JOHN\QA\ONEX4.TXT”  “hlq.QA.FILE.FB.ONEX4”
```

3.3 File to Job Transfers

To have the output of a transfer executed as a job, you must specify the positional parameter /JOB.

```
FTMSCMD /send | /recv /DS:dest /JOB /[parameters] file_name
```

File to job transfers can be performed in either direction. You can receive a file from a remote system and run it locally, or send a file to a remote system and run it remotely. The file name can be either local or remote, depending on transfer directions.

For example, if you are receiving a file from the remote side and having it executed on the local system, you must specify *file name* in the example above as the name of the remote file. You do not have to specify a local file name because the output will not be written to any local file. If you are sending a file to the remote side and having it executed on the remote system, specify the *file name* in the preceding example as the local file name.

Example of Sending a Job to a Remote System

```
FTMSCMD /S /JOB /DS:HOSTNAME /DT=E /CR=YES /RI=USERID C:\JOHN\IEBCOPY
```

Example of Receiving a Job from a Remote System

```
FTMSCMD /R /JOB /DS:HOSTNAME /CR=YES /RI=USERID HLQ.TEST.JOB
```

The destination (host name or address) must be set when performing a transfer.

Note: When receiving a file to be executed as a job on a Windows system, the job is executed in the \WINDOWS\SYSTEM32 directory. Remember that when writing your batch jobs, you must change the directory in which the batch job is executed.

3.4 File to Print Transfers

To print the output of the transfer to the destination printer, you must specify the positional parameter /PRINT or /P.

```
FTMSCMD [parameters] /SEND /PRINT /RemotePrinterName=printer_name
file_name
```

This example illustrates a file transfer whose output is directed to a printer. Because the transfer can be in either direction (receiving a file from the remote side and printing it on a local printer, or sending a file and printing it on the remote side), the file name depends on the direction in which the transfer occurs.

For example, if you are receiving a file from the remote side and printing it to a local printer, you must specify *file name* in the example above as the remote file name. You do not have to specify a local file name because the output will not be written to any local file.

If you are sending a file to the remote side and printing it to a remote printer, you must specify *file name* in the example above as the local file name.

```
SET PRINTER_NAME=printer_name

FTMSCMD [parameters] file_name
```

In this example, the mandatory parameters are specified in the environment variables. You do not have to specify any parameters in the parameter section; however, you can specify any of the additional parameters in the parameter section or in the environment variables.

3.4.1 Specifying the Printer Name

To specify a LAN printer, use the UNC for that device. To specify a printer name by using UNC, precede the computer name with two backslashes (\\) and separate the computer name from the shared printer's name with a single backslash (\). For example:

```
\\SERVER1\HP_LASERJET_QUEUE
FTMSCMD [parameters] /RECEIVE /PRINT
/RemotePrinterName=\\SERVER1\HP_LASERJET_QUEUE file_name
```

To specify a z/OS printer, type $SYSOUT@, where @ is the class to which you want to send the output.

```
FTMSCMD [parameters] /SEND /PRINT /REMOTE_PRINTER_NAME=$SYSOUT@
file_name
```

3.5 Remote Command Transfers

To execute a command on a remote system, you must specify both the type of command and the actual command to execute.

If the remote system is a Window or UNIX system, the parameter is /RC or /RemoteCommand. For a z/OS system, several options (/E, /EXEC, /RE, and /REXXEXEC) are all acceptable for an executable file. /SJ and /SUBJCL are used for submitting job control language; /CJ and /CALLJCL are used for calling programs with JCL linkage; and /CPG and /CALLPGM are used to call a program with standard linkage. Each of these must be followed by the command to be executed.

To have a command be executed remotely, specify the positional parameter /COMMAND followed by the option and command to be executed.

```
FTMSCMD /SEND [parameters] /COMMAND /RemoteCommand: command_to_execute
local_file_name
```

Note: Remote commands can only be executed as a Send request. The local file name is used to store the output of the remote command if the remote system is Windows or UNIX. The z/OS system does not send back output.

Examples of Executing a Remote Command on a Remote System

```
FTMSCMD /SEND [parameters] /COMMAND /RemoteCommand:dir local_file_name
```

```
FTMSCMD /SEND [parameters] /COMMAND /CALLJCL="TESTJCL ABC123″
```

The first example illustrates an execution of the command “dir” on a remote machine and the output is stored on the local machine in the “*local_file_name*” file. In the second example, “TESTJCL ABC123” is sent to a remote z/OS machine for execution. With remote command execution to a z/OS machine, no output is returned; therefore, a local file is unnecessary.

3.6 Command Line Parameters

This section describes each parameter that TIBCO MFT Platform Server for Windows uses. Some variables are specified as part of the parameters on the program call.

When entering parameters on the command line before the parameter name, you must type a forward slash (/). For example, `/DATATYPE=E`.

3.6.1 Required Parameters

To send or receive a file, you must specify the following parameters on the command line:

- The transfer direction
- The transfer type: action that must be performed at the destination (written to a file, sent to printer, or executed as a job).
- The file name

The following tables include brief descriptions of the different functions that you can specify on the command line.

Send	Indicates that the file will be sent from the local to the remote system.
Recv	Indicates that the file will be received from the remote system.
Submit	Submits a transfer to the platform server in conjunction with the FS:ServerName parameter. Specify the transfer parameters as you normally do on the command line. If the /submit parameter is selected and a server name is not specified (`/fs:ServerName`), an error is returned. **Note**: When using the /submit parameter, the file transfer is processed through the platform server (GUI Interface) and not from the Command Line Interface. When a transfer is submitted, the ftmscmd command ends when the transfer is submitted, not after the transfer is completed. The ftmscmd return code defines whether the transfer is submitted successfully, not whether the transfer is completed successfully. If you require the return code to show the file transfer status, then do not define the Submit parameter.

File	Store the contents of the file transfer in a file. This is the default setting. Local and remote file names are required.
Print	Send the file being transferred directly to the print queue or spool on the remote side. Local file and destination printer names are required.
Job	Send a local file to the remote system, where the partner executes it as a batch job. Local file name is required.
Command	Executes a command on a remote system. Output is stored in a local file specified by the user. **Note**: When the remote system is z/OS, the output is not returned.

3.6.2 Optional Parameters

You can define the following parameters either directly on the command line or in the environment variables.

Accelerate

Default	N
Allowable Values	Y, N
Minimum	N/A
Maximum	N/A
Alternate Specification	ACC

Setting this parameter to Y forces a transfer to be conducted through a Windows TIBCO Accelerator server using the TIBCO Accelerator technology with which you can improve data transfer speeds over IP networks with high latency.

ACCCompression

Default	N
Allowable Values	Y, BEST, DEFAULT, FAST, NO
Minimum	N/A
Maximum	N/A
Alternate Specification	ACCC, ACCCompress

When conducting file transfers through an Accelerator, you can configure the Accelerator server to compress the data being transferred. TIBCO Accelerator uses a proprietary compression compatible with ZLIB. By setting the compression to DEFAULT, your file receives the greatest compression and might take slightly longer to transfer than using FAST which results in your file being less compressed but sent faster.

Note: Do not specify Accelerator compression if you specify the /Compression parameter.

ACCEncryption

Default	N
Allowable Values	Y, N
Minimum	N/A
Maximum	N/A
Alternate Specification	ACCE, ACCEncrypt

When conducting file transfers through an Accelerator, you can tell the Accelerator server to encrypt the data with a 256-bit Blowfish encryption key by setting this parameter to Yes.

Note: Do not use Accelerator encryption if you specify the /EncryptionType parameter.

ACCHost

Default	None
Allowable Values	Host name or address
Minimum	N/A
Maximum	N/A
Alternate Specification	ACCH

This is the IP or host name of the Windows TIBCO Accelerator server. By defining a host on the command line or in a transfer template, you override the ACCHost value configured in the

RocketStreamConfig file. If the value is not defined and Accelerate is set to Yes, the value configured for ACCHost in the RocketStreamConfig file is used.

ACCMaxSpeed

Default	1000000
Allowable Values	256 - 1000000
Minimum	N/A
Maximum	N/A
Alternate Specification	ACCMAX

When conducting file transfers through an Accelerator, you can set the Max Speed in kilobytes per second to be used by the Accelerator server when you set this parameter in your command line or transfer template. If this parameter is not specified, no throttling of the transfer is performed.

ACCPort

Default	None
Allowable Values	Port number
Minimum	N/A
Maximum	N/A
Alternate Specification	N/A

This is the port number the Windows TIBCO Accelerator server is listening on for transfers using the TIBCO Accelerator technology. By defining a port number on the command line or in a transfer template, you override the ACCPort value configured in the RocketStreamConfig.txt file. The default value is 9099. If the value is not defined and Accelerate is set to Yes, the value configured for ACCPort in the RocketStreamConfig file is used.

ACCProtocol

Default	PDP
Allowable Values	TCP, UDP, PDP
Minimum	N/A
Maximum	N/A
Alternate Specification	ACCP

When conducting file transfers through an Accelerator, you can tell the Accelerator server to use its own enhanced version of User Datagram Protocol (UDP), TIBCO Accelerator's parallel implementation of TCP, called Parallel Delivery Protocol (PDP), or straight TCP.

AllocationPrimary

Default	N/A
Allowable Values	Numeric value
Minimum	N/A
Maximum	N/A
Alternate Specification	AP
Environment Variable	ALLOCATION_PRIMARY

This parameter indicates the primary allocation quantity in tracks or cylinders as indicated in the allocation type field.

AllocationSecondary

Default	N/A
Allowable Values	Numeric values
Minimum	N/A
Maximum	N/A
Alternate Specification	AS
Environment Variable	ALLOCATION_SECONDARY

This parameter indicates the secondary allocation quantity in tracks or cylinders as indicated in the allocation type field.

AllocationType= { TRACKS | CYLINDERS}

Default	TRACKS
Allowable Values	TRACKS, CYLINDERS
Minimum	N/A
Maximum	N/A
Alternate Specification	AT
Environment Variable	ALLOCATION_TYPE

This parameter instructs z/OS for creating new files. This parameter is ignored when sent to a platform other than z/OS. The valid values are as follows:

- T Tracks If data set size is expressed in tracks.
- C Cylinders If data set size is expressed in cylinders.
- M Megabytes If data set size is expressed in megabytes.
- K Kilobytes If data set size is expressed in kilobytes.

BlockSize

Default	N/A
Allowable Values	Numeric values
Minimum	N/A
Maximum	N/A
Alternate Specification	BS
Environment Variable	BLOCK_SIZE

This parameter specifies the size of the block. For FB, the block size must be a multiple of the record length, and for VB, the record length can be any size up to the block size minus four. The maximum number is 32760.

ClassOfService

Default	None
Allowable Values	Levels of buffer sizes specified in the cfcos.cfg configuration file.
Minimum	N/A
Maximum	N/A
Alternate Specification	COS
Environment Variable	None

This parameter defines the Class of Service used for this transfer. The Class of Service defines the sockets buffer sizes used for the transfer. Transfer performance can be improved by specifying larger Send and Receive buffers in the Class of Service table.

Note: The Initiator and Responder class of service are configured separately.

CheckPointRestart= { YES | NO | check_point_interval }

Default	YES (the default value is 5 minutes)
Allowable Values	YES, NO, check_point_interval
Minimum	1 minute
Maximum	90 minutes
Alternate Specification	CP
Environment Variable	CHECK_POINT_RESTART

Note: This parameter requires you to submit your transfer. (See Submit.) Otherwise, this parameter is ignored.

When enabled, packets of data can be sent periodically with the file transfer. These packets of data inform the receiver of the current point within the file. The receiver commits the latest data received to the file system and records the sender's checkpoint and its own checkpoint in the persistent queue. In the event of a failure, the initiator and the responder negotiate the saved checkpoint information and restart from the last known good checkpoint. Checkpoint is specified in units of time.

- YES Turn on checkpoint restart using the default interval of 5 minutes.
- NO Turn off checkpoint restart.
- nn Turn on checkpoint restart using the interval of nn minutes.

See the following sample command using CheckPointRestart:

```
FTMSCMD /S /F /SUBMIT /fs:LOCALHOST /LI:userid /LW:psw
/CheckPointRestart:1 /Ti:30 /TryCount:5 /DS:HOSTNAME /DT=BINARY  /RL=1
/RI=USERID /RW=pswd "F:\JOHN\QA\ONEX1.BIN"  "JTPLM.QAL.BATCHB.ONEX1"
```

COMMAND

Default	N/A
Allowable Values	Command to be executed
Minimum	N/A
Maximum	N/A
Alternate Specification	RC, RemoteCommand, E, EXEC, RE, REXXEXEC, SJ, SUBJCL, CJ, CALLJCL, CPG, CALLPGM

This parameter is used with the Remote Command Transfer feature.

Note: The alternate specifications for this parameter depend on the remote system on which the command is executed. The commands and platforms are as follows:

Alternate Specification	**Platform**
RC	Windows or UNIX
RemoteCommand	Windows or UNIX
E	z/OS
EXEC	z/OS
RE	z/OS
REXXEXEC	z/OS
SJ	z/OS
SUBJCL	z/OS
CJ	z/OS
CALLJCL	z/OS

CPG	z/OS
CALLPGM	z/OS
CALLPROG	z/OS

Compression= { YES | RLE | LZ | Z1 | Z2 | Z3 | Z4 | Z5 | Z6 | Z7 | Z8 | Z9 | NO }

Default	NO
Allowable Values	YES, RLE, LZ, Z1 – Z9, NO
Minimum	N/A
Maximum	N/A
Alternate Specification	CM
Environment Variable	COMPRESSION

This parameter compresses data on the sender side of the transfer and decompresses the data on the receiver side of the transfer. The default value is NO. If YES is specified, RLE is used.

LZ provides better compression ratios and compresses a wider variety of data types than RLE, but uses substantially more CPU. Choose LZ if you want better compression ratios and can spare CPU cycles.

RLE is more data-dependent than LZ. That is, the compression ratio might vary widely based upon the type of data being compressed. Choose RLE if your network bandwidth is not a critical bottleneck for your network and you must save CPU cycles.

Z1 through Z9 refer to levels of ZLIB compression. As the ZLIB level (1-9) increases, compression is better, but CPU usage rises drastically. ZLIB2 typically offers the best compromise between compression and speed.

CrLf = { YES | NO }

Default	NO
Allowable Values	YES, NO
Minimum	N/A
Maximum	N/A
Alternate Specification	CR
Environment Variable	CR_LF

CR_LF indicates that Carriage Return/Line Feed translation is performed during the transfer.

DataType= { B | E }

Default	E
Allowable Values	B, E
Minimum	N/A
Maximum	N/A
Alternate Specification	DT
Environment Variable	DATA_ TYPE

Data type specifies in what format the data is stored on the remote system. Binary indicates that no conversion can be done.
When communicating with EBCDIC systems such as zOS or IBMi, use /DT:E to convert data form ASCII to EBCDIC and vice versa.

DataClass

Default	Not Applicable
Allowable Values	1 - 8 character value

Minimum	1 character
Maximum	8 characters
Alternate Specification	DC
Environment Variable	DATA_CLASS

This represents the z/OS Data Class as defined to the Data Facility/System Managed Storage. In addition, you can use it to indirectly select file attributes such as Record Format and Logical Record Length. This value can contain 1 to 8 characters, including numeric, alphabetic, or national characters ($, #, or @). The first character must be alphabetic or national.

Destination

Default	N/A
Allowable Values	Host name or IP address
Minimum	N/A
Maximum	N/A
Alternate Specification	DS, Host, HO
Environment Variable	DESTINATION

This is the address of the remote system. This can be an IP name, IP address, or host name.

EncryptionType= {DES | 3DES | BF | BFL | RJ | NO}

Default	NO
Allowable Values	DES, 3DES, BF, BFL, RJ, NO
Minimum	N/A
Maximum	N/A
Alternate Specification	EN
Environment Variable	None

This parameter determines the level of encryption that is used by default in your system. The valid values are as follows:

- 128 No No default encryption.
- 64 DES DES encryption is used.
- 32 3DES Triple DES encryption is used.
- 16 BLOWFISH Blowfish encryption is used.
- 8 BLOWFISHLONG Blowfish Long encryption is used.
- 4 AES or Rijndael AES encryption is used.

Note: You can only select one type of encryption per transfer.

ExpirationDate

Default	None
Allowable Values	MM/DD/YYYY, HH:MM:SS, SUN, MON, TUES, WED, THURS, FRI, SAT
Minimum	N/A
Maximum	N/A
Alternate Specification	ED
Environment Variable	EXPIRATION_DATE

This parameter specifies the exact date and time when a transfer is no longer attempted. However, if this transfer is scheduled, that takes precedence over expiration. If expiration and retention are used, whichever value occurs first takes precedence.

FileAvailability= { I | D }

Default	I
Allowable Values	I, D
Minimum	N/A
Maximum	N/A
Alternate Specification	FA
Environment Variable	FILE_AVAIL

- I Immediate — This indicates that the file can be transferred immediately. This is the default setting.
- D Deferred — This specifies that the remote file availability might be deferred if the remote system uses this option. In the responder function, the platform server treats Deferred as tape and Immediate as disk.

FileTransferServer

Default	None
Allowable Values	1 - 31 character value
Minimum	1 character
Maximum	31 characters
Alternate Specification	FS
Environment Variable	File_Transfer_Server

This parameter is used in conjunction with the Submit parameter to submit a transfer to another platform server. MFT Platform Server uses the server name to obtain an RPC Binding Handle to MFT Platform Server that processes the file transfer and then submits the transfer to the server's queue.

When the server name specified in this parameter is invalid or no active MFT Platform Server is running on the machine, an error is returned.

If a platform server is selected and Submit is not specified, MFT Platform Server accepts the request for transfer; however, it only performs a two stage client to host transfer.

You can select a server name that is located in a different domain than the domain from where the file transfer is initiated. This is accomplished by specifying the domain and server names in the file transfer server parameter as follows:

```
FTMSCMD /send /file /FS:DOMAIN /SERVER
```

/?

This parameter is used to display the help message.

LocalCTFile

Default	N/A
Allowable Values	1 - 16 characters
Minimum	1 character
Maximum	16 characters
Alternate Specification	LCT
Environment Variable	LOCAL_CTFILE

This parameter is used to convert data between ASCII and EBCDIC. This parameter contains the name of the file, which is used to translate on the local side. You do not have to set this parameter if you are communicating from PC to PC.

Note: If you specify LCT, then the RCT (RemoteCTFile) must be set to NULL.

LocalDomain

Default	N/A
Allowable Values	1 - 15 characters
Minimum	1 character
Maximum	15 characters
Alternate Specification	LD
Environment Variable	LOCAL_DOMAIN

This parameter provides information about the user who initiated the transfer.

LocalPassword

Default	X:
Allowable Values	1 - 15 characters
Minimum	1 character
Maximum	15 characters
Alternate Specification	LW
Environment Variable	LOCAL_PASSWORD

This is the local logon password. It can be up to 20 characters and is case sensitive.

LocalUserId

Default	None
Allowable Values	20 characters
Minimum	1 character
Maximum	20 characters
Alternate Specification	LI
Environment Variable	LOCAL_USER_ID

This parameter defines the user who submits the transfer and is not case sensitive. It is used only when a transfer is submitted to the MFT Platform Server service.

LIST

Default	None
Allowable Values	32 characters
Minimum	1 character
Maximum	32 characters
Alternate Specification	None
Environment Variable	None

This parameter assigns the distribution list to use for the transfer request.

MaintainBDW

Default	N
Allowable Values	Y, N
Minimum	N/A
Maximum	N/A

Alternate Specification	MBDW
Environment Variable	None

This parameter defines whether to maintain the Block Descriptor Word (BDW) when sending or receiving variable block binary files to or from a z/OS system. If the data being sent or received is not in the proper BDW format, the transfer will fail.

MaintainRDW

Default	N
Allowable Values	Y, N
Minimum	N/A
Maximum	N/A
Alternate Specification	MRDW
Environment Variable	None

This parameter defines whetehr to maintain the Record Descriptor Word (RDW) when sending or receiving variable block binary files to or from a z/OS system. If the data being sent or received is not in the proper RDW format, the transfer will fail.

MgmtClass

Default	None
Allowable Values	1 - 8 character value
Minimum	1 character
Maximum	8 characters
Alternate Specification	MC
Environment Variable	MGMT_CLASS

This represents the z/OS Management Class as defined to the Data Facility /System Managed Storage. This value can contain 1 to 8 characters, including numeric, alphabetic, or national characters ($, #, or @). The first character must be an alphabetic or national character.

NodeName

Default	None
Allowable Values	1 - 256 character
Minimum	1 character
Maximum	256 characters
Alternate Specification	NODE
Environment Variable	NODENAME

This is the name of the remote node used in the transfer. The node name cannot contain any spaces.

NoLogo

Do not display logo or version information at the start of the command prompt.

NotifyLocalUser

Default	None
Allowable Values	1 - 80 character name or email address
Minimum	1 character name
Maximum	80 character name or email address
Alternate Specification	NLU
Environment Variable	NOTIFY_LOCAL_USER

This is the name of the local user to notify when this file transfer is completed, either successfully or unsuccessfully. The value can contain 1 to 80 characters. For this name, it is good practice that you use either your own user ID or one of your Operations Support team members'. When NotifyLocaluserType is set to MAIL, then this parameter defines the email address where the notification is sent.

NotifyLocalUserType= {MAIL} | : {SUCCESS | FAILURE}

Default	None
Allowable Values	MAIL
Minimum	N/A
Maximum	N/A
Alternate Specification	NLT
Environment Variable	NOTIFY_LOCAL_USER_TYPE

This is the type of user ID to notify after a file transfer terminates. This is used in conjunction with the NOTIFY_LOCAL_USER parameter.

The allowable values are as follows:

- MAIL — Give email notification for both successful and failed transfers.
- MAIL:SUCCESS — Give email notification for only successful transfers.
- MAIL:FAILURE — Give email notification for only failed transfers.

NotifyRemoteUser

Default	None
Allowable Values	1 - 20 character name
Minimum	1 character name
Maximum	20 character name
Alternate Specification	NRU
Environment Variable	None

This is the name of the remote user to notify when this file transfer is completed, either successfully or unsuccessfully. The value can contain 1 to 20 characters. For this name, it is good practice to use either your own user ID or one of your Operations Support team members’.

NotifyRemoteUserType= { NONE | M } : { S | F }

Default	None
Allowable Values	NONE, M
Minimum	N/A
Maximum	N/A
Alternate Specification	NRT
Environment Variable	None

This is the type of user ID to notify after a file transfer terminates. This is used in conjunction with the **NotifyRemoteUser** parameter. If MAIL is selected, you can decide whether you want notification for successful or failed transfers only.

The valid values are as follows:

- NONE — No notification.

- M MAIL Give email notification for both successful and failed transfers.
- M:S MAIL:SUCCESS Give email notification for only successful transfers.
- M:F MAIL:FAILURE Give email notification for only failed transfers.

PermittedActions= {S | H | A | R | C | Z | E | T}

Default	None
Allowable Values	S, H, A, R, C, Z, E, T
Minimum	N/A
Maximum	N/A
Alternate Specification	PA
Environment Variable	None

- S System Indicates that the file is a system file and can be viewed only by the operating system and not the user.
- H Hidden Indicates that the file cannot be seen by the user.
- A Archive Select archive if you want to mark a file that has changed since it was last backed up.
- R Read Only Indicates that the file being accessed can only be viewed by the user. No changes can be made to the file.
- C NTFS Compressed Indicates that the system compresses a file that is sent to an NTFS drive.
- Z Control Z Indicates that when enabled, the system appends a CR/LF (0x0d, 0x0a) to the end of the file, followed by the DOS End of File character: Control Z (0x1a). If a trailing Control Z or CR/LF is already present, the system does not add them again. This feature is only available when Carriage Return/Line Feed processing is enabled.
- E Control Z added to EOF Indicates that when enabled, the system appends a Control Z (0x1a) to the end of the file.
- T CR/LF added to EOF Indicates that when enabled, the system appends a CR/LF (0x0d, 0x0a) to the end of the file.

Port

Default	46464
Allowable Values	1025 - 65535
Minimum	1025
Maximum	65535
Alternate Specification	PT
Environment Variable	PORT

This parameter names the port number for a TCP/IP transfer. The default IP port number for MFT Platform Server is 46464, but you can change it to any number from 1025 to 65535. For an SSL transfer, the SECURE parameter must also be used.

PostAction={S|F,L|R,COMMAND|CALLJCL|CALLPGM|SUBMIT,data to be processed}

Default	None
Allowable Values Parameter 1	S=Success F=Failure
Allowable Values	L=Local R=Remote

Parameter 2	
Allowable Values Parameter 3	CALLJCL: CALL with JCL Linkage CALLPGM: CALL with Program linkage SUBMIT: Submit JCL to internal reader COMMAND: Execute command
Allowable Values Parameter 4	Data to be processed.
Minimum	N/A
Maximum	N/A
Alternate Specification	PPA1 | 2 | 3 | 4
Environment Variable	POST_ACTION1 | 2 | 3 | 4

This parameter defines the post processing actions that can be performed for a file transfer. Up to 4 post processing actions can be defined for an individual file transfer. Actions can be done based on success or failure of a file transfer. Actions can be done locally or remotely as well. Any combination of up to four S|F, L|R and command types are allowed. Parameters can be delimited by commas or colons. If any spaces are imbedded in the data, then the entire parameter must be enclosed in double quotation marks.

The table below indicates the type of actions that can be performed, and the data necessary for each.

Action	**Support**	**Data necessary**
CALLJCL	z/OS	Program name (1 to 8 bytes) followed by data
CALLPGM	z/OS	Program name (1 to 8 bytes) followed by data
SUBMIT	z/OS	DSN(MEMBER) or MEMBER name
COMMAND	Windows, UNIX	Command name followed by data

Note that the difference between CALLJCL and CALLPGM is the way that the parameters are passed to the called program. CALLJCL uses JCL linkage in which the parameter length is the first two bytes of the data passed. CALLPGM used standard IBM linkage conventions for calling programs.

You can run CALLJCL, CALLPGM and SUBMIT if the remote system is mainframe, otherwise you can only run COMMAND. The data to be processed section is limited to 256 bytes. To alleviate that restriction and to make it easier to define the command data, you can use PPA tokens in the data to be processed section. For more information, see C.4 PPA Tokens.

Priority= { 3 | n }

Default	3
Allowable Values	1 - 6
Minimum	1
Maximum	6
Alternate Specification	PR
Environment Variable	PRIORITY

This is the priority that is applied when the thread for the file transfer is created. This does not indicate the priority that the job has in the MFT Platform Server work queue. n is a decimal number from 1 to 6 (highest priority to lowest priority), which indicates the priority of the file transfer. It is good practice to use the default value 3.

ProcessName

Default	CyberFus
Allowable Values	1 - 8 characters
Minimum	1 character
Maximum	8 characters
Alternate Specification	PN
Environment Variable	PROCESS_NAME

This is an eight-character field that describes the transaction being processed.

RemoteCTFile

Default	N/A
Allowable Values	1 - 16 characters
Minimum	1 character
Maximum	16 characters
Alternate Specification	RCT
Environment Variable	REMOTE_CTFILE

This is used to convert data between ASCII and EBCDIC. This parameter contains the name of the file, which is used to translate on the remote side. You do not have to set this parameter if you are communicating from PC to PC.

Note: If you specify RCT, then LCT (LocalCTFile) must be set to NULL.

RecordFormat = { F | FB | V | VB | U | FA | VA | FM | VM | FBA | VBA | FBM | VBM }

Default	FB
Allowable Values	F, FB, V, VB, U, FA, VA, FM, VM, FBA, VBA, FBM, VBM
Minimum	N/A
Maximum	N/A
Alternate Specification	RF
Environment Variable	RECORD_FORMAT

This parameter determines the significance of the character logical record length. You can specify the parameter as fixed, variable, or undefined. This is a z/OS specific parameter. Choose one of the following formats:

- F Fixed — Each record contains exactly the number of characters defined by the record length parameter.
- FB Fixed Block — All blocks and all logical records are fixed in size. One or more logical records are included in each block.
- V Variable — The length of each record is less than or equal to the record length parameter.
- VB Variable Block — Blocks as well as logical record length can be of any size. One or more logical records are included in each block.
- U Undefined — Blocks are of an undefined size. No logical record is included in the blocks. The logical record length is displayed as zero. This record format is usually only used in load libraries. Block size must be used if you are specifying Undefined.
- FA Fixed ASA — Each record contains exactly the number of characters and uses ASA characters on z/OS.

- VA Variable ASA The length of each record is less than or equal to this number and uses ASA characters on z/OS.
- FM Fixed Machine Each record contains exactly the number of characters defined by the record length parameter and uses MACHINE characters on z/OS.
- VM Variable Machine The length of each record is less than or equal to the record length parameter and uses MACHINE characters on z/OS.
- FBA Fixed Blocked ASA All blocks and all logical records are fixed in size. One or more logical records are included in each block and use ASA characters on z/OS.
- VBA Variable Blocked ASA Blocks, as well as logical record length, can be any size. One or more logical records are included in each block and use ASA characters on z/OS.
- FBM Fixed Blocked Machine All blocks and all logical records are fixed in size. One or more logical records are included in each block and use MACHINE characters on z/OS.
- VBM Variable Blocked Machine Blocks, as well as logical record length, can be any size. One or more logical records are included in each block and use MACHINE characters on z/OS.

RecordLength= { nnnnn | 0 }

Default	1 (F or FB) 5 (V or VB)
Allowable Values	1 - 32760
Minimum	1 (F or FB) 5 (V or VB)
Maximum	32760 (F or FB) 32756 (V or VB)
Alternate Specification	RL
Environment Variable	RECORD_LENGTH

This is the maximum logical record length, which is occasionally called the string length used to encode the data records of the file. The maximum logical record length in z/OS is 32,760. Omit this parameter if you are sending or receiving a file into a file that already exists because the platform server will determine the appropriate length. This parameter is ignored when sent to TIBCO MFT Platform Server for Windows because it is a z/OS-specific parameter.

Note: If RecordFormat=F or FB, the allowable values are 1 - 32760. If RecordFormat=V or VB, the allowable values are 5 - 32756.

RemoteDomain

Default	The domain of the remote system where MFT Platform Server is running.
Allowable Values	Domain name up to 15 characters
Minimum	1 character
Maximum	15 characters
Alternate Specification	RD
Environment Variable	REMOTE_DOMAIN

By specifying the domain name as part of the transfer, you can specify the network user under whose authority the transfer is executed.

RemotePassword

Default	None
Allowable Values	Password up to 20 characters
Minimum	1 character
Maximum	20 characters
Alternate Specification	RW
Environment Variable	REMOTE_PASSWORD

This is the remote logon password. It can be up to 20 characters and can be case sensitive. Specify this only if required by the remote computer.

RemotePrinterName

Default	N/A
Allowable Values	Any combination of up to 255 characters
Minimum	1 character
Maximum	255 characters
Alternate Specification	RP
Environment Variable	PRINTER_NAME

This is the name of the remote printer to which the job is sent when performing File to Print transfer.

RemoteUserId

Default	None
Allowable Values	1 - 20 characters
Minimum	1 character
Maximum	20 characters
Alternate Specification	RI
Environment Variable	REMOTE_USER_ID

The remote user ID specifies the ID to use when remote system security is checked. The remote user ID is generally not case sensitive, unless going to a UNIX system.

RemoveTrailingSpaces

Default	N
Allowable Values	Y, N
Minimum	N/A
Maximum	N/A
Alternate Specification	RTS
Environment Variable	REMOVE_TRAILING_SPACES

This option removes all spaces or binary zeros at the end of a record when receiving text files from the z/OS platform.

RetentionPeriod

Default	0
Allowable Values	0 - 32,767
Minimum	0
Maximum	32,767
Alternate Specification	RT

Environment Variable	RETENTION_PERIOD

This parameter specifies the number of days that pass from the start of the transfer to the point it is no longer attempted. If expiration and retention are used, whichever value occurs first takes precedence.

RetenPeriodExpDate

Default	N
Allowable Values	N, number of days, yyyy/ddd
Minimum	N/A
Maximum	N/A
Alternate Specification	RPED
Environment Variable	None

This parameter specifies the retention period or expiration date of the file in the remote system. The format of the entered value determines whether the parameter is used as a retention period or as an expiration date. The retention period is the number of days, after which the file expires. The expiration date is the date, in Julian format, when the file expires. This parameter is typically used on z/OS platforms for tape processing to prevent a tape from being overwritten. This parameter must be carefully defined with a disk file. The default is no expiration date on the file. Value 99/000 or 1999/000 means that the file will not expire. This parameter is only supported for send transfers to a z/OS system.

ScanSubDir

Default	NO
Allowable Values	Y, N
Minimum	N/A
Maximum	N/A
Alternate Specification	SSD, Scan_Sub_Dir
Environment Variable	None

This parameter defines whether to scan all the subdirectories from the file path. This parameter only applies to directory transfers.

ScheduleAt

Default	None
Allowable Values	MM/DD/YYYY, HH:MM:SS
Minimum	N/A
Maximum	N/A
Alternate Specification	SAT
Environment Variable	SCHEDULE_AT

Note: This parameter requires you to submit your transfer.

This parameter specifies the date and time when a transfer is executed. The valid ranges for HH:MM:SS are as follows:

HH: 0 - 23
MM: 0 - 59
SS: 0 - 59

ScheduleHoldErrors

Default	N
Allowable Values	Y, N
Minimum	N/A
Maximum	N/A
Alternate Specification	SHE
Environment Variable	SCHEDULE_HOLD_ERRORS

This parameter defines whether to put a scheduled transfer on hold if a permanent error occurs. If N is specified, the system continues to try the transfer even after a permanent error occurred. Examples of permanent errors include the remote file not existing, bad user ID or password.

ScheduleInterval = {D7|WK|2WK|MON|2MON|QTR|2QTR|YR|2YR| E:n:u }

Default	None
Allowable Values	D7, WK, 2WK, MON, 2MON, QTR, 2QTR, YR, 2YR, E:n:u
Minimum	N/A
Maximum	N/A
Alternate Specification	SRI
Environment Variable	SCHEDULE_INTERVAL

This parameter specifies the interval at which the transfer is repeated. This parameter is used only if you are scheduling the transfer.

The allowable values are as follows:

- D7 Daily 7 Sunday through Saturday.
- WK Weekly Every week.
- 2WK Bi-Weekly Every other week.
- MON Monthly Once a month.
- 2MON Bi-Monthly Every other month.
- QTR Quarterly Every 3 months.
- 2QTR Semi-Annually Every 6 months.
- YR Yearly Once a year.
- 2YR Bi-Yearly Once every other year.
- E:n:u Every Every n number of seconds, minutes, hour(s), day(s), week(s), month(s), or year(s).

ScheduleRepeat= { N | I | T:x | U }

Default	None
Allowable Values	N, I, T:x, U
Minimum	N/A
Maximum	N/A
Alternate Specification	SRE
Environment Variable	SCHEDULE_REPEAT

This parameter specifies the rate at which the schedule is repeated.

The allowable values are as follows:

- N NO Do not repeat the transfer.

- I INFINITE Repeat the transfer infinitely.
- T:x TIMES Repeat the transfer x amount of times.
- U UNTIL Repeat the transfer until the specified date and time. Format – MM/DD/YYYY, HH:MM:SS

SecureCommunication

Default	0
Allowable Values	0, 1
Minimum	N/A
Maximum	N/A
Alternate Specification	SSL
Environment Variable	SECURE

Specify this parameter to use SSL communication; it must be positioned directly before the local and remote file names in the command. Be sure to define the SSL port being used in the PORT parameter when making secure transfers

SecurityAttribFileName

Default	None
Allowable Values	1 - 8 character value
Minimum	1 character
Maximum	8 characters
Alternate Specification	SA
Environment Variable	SECURITY_ATTRIB_FILE

This parameter specifies the name of the file that the receiving partner uses as a template for its Access Control List (ACL). The ACL of this file is copied to the ACL of the destination file. For this feature to function properly on Windows, the file specified must be readable by the partner which is receiving the File to File transfer and the file being created must be located on an NTFS drive.

Status

Displays the current transfer status. This will display the packet count and the byte count sent or received. When sending a file, this will display the percentage completed.

StopOnFailure

Default	N
Allowable Values	Y, N
Minimum	N/A
Maximum	N/A
Alternate Specification	SOF, SONF
Environment Variable	None

This parameter is used for directory transfers and transfers using a distribution list and indicates if any file transfer fails, the platform server will not try to transfer the rest of the files.

StoreClass

Default	None
Allowable Values	1 - 8 character value
Minimum	1 character
Maximum	8 characters

Alternate Specification	SC
Environment Variable	STORAGE_CLASS

This represents the z/OS Storage Class as defined to the Data Facility /System Managed Storage, which is used to indicate the media type of the host file and the backup, restore, and archive policies of the installation. This value can contain 1 to 8 characters, including numeric, alphabetic, or national characters ($, #, or @). The first character must be alphabetic or national.

SysoutClass

Default	None
Allowable Values	0 - 9, A - Z
Minimum	0 or A or a
Maximum	9 or Z or z
Alternate Specification	CL
Environment Variables	SYSOUT_CLASS

SysoutClass describes to which class the JES output is routed. On z/OS system, the printer queues are organized around a printer class, and not a specific printer. The class has a one-character value which is either alphabetic or numeric. Query the z/OS staff for the values to supply.

SysoutCopies

Default	None
Allowable Values	1 - 999
Minimum	1
Maximum	999
Alternate Specification	SP
Environment Variables	SYSOUT_COPIES

This is the number of copies to print of a particular report on the remote computer.

SysoutDestination

Default	None
Allowable Values	1 - 8 characters
Minimum	N/A
Maximum	N/A
Alternate Specification	SD
Environment Variables	SYSOUT_DESTINATION

This is the destination of the job submitted to the internal reader.

SysoutFcb

Default	None
Allowable Values	1 - 4 characters
Minimum	N/A
Maximum	N/A
Alternate Specification	SB
Environment Variables	SYSOUT_FCB

This field is applied when the remote computer is a z/OS system. This is the Form Control Buffer name as defined to JES.

SysoutForm

Default	None
Allowable Values	1 - 8 characters
Minimum	N/A
Maximum	N/A
Alternate Specification	SF
Environment Variables	SYSOUT_FORM

This is the form name upon which the report is printed on the remote computer.

SysoutUserId

Default	None
Allowable Values	1 - 8 characters
Minimum	N/A
Maximum	N/A
Alternate Specification	SI
Environment Variables	SYSOUT_USER_ID

This is the user name assigned to a job submitted to the internal reader.

SysoutWriter

Default	None
Allowable Values	1 - 8 characters
Minimum	N/A
Maximum	N/A
Alternate Specification	SW
Environment Variables	SYSOUT_WRITER

This indicates the external writer name that is used to process this printer file on the z/OS system.

TCP

If not defined, TCP is the default value.

Test

Default	N
Allowable Values	Y
Minimum	N/A
Maximum	N/A

With this parameter, you can display the local and remote file names to verify that the file names are correct rather than perform the actual transfers. This is used when running directory transfer requests and transfers using a distribution list.

Timeout

Default	120
Allowable Values	0 - 1440
Minimum	0
Maximum	1440
Alternate Specification	TO
Environment Variable	None

This parameter specifies the amount of time (in minutes) a connection stays open while waiting for a response from the remote side. When the time is reached, the connection is ended.

TraceLevel

Default	0
Allowable Values	0 - 9
Minimum	0
Maximum	9
Alternate Specification	TL
Environment Variable	TRACE_LEVEL

This parameter indicates the level of messages that is produced during the transfer. Higher values produce more output—although they slow system performance. These traces are created through ftmscmd – to receive the traces you must manually enter the parameter in the batch command.

TryCount= { nn | 1 }

Default	1
Allowable Values	0 - 9998
Minimum	1
Maximum	Unlimited (0)
Alternate Specification	TC
Environment Variable	TRY_COUNT

Where nn is a decimal number from 0 to 10 that indicates the maximum number of times that this file transfer can be attempted before it is removed from the MFT Platform Server work queue. TryCount=0 indicates no limit. It is good practice to use the default value.

TryInterval

Default	1
Allowable Values	1 - 9998
Minimum	1
Maximum	9998
Alternate Specification	TI
Environment Variable	None

This parameter defines the interval (in minutes) at which the transfer is retried.

Unit

Default	SYSDA
Allowable Values	1 - 8 character name
Minimum	1 character
Maximum	8 characters
Alternate Specification	UN
Environment Variable	UNIT

This is the 1 - 8 character name of the z/OS unit where the host data set is to be allocated.

UnixPermissions

Default	File permissions of the file being transferred
Allowable Values	Three digit number. Each digit must be between 0 and 7.
Minimum	N/A
Maximum	N/A
Alternate Specification	UPERM

Environment Variable	UNIX_PERMISSIONS

When a file is created on a UNIX system, you can set UnixPermissions on the file. UNIX permissions are defined by a three digit number such as 777 (the same as for chmod command). **Note:** Permissions are set under the file only if the file is created. In other words, this parameter only works when the file is being created.

UserData

Default	none
Allowable Values	Any string of up to 25 characters
Minimum	0 or none
Maximum	25 characters
Alternate Specification	UD
Environment Variable	USER_DATA

This parameter describes the transfer on the local and remote systems. The description is logged into the history files. If you must imbed spaces in this field, you can either specify this parameter in the Environment Variable (SET command) or enclose the value in double quotation marks ("x"). The description can contain any alphabetic, numeric, or national characters of up to 25 characters.

VolumeSerialNumber

Default	none
Allowable Values	1 - 6 character name
Minimum	none
Maximum	6 characters
Alternate Specification	VS
Environment Variable	VOLUME

This parameter indicates the z/OS default volume serial to use for new data sets created by MFT Platform Server Responder. If you leave this parameter blank, it uses the VOLSER that is specified in the GLOBAL parameters on the z/OS system. This parameter is ignored when sent to TIBCO MFT Platform Server for Windows.

WriteMode= { C | R | A | CR | CA | CN }

Default	CR
Allowable Values	C, R, A, CR, CA, CN
Minimum	N/A
Maximum	N/A
Alternate Specification	WM
Environment Variable	WRITE_MODE

This parameter indicates the effect on the remote file.

- C Create — Create the remote file. Abort the transfer if the file already exists.
- R Replace — Replace the remote file only. If the file does not yet exist, terminate the transfer.
- A Append — Append to the remote file.
- CR Create Replace — Create the remote file or replace it if it already exists.
- CA Create Append — Create the remote file or append it if it already exists.
- CN Create Replace New — Create the remote file or replace it with new attributes. When specified for transfers to Windows, CN indicates

that the system must create directory paths as needed.

3.7 Use of Errorlevel with FTMSCMD

FTMSCMD passes back return codes to assist the programmer when writing batch jobs.

The following example batch job executes a transfer. A message is displayed to the screen indicating the success or failure of the transfer.

```
@echo off
FTMSCMD /nologo /S /F /host:danl1i2 /ri:ftmsusr1 /rw:ftmspswd /rl:80 /rf:f
“c:\data\production information file1.dat” prftms.xabl.data.prodinf1
2>errorlog.txt

if errorlevel 1     goto ERROR
if errorlevel 0     goto SUCCESS

:ERROR
       echo transfer failed
       goto END

:SUCCESS
       echo transfer successful
       goto END

:END
       echo batch program complete
```

3.7.1 Overview of Sample Batch Program

The first line, `@echo off`, instructs the batch program not to write messages to the screen. The second and third lines indicate the file transfer.

Note: `/NOLOGO` instructs the FTMSCMD program not to display product information when performing the transfer. `2>errorlog.txt` writes any message that is issued during this batch job to errorlog.txt.

The next line directs the batch job to jump to the area labeled `:ERROR` and perform the tasks in that section if the error level passed back from the FTMSCMD program is 1.

The next line directs the batch job to jump to the area labeled `:SUCCESS` and perform the tasks in that section if the error level passed back from the FTMSCMD program is 0.

Note: The echo specified in each of the two sections instructs the batch program to write the trailing text to the screen, overriding the previous command to turn echo off.

For additional information on how to write batch programs using `errorlevel`, see Microsoft’s MS DOS documentation.

4

4. Extended Features

TIBCO MFT Platform Server for Windows provides the following features:

- Access Control
- CFAlias
- CFINQ
- Configured Post Processing
- Custom Code Page Conversion
- Directory Named Initiation (DNI) GUI and Command Line Interface
- Directory Transfer and Wildcard Support
- fusping Utility
- fusutil Utility
- Nodes, Profiles, and Distribution Lists
- TIBCO Accelerator
- SSL

4.1 Access Control

For MFT Platform Server, you can send a file to the Windows platform and the file automatically goes to a predefined directory based on user-defined criteria (USERID, NODE, and/or IPADDR).

To perform Access Control, the Access Control configuration file, called AccessControl.cfg by default, must be selected under the Responder tab under Server Properties. This feature is only used for TIBCO MFT Platform Server for Windows Responder.

4.1.1 Access Control Parameters

A sample of the Access Control file, AccessControl.cfg, is located in the MFT Platform Server directory.

The following table lists the definition of each parameter:

Parameter	Description
USERID	Defines the local user ID. Either this or **NODE/IPADDR** must be specified. Both **USERID** and **NODE/IPADDR** can be specified. A value of DEFAULT indicates that this is the default value for a system.
NODE	Defines the node definition. Either the **NODE/IPADDR** or **USERID** must be specified. Both **USERID** and **NODE/IPADDR** can be specified. A value of DEFAULT indicates that this is the default value for a system. This parameter is mutually exclusive with the **IPADDR** parameter. When defining nodes in this file, ensure that you use the proper case as these files are case sensitive.
IPADDR	Defines the IP address in dotted decimal notation. Either the **NODE/IPADDR** or **USERID** must be specified. Both **USERID** and **NODE/IPADDR** can be specified. This parameter is mutually exclusive with the **NODE** parameter.
DESCRIPTION	Defines a 32-byte description or comment.
SEND_DIR	Defines the default directory for files to be sent to another system. If this parameter is not defined, no default value is available for the files sent.
RECEIVE_DIR	Defines the default directory for files to be received from another system. If this parameter is not defined, no default value is available for the files received.
COMMAND_DIR	Defines the default directory for commands executed on this system. If this parameter is not defined, no default value is available for the commands executed.
SUBMIT_DIR	Defines the default directory for files to be submitted into the z/OS internal reader. For MFT Platform Server on z/OS, you can also set this parameter to SUBMIT_HLQ. This parameter is required if **SUBMIT_OPTION** is set to ROOT or FORCE. This parameter is valid only for MFT Platform Server on z/OS. .
SEND_OPTION	Defines the options for sending files. The valid values are as follows: **ROOT -** If a directory is specified, the directory is appended to the

	directory defined by the SEND_DIR parameter. **FORCE -** If a directory is specified, the directory is changed to the directory defined by the SEND_DIR parameter. The directory name defined in the request is ignored. The file name is appended directly to the SEND_DIR parameter. **ALLOW -** If a directory is specified, the directory is used. If a directory is not defined, it is changed to the directory defined by the SEND_DIR parameter. **REJECT -** If a directory is specified on a Send, the file transfer terminates with errors. Otherwise, data is processed from the SEND_DIR directory. **NEVER -** The **NODE**, **USERID**, or **IPADDR** cannot send a file. **USE -** The directory name specified in the file transfer request is used. If no directory name is defined in the file transfer request, the Windows default directory is used.
RECEIVE_OPTION	Defines the options for receiving files. The valid values are as follows: **ROOT -** If a directory is specified, the directory is appended to the directory defined by the RECEIVE_DIR parameter. **FORCE -** If a directory is specified, the directory is changed to the directory defined by the RECEIVE_DIR parameter. The directory name defined in the request is ignored. The file name is appended directly to the RECEIVE_DIR parameter. **ALLOW -** If a directory is specified, the directory is used. If a directory is not defined, it is changed to the directory defined by the RECEIVE_DIR parameter. **Note**: By setting ALLOW, files can be written to directories other than that is defined in the RECEIVE_DIR parameter. If a relative path (directory without a slash in the beginning. For example, tmpdir\filename.txt) is used for a remote file name in the transaction coming in, MFT Platform Server places files in the current directory where platform server is executing if the user has access rights. This is the MFT Platform Server System directory. **REJECT -** If a directory is specified on a Send, the file transfer terminates with errors. Otherwise, data is processed from the RECEIVE_DIR directory. **NEVER -** The **NODE** or **USERID** cannot receive a file. **USE -** The directory name specified in the file transfer request is used. If no directory name is defined in the file transfer request, the Windows default directory is used.
COMMAND_OPTION	Defines the options for executing commands. The valid values are as follows: **ROOT -** If a directory is specified, the directory is appended to the directory defined by the COMMAND_DIR parameter. **NEVER -** The **NODE**, **USERID**, or **IPADDR** cannot execute commands. **USE -** The directory name specified in the file transfer request is used.

	If no directory name is defined in the file transfer request, the Windows default directory is used.
SUBMIT_OPTION	Defines the options for submitting jobs. The valid values are as follows: **ALLOW –** The user can submit jobs. **NEVER -** The **NODE** or **USERID** cannot receive a file.

4.1.1.1 Directory Name Used in Request

If the directory name is defined in the RECEIVE_DIRECTORY parameter and the FORCE parameter is defined, the file name is extracted from the local file path in the request, and is appended to the directory defined by the RECEIVE directory.

Example:

RECEIVE_DIR=c:\sales\
RECEIVE_OPTION=FORCE
The local file in the request is: c:\2005\accounting\tax.xls
The actual file name is: c:\sales\tax.xls

If the directory name is defined in the RECEIVE_DIRECTORY parameter and the ROOT parameter is defined, the local file name (which can consist of a directory and file name) is appended to the directory defined by the RECEIVE directory.

Example:

RECEIVE_DIR=c:\sales\
RECEIVE_OPTION=ROOT
The local file in the request is: c:\2005\accounting\tax.xls
The actual file name is: c:\sales\2005\accounting\tax.xls

4.1.1.2 Continuation and Comments

Parameters can be entered on a single line or on multiple lines. Parameters are delimited by a comma. If a space follows the comma, the parameter is continued on the next line. If the parameter contains a special character, enclose the parameter in double quotation marks. A parameter that is not terminated by a comma signifies the end of the Access Control entry.

Example:

USERID=DEFAULT,
NODE=NODEA,
SEND_DIR=c:\temp\,
SEND_OPTION=ROOT,
RECEIVE_OPTION=NEVER

The following command is the same as those above:
USERID=DEFAULT,NODE=NODEA,SEND_DIR="c:\temp\",SEND_OPTION=ROOT,RECEIVE_OPTION=NEVER

Comments are defined by placing an asterisk (*) in column 1. UNIX comments such as // and /* */ can be implemented as well.

4.1.1.3 Default Entries

You can specify default entries for the USERID and NODE parameters by using the value DEFAULT. This provides a default entry in case no matches are made.

Example:

```
USERID=DEFAULT,
NODE=NODEA,
SEND_DIR=c:\temp\,
SEND_OPTION=ROOT,
RECEIVE_OPTION=NEVER
*
USERID=DEFAULT
NODE=DEFAULT
SEND_OPTION=NEVER
RECEIVE_OPTION=NEVER
```

4.1.1.4 Parameter Validation

On Windows and UNIX platforms, the Access Control file is read each time a transfer is received. Parameter validation is only performed when there is a match for the NODE/USER and transfer type (Send, Receive, Command, File…).

On z/OS, the platform server validates all CFACCESS parameters at startup and whenever the CFACCESSREFRESH command is executed. Only valid entries are saved into memory. When file transfer requests are received, the entries in memory are checked.

4.1.2 Sample of AccessControl.cfg File

The following example is a sample Access Control configuration file, called AccessControl.cfg by default.

The platform server does not look for the best match; it looks for the first match. Therefore, it is good practice to list the most specific information first in the AccessControl.cfg file and the more generic information last.

```
USERID=JohnDoe,
NODE=Billing,
DESCRIPTION=restrict billing dept from sending files,
SEND_DIR=c:\jdoe\sendfiles,
RECEIVE_DIR=c:\jdoe\recvfiles,
COMMAND_DIR=c:\jdoe\cmdfiles,
SEND_OPTION=ROOT,
RECEIVE_OPTION=FORCE,
COMMAND_OPTION=NEVER,
SUBMIT_OPTION=NEVER
```

SEND_OPTION, RECEIVE_OPTION, and COMMAND_OPTION all have ROOT as the default value. SUBMIT_OPTION has NEVER as the default value. The rest of the parameters do not have default values.

4.2 CFAlias

Some architectures do not want users to know the file names or locations of the files they send to the server, or perhaps the administrator wants to handle file naming and locations automatically for the user based on the **USERID**, **NODE**, and/or **IPADDR**. CFAlias allows the administrator to associate an alias with an actual fully qualified file name, where the end user has no idea of the actual file name used by the system. MFT Platform Server also supports substitutable parameters that can be used to assign values to the Responder's file names. To use this feature, the CFAlias configuration file, called CfAlias.cfg by default, must be selected under the Responder tab under Server Properties. This feature is only used for the TIBCO MFT Platform Server for Windows Responder.

4.2.1 CFAlias Parameters

The following table lists the parameter values supported. The syntax is similar to the AccessControl syntax. Parameters must be entered one per line and continuations are defined by a comma followed by a space.

Parameter	Description
USERID	Defines the name of the user that initiated the transfer. The special value DEFAULT indicates a match with any user.
NODE	Defines the name of the node that initiated the transfer. The special value DEFAULT indicates a match with any node. When defining nodes in this file, ensure that you use the proper case because these files are case sensitive.
IPADDR	Defines the name of the IP address that initiated the transfer.
TYPE	Defines the type of the request. The valid values are SEND, RECEIVE, or BOTH. This parameter is relative to the Responder. If the Initiator issues a SEND request, the CFAlias feature considers this a RECEIVE request because it is operating as the responder.
FILE	Defines the actual fully qualified file name to be used.
ALIAS	Defines the name of the file that is sent by the initiator.
ALLOW	Defines whether you can define the actual file name when no match is made with an alias grouping .The valid values are YES or NO. When specified as YES, a match indicates that you can define the actual file name if no match is made on an alias grouping. When defined as NO, the request fails if no match is made with an alias grouping. NODE/IPADDR and/or USERID must be defined. When ALLOW is not defined, FILE and ALIAS must be defined. When ALLOW is defined, the FILE and ALIAS parameters are not supported. If a sender's parameters do not match any entry in the alias config file, the transfer is rejected.

4.2.2 Substitutable Parameters

The MFT Platform Server administrator can define substitutable parameters in the FILE parameter of the CFAlias file. Substitutable parameters are defined by a percent sign (%) followed by the parameter name.

The following substitutable parameters are supported:

%JDATE	Julian Date (YYDDD)
%JDATEC	Julian Date (CCYYDDD)
%GDATE	Gregorian Date (YYMMDD)
%GDATEC	Gregorian Date (CCYYMMDD)
%TIMET	Time (HHMMSST)
%TIME	Time (HHMMSS)
%NODE	Node Name (if no node is defined, use the value NODE)
%USER	User Name
%TRN	Transaction Number
%SYSID	System Name
%ACB	VTAM ACB Name (z/OS only)

Example:

```
FILE=c:\%USER\abc123.%GDATEC.%TIMET
```

Would be changed to:

```
FILE=c:\john\abc123.20050718.1601029
```

4.2.3 Example of CFAlias Configuration

The following example shows how to use the CFAlias feature to send a daily report to the specified directory, limits the user's access to the actual file name, and keep a record of the reports sent on a UNIX file server using MFT Platform Server on a Windows machine.

```
USERID=JohnDoe,
NODE=DEFAULT,
TYPE=RECEIVE,
FILE=c:\JohnDoe\DailyReports\report.%GDATE.doc,
ALIAS=report.doc

USERID=JohnDoe,
NODE=DEFAULT,
ALLOW=NO
```

Under this configuration, JohnDoe sends his daily report every day exactly the same way. Each time he sends his report to the server, the report is put in the **c:\JohnDoe\DailyReports\report.%GDATE.doc** file; therefore, each day the report has a different file name based on the current date. For example, if the date is July 18th, it is stored as the **report.030718.doc** file. Further, JohnDoe has no knowledge of where on the server his report is stored. Also, JohnDoe's aliased access only applies to a send of his report (because RECEIVE on the Responder is a SEND from the initiator). Finally, the second Alias grouping restricts JohnDoe from having any other access to the server with any file that is not report.doc.

4.2.4 Sample of CfAlias.cfg File

The following example is a sample CfAlias configuration file, called CfAlias.cfg by default.

```
*************************************************************************
*   This file contains the CFAlias Configuration parameters.            *
*   This file will be searched for parameters that match the            *
*   USERID and/or NODE.                                                 *
*                                                                       *
*   NOTE: This feature is only supported on the Responder side.         *
*                                                                       *
*   Allowable parameters are:                                           *
*      USERID= identifies the local user or DEFAULT for all users       *
*      NODE= identifies the node name or DEFAULT for all nodes          *
*      IPADDR= the IP address that initiated the transfer               *
*      TYPE= valid values are SEND, RECEIVE or BOTH                     *
*      FILE= fully qualified file name                                  *
*      ALIAS= name of file sent by initiator                            *
*      ALLOW= valid values are YES and NO, if no match on alias, user is *
*             allowed to define actual file name                        *
*                                                                       *
*   A grouping must have a USERID or NODE or IPADDR                     *
*   A grouping must have entries for both FILE and ALIAS or             *
*   an entry for ALLOW                                                  *
*   ALLOW and FILE/ALIAS are mutually exclusive                         *
*                                                                       *
*   NOTE:                                                               *
*   A line can be commented with a * or // or # at the beginning of line.*
*   There can only be one parameter per line.                           *
*   Parameters will be considered as part of the same grouping if the line *
*   ends with a comma.  The last line in a grouping MUST NOT contain a  *
*   comma.                                                              *
*   Each grouping must contain a USERID or NODE or both.                *
*                                                                       *
*   Requests are processed in the order that they are defined.  The first *
*   config entry that matches the transfer USERID and/or NODE is used.  *
*                                                                       *
*************************************************************************
USERID=Admin,
NODE=DEFAULT,
IPADDR=165.16.93.114,
TYPE=SEND,
FILE=/home/johnd/files/remotefile,
ALIAS=monthlysales

IPADDR=22.163.19.177,
TYPE=SEND,
ALLOW=no
```

4.3 CFINQ

This section describes the logging function and how to query past transactions. The log file stores the basic parameters of a transfer, but little information about how the transaction deals with the exception of Success/Failure and any error/success messages. Logging happens on every transfer, which helps maintain records with little overhead to the system.

4.3.1 Log Files

TIBCO MFT Platform Server for Windows has comprehensive logging to provide information about transfers that are initiated as well as transfers that are received on the Windows platform.

The platform server provides a common log to record this information from both the initiator and the responder. A new log file (Log.txt.yyyymmdd) is created each day with the date appended to the end of the file name entered in the configuration file. This file is accessed when inquiring on transactions by using the cfinq utility as well as by MFT Command Center. It is a standard ASCII text file that contains one record per line. The logs are located in the MFT Platform Server\Trace directory by default. The following example shows a sample daily log:

```
VersionNumber=7.2,LocalFileName=D:\send.txt,RemoteFileName=d:\receive.txt,Prior
ity=Normal,LocalTranNumber=IC05500000,RemoteTranNumber=RC05500001,TransferStart
Time=151119,TransferStartDate=20151205,TransferEndTime=151120,TransferEndDate=2
0151205,TransferDirection=Send,TransferWork=File,TransferCommand=N/A,TransferPr
ocessName=Name,TransferScheduleDate=N/A,TransferScheduleTime=N/A,TransferExpira
tionDate=N/A,TransferExpirationTime=N/A,CompressionType=None,CompressedBytes=0,
ConvertCRLF=no,EBCDICTranslate=no,SSL=no,SSLPortNumber=N/A,EncryptionType=N/A,R
ecordFormat=N/A,FileCreateOptions=CreateReplace,FileAttributes=N/A,UNIXFilePerm
issions=N/A,AllocationType=N/A,AllocationDirectory=N/A,AllocationPrimary=0,Allo
cationSecondary=0,Volume=N/A,Unit=N/A,NodeClass=N/A,StorClass=N/A,MgtClass=N/A,
DataClass=N/A,BlockSize=0,RecordLength=0,UserData=N/A,LocalUserid=tw,LogonDomai
n=N/A,RemoteUserid=tw,RemoteNodeName=fanwindows,RemoteNodeType=Node,RemotePortN
umber=46464,TryCount=1,TryMaxCount=1,ByteCount=0,RecordCount=1,MemberCount=N/A,
CheckPointCount=0,CheckPointRestart=N,CheckPointInterval=5,StatusMsg=Transfer
Completed
Successfully.,CrlMsg=N/A,StatusDiagCode=00,StatusSeverity=00,StatusReturnCode=N
/A,TransferStatus=Success,LocalCTFile=N/A,RemoteCTFile=N/A,TempError=No,PPA1Act
ion=N/A,PPA1Source=N/A,PPA1Status=N/A,PPA1Data=N/A,PPA1ReturnCode=N/A,PPA2Actio
n=N/A,PPA2Source=N/A,PPA2Status=N/A,PPA2Data=N/A,PPA2ReturnCode=N/A,PPA3Action=
N/A,PPA3Source=N/A,PPA3Status=N/A,PPA3Data=N/A,PPA3ReturnCode=N/A,PPA4Action=N/
A,PPA4Source=N/A,PPA4Status=N/A,PPA4Data=N/A,PPA4ReturnCode=N/A,EmailSuccessAdd
r=N/A,EmailFailureAddr=N/A,Accelerate=N/A,ACCProtocol=N/A,ACCEncryption=N/A,ACC
Compression=N/A,ACCMaxSpeed=N/A,ACCHost=N/A,ACCPort=N/A,SecurityPolicy=None,Rem
oveTrailingSpaces=N,ScanSubDir=N,ClassOfService=N/A,MaintainBDW=N/A,MaintainRDW
=N/A,RetentionPeriodExpirationDate=N/A,NodeWinners=10,
```

4.3.2 CFINQ Program

The MFT Platform Server inquiry program, CFINQ, provides two ways of showing the audit information to a user in a more convenient and clearer way than a text editor. This can be by summary or detailed views. The summary view consists of the following columns: Index, Transaction, Status, IP Address, and Local File.

The CFINQ program accepts the command-line parameters, which give the criteria for a specific query of the MFT Platform Server log files. CFINQ Parameters provides a list of the parameters that can be utilized for the execution of the CFINQ program. An equal sign can be used to separate the parameter from the value.

To obtain all transactions in the specified query, you must be either the administrator, or be in the cfbrowse or cfadmin group. Otherwise, you can view only your own transactions.

Note: The cfbrowse group is used for audit inquiry purposes alone and does not allow other rights that a user in the cfadmin group might have.

By default, 500 records are displayed in the CFINQ program. Use the MAXXFER parameter to increase or decrease the number of records to be viewed.

4.3.3 CFINQ Parameters

You can specify the following parameters on the CFINQ command line.

Parameter	Alternate Specification	Short Description
DAYS	None	Number of days to search
DESCRIPTION	DESCR	MFT Platform Server user data
ENDDATE	EDATE	End date
ENDTIME	ETIME	End time
EXCEPTIONS	EXC	Status of the transaction
IRFLAG	IRF	Initiator or Responder records
LOCALFILE	LF	Local file name
LOCALUSER	LUSER	Local user ID
LOCTRANSNUM	LTRN	Local transaction number
LOGDIR	LOGD	MFT Platform Server log files directory
MAXXFER	MAX	Number of transactions to list
PROCESS	PRO	MFT Platform Server process name
REMHOST	RHOST	Remote system name
REMTRANSNUM	RTRN	Remote transaction number
STARTDATE	SDATE	Start date
STARTTIME	STIME	Start time
TEMPERROR	TMPERR	Return temporary error records

DAYS

If **SDATE** and **EDATE** are both defined, this field is ignored. **DAYS** must not exceed 1826 (5 years). If **SDATE** is not defined, the start date equals the current date minus the number of days. If **SDATE** is not defined and **EDATE** is defined, the CFINQ program starts searching at the date calculated from (**EDATE** - # of **DAYS**) and ends at the **EDATE** date.

DESCRIPTION

This parameter defines the MFT Platform Server user data. When the **DESCRIPTION** parameter is specified, the CFINQ program searches for the MFT Platform Server log files and presents the detailed information for any transfers matching that description. A message is displayed on the screen if no transaction for the description is specified.

ENDDATE

The **ENDDATE** defines the end date in the format of *yyyymmdd.*

EDATE=TOD or EDATE=TODAY means today.

EDATE=YES or EDATE=YESTERDAY means yesterday.

If EDATE is not defined, the default value is TODAY.

ENDTIME

This parameter defines the end time in the 24 hour format of *hhmmss*. The default value is 240000. If **STIME** is not defined, the CFINQ program searches for the MFT Platform Server transaction only within 000000 - ETIME period.

EXCEPTIONS

This parameter defines the type of transfers to select.

U = Unsuccessful
S = Successful
Default = Successful and Unsuccessful

IRFLAG

This parameter defines the type of records to select.

B = Both
I = Initiator
R = Responder
Default = Both

LOCALFILE

This parameter defines the MFT Platform Server local file name.

LOCALUSER

This parameter defines the local user name (user ID). If you specify a user name other than your own, you must have security authorization.

LOCTRANSNUM

The **LOCTRANSNUM** parameter defines the unique local transaction number of the MFT Platform Server transfer. When the **LOCTRANSNUM** parameter is specified, the CFINQ program searches the MFT Platform Server log files and presents the detailed information for that transaction number. A message is displayed on the screen if no transaction for the **LOCTRANSNUM** is specified.

LOGDIR

This parameter defines the MFT Platform Server log files directory.

MAXXFER

This parameter defines the maximum number of requests that are returned. The default value is 500. The valid values are 1 to 100,000.

PROCESS

This parameter defines the MFT Platform Server process name.

REMHOST

This parameter defines the MFT Platform Server remote system name. This can be a node name, IP name, or IP address in dotted decimal notation. Generic selection cannot be used for IP addresses.

REMTRANSNUM

The **REMTRANSNUM** parameter defines the unique remote transaction number of the MFT Platform Server transfer. When the **REMTRANSNUM** parameter is specified, the CFINQ

program searches the MFT Platform Server log files and presents the detailed information for that remote transaction number. A message is displayed on the screen if no transaction for the remote transaction number is specified.

STARTDATE

The **STARTDATE** defines the start date of the search in the format of *yyyymmdd*.

SDATE=TOD or SDATE=TODAY means today.

SDATE=YES or SDATE=YESTERDAY means yesterday.

If **SDATE** is not defined, the default value is TODAY.

STARTTIME

This parameter defines the start time in the 24 hour format of *hhmmss*. The default value is 000000. If **ETIME** is not defined, the CFINQ program searches for the MFT Platform Server transaction only within STIME – 24 hour period.

TEMPERROR

This parameter defines whether temporary error records are selected.

Y = Yes
N = No
Default = Yes

4.3.4 Example of Using CFINQ Utility

This example shows how to use CFINQ on the command line. In this example, the start date is June 1, 2015 looking 20 days forward with a start time on 9:01 a.m. and an end time of 3 p.m. The local user is abc. Only successful transfers are listed, with a maximum of 1000 records.

The following command is entered:

```
cfinq sdate=20150601 days=20 stime=090100 etime=150000 luser=abc
lf=”c:\cfserver\log.txt” EXC=S max=1000
```

An equal sign can be used to separate the parameter from the value.

The output is as follows:

```
*************************************************************************
*****
         YOU HAVE ENTERED THE FOLLOWING VALUES FOR YOUR INQUIRY:
         LOCTRANSNUM.............[]
         REMTRANSNUM.............[]
         LOGDIR..................[]
         STARTDATE...............[20150601]
         ENDDATE.................[]
         DAYS....................[20]
         STARTTIME...............[090100]
         ENDTIME.................[150000]
         MAXXFER.................[1000]
         LOCALFILE...............[c:\cfserver\log.txt]
         LOCALUSER...............[abc]
         REMHOST.................[]
         DESCRIPTION.............[]
         PROCESS.................[]
         EXCEPTIONS..............[S]
         TEMPERROR...............[]
```

```
         INITRESPFLAG.............[]

*************************************************************************
******
***      PRESS [q] [enter] TO QUIT THE PROGRAM
***
***      PRESS [a] [enter] TO OBTAIN WHOLE RECORD LIST
***
***      PRESS [c] [enter] TO OBTAIN CURRENT RECORD LIST
***
***      PRESS [p] [enter] TO OBTAIN PREVIOUSLY VIEWED RECORD LIST
***
***      PRESS [m] [enter] TO OBTAIN MENU SCREEN
***
***      PRESS [n] [enter] or [enter] TO OBTAIN NEXT RECORD LIST
***
***      PRESS [h] or [?] [enter] TO OBTAIN HELP SCREEN
***
***      PRESS [index # ] [enter] TO OBTAIN DETAILED RECORD INFORMATION
***
*************************************************************************
*****
==>
```

To view the transactions, select one of the following menu options.

Menu Option	Short Description
a	Whole record list
c	Current record list
h or **?**	Help
index #	Detailed record information
m	Menu
n	The next record list. Pressing **Enter** without entering a value also display the next record.
p	Previous record list
q	Quit

To view the next 20 transactions for the sample above, select n and the records are as follows:

```
INDEX TRANSACTION STATUS  IPADDRESS             LOCALFILENAMEDIRECTORY
*************************************************************************
   1  I629500007  Success 111.22.33.44:46464   c:\cfserver\log.txt
   2  I629500009  Success 111.22.33.44:46464   c:\cfserver\log.txt
   3  I629500011  Success 111.22.33.44:46464   c:\cfserver\log.txt
   4  I629500012  Success 111.22.33.44:46464   c:\cfserver\log.txt
   5  I629500013  Success 111.22.33.44:46464   c:\cfserver\log.txt
   6  I629500014  Success 111.22.33.44:46464   c:\cfserver\log.txt
   7  I629500015  Success 111.22.33.44:46464   c:\cfserver\log.txt
   8  I629500017  Success 111.22.33.44:46464   c:\cfserver\log.txt
   9  R629500020  Success 111.22.33.44:46464   c:\cfserver\log.txt
  10  I629500019  Success 111.22.33.44:46464   c:\cfserver\log.txt
  11  I629500021  Success 111.22.33.44:46464   c:\cfserver\log.txt
  12  R629500025  Success 111.22.33.44:46464   c:\cfserver\log.txt
  13  R629500027  Success 111.22.33.44:46464   c:\cfserver\log.txt
  14  R629500029  Success 111.22.33.44:46464   c:\cfserver\log.txt
  15  R629500032  Success 111.22.33.44:46464   c:\cfserver\log.txt
  16  R629500028  Success 111.22.33.44:46464   c:\cfserver\log.txt
  17  R629500033  Success 111.22.33.44:46464   c:\cfserver\log.txt
  18  R629500002  Success 111.22.33.44:46464   c:\cfserver\log.txt
  19  R629500000  Success 111.22.33.44:46464   c:\cfserver\log.txt
```

```
  20  R629500034  Success 111.22.33.44:46464   c:\cfserver\log.txt
  21  R629500031  Success 111.22.33.44:46464   c:\temp\DRIVE.DLL
```

To view the details on one of the records listed above, type the index number of the transaction and press Enter. In this example, index number 10 (bold) is selected to show the details of transaction I629500019.

```
**********************************************************************
RECORD:10                        INITIATOR
**********************************************************************
Version Number ..............7.2
Priority..................... Normal
Local Transaction Number.... I629500019
Remote Transaction Number... R629500020
Transfer Start Time......... 151119
Transfer Start Date......... 20151205
Transfer End Time........... 151120
Transfer End Date........... 20151205
Transfer Direction.......... Send
Transfer Work............... File
Transfer Command............ N/A
Transfer Process Name....... Name
Transfer Schedule Date...... N/A
Transfer Schedule Time...... N/A
Transfer Expiration Date.... N/A
Transfer Expiration Time.... N/A
Compression Type............ None
Compressed Bytes............ 0
Convert CRLF................ no
EBCDIC Translate............ no
SSL......................... no
SSL Port Number............. N/A
Encryption Type............. Blowfish_448
Record Format............... FixedBlock
File Create Options......... CreateReplaceNew
File Attributes............. N/A
UNIX File Permissions....... 666
Allocation Type............. N/A
Allocation Directory........ N/A
Allocation Primary.......... N/A
Allocation Secondary........ N/A
Volume...................... N/A
Unit........................ N/A
Stor Class.................. N/A
Mgt Class................... N/A
Data Class.................. N/A
Block Size.................. 0
Record Length............... 80
User Data.................... N/A
Logon Domain................ N/A
Local File Name............. c:\cfserver\log.txt
Local User ID............... abc
Remote File Name............ /home/remotefile
Remote User ID.............. xyz
Remote Node Name............ WindowsNode
Remote Port Number.......... 46464
Try Count................... 1
Try Max Count............... 1
Byte Count.................. 27
Record Count................ 1
Member Count................ N/A
Check Point Count........... 0
Check Point Restart......... N
```

```
Check Point Interval........ 5
Status Msg.................. Transfer Completed Successfully.
Crl Msg..................... N/A
Status Diag Code............ 00
Status Severity............. 00
Status Return Code..........N/A
Transfer Status............. Success
Node Class ..................N/A
Remote Node Type.............Node
LocalCTFile..................N/A
RemoteCTFile.................N/A
PPA1 Action..................N/A
PPA1 Source..................N/A
PPA1 Status..................N/A
PPA1 Data....................N/A
PPA1 ReturnCode..............N/A
PPA2 Action..................N/A
PPA2 Source..................N/A
PPA2 Status..................N/A
PPA2 Data....................N/A
PPA2 ReturnCode..............N/A
PPA3 Action..................N/A
PPA3 Source..................N/A
PPA3 Status..................N/A
PPA3 Data....................N/A
PPA3 ReturnCode..............N/A
PPA4 Action..................N/A
PPA4 Source..................N/A
PPA4 Status..................N/A
PPA4 Data....................N/A
PPA4 ReturnCode..............N/A
Temporary Error..............No
Email Success Address........N/A
Email Failure Address........N/A
Accelerator..................N/A
Accelerator Protocol.........N/A
Accelerator Encryption.......N/A
Accelerator Compression......N/A
Accelerator MaxSpeed.........N/A
Accelerator Host.............N/A
Accelerator Port.............N/A
Security Policy..............None
Remove Trailing Spaces.......N
Scan Subdirectories..........N
Class Of Service.............N/A
Node Winners.................10
```

To obtain online help, type **h** or **?** on the command line.

```
COMMAND-LINE PARAMETERS allowed
*********************************************************************
        LOCTRANSNUM=  or LTRN=   Defines the Local Transaction number
        REMTRANSNUM=  or RTRN=   Defines the Remote Transaction number
        LOGDIR=       or LOGD=   Defines the MFT Platform Server log files
                                  directory
        STARTDATE=    or SDATE=  Defines the Start date in format yyyymmdd
        ENDDATE=      or EDATE=  Defines the End date in format yyyymmdd
        DAYS=                    Defines number of days to process
        STARTTIME=    or STIME=  Defines the Start time in 24 hour format: hhmmss
        ENDTIME=      or ETIME=  Defines the End time in 24 hour format: hhmmss
        MAXXFER=      or MAX=    Defines the Maximum number of requests returned

        LOCALFILE=    or LF=     Defines the MFT Platform Server local file name
```

```
        LOCALUSER=    or LUSER=  Defines the Local User Name
        REMHOST=      or RHOST=  Defines the Remote System Name
        DESCRIPTION=  or DESCR=  Defines the MFT Platform Server
                                  USERDATA(DESCRIPTION)
        PROCESS=      or PRO=    Defines the MFT Platform Server Process name
        EXCEPTIONS=   or EXC=    Return Successful or Unsuccessful transfers
        TEMPERROR=    or TMPERR= [yes | no] Return temporary error records
        PRINT=        or PRI=    Print data without screen prompts
       IRFLAG=       or IRF=   Defines Initiator or Responder records
 ****************************************************************
```

Note:

- Navigation commands are case sensitive.
- Date is returned in an LIFO (Last In, First Out) order. If the number of records exceeds the value of maxxfer (default: 500), the most recent transfers are selected first. Older transfers might not be included in the presented transactions list.
- The entered date must be in this format: *YYYYMMDD***.**
- The CFINQ program does not accept any negative values.
- No space is allowed between the parameter name, equal sign, and the parameter value.

4.4 Configured Post Processing

With the Configured post processing feature, you can trigger any executable command (.bat, .com, .exe, and so on) upon the completion of a file transfer. This feature offers greater flexibility than user-exits through the use of parameters and argument substitution. A configuration file, containing the commands and their associated parameters, is searched upon the completion of a transfer. If the properties of the transfer match the parameters, the executable command is triggered.

4.4.1 Configuration Parameters

A sample of the Configured Post Processing file, CfgPostProc.cfg, is located in the MFT Platform Server directory.

The definition of each parameter is as follows:

Parameter	Description
COMMAND	Defines the file to be executed. This is a required parameter.
TYPE	Defines the type of the file transfer request. The following values can be defined by the TYPE parameter: • SEND • RECEIVE • BOTH
SOURCE	Defines the source of the file transfer request. The following values can be defined by the SOURCE parameter: • INITIATOR • RESPONDER • BOTH

STATUS	Defines whether a transfer request is successful or unsuccessful. The following values can be defined by the STATUS parameter: • SUCCESS • FAILURE • BOTH
FILENAME/DSN	Defines the fully qualified file name. This field is compared against the local file name in the file transfer request.
PROCESS	Defines the PROCESS name associated with the transfer request.
IPADDR	Defines the IP address of the machine that is communicating with MFT Platform Server.
NODE	Defines the NODE name of the transfer request. For initiator requests, this parameter is used when the NODE parameter is used on a request. For responder requests, the platform server scans the list of node for matches on the IP address. These entries are then matched against the value specified in the NODE parameter. When defining nodes in this file, make certain that you use the proper case as these files are case sensitive.

Example of CfgPostProc.cfg File

```
SUBMIT,COMMAND=loaddb.exe,
     TYPE=RECEIVE,
     STATUS=SUCCESS,SOURCE=RESPONDER,
     FILENAME=jan.slaes,
     NODE=ACCOUNTING,
     PROCESS=fusion

SUBMIT,COMMAND=cmdfile.txt,TYPE=SEND,
     STATUS=BOTH,SOURCE=INITIATOR,
     FILENAME=infile.txt,
     IPADDR=111.222.1.2
```

4.4.2 Argument Substitution

Transfer properties can be passed to the executable command as substitutable command line arguments. Enter any of the following argument names after the COMMAND entry in the configuration file.

For example:

```
COMMAND=cmdfile.txt &FILENAME &TYPE
```

The file name and type of the transfer request are substituted for &FILENAME and &TYPE and passed to the executable command as command line arguments.

Argument Name	Data Substituted
&TYPE	SEND or RECEIVE
&SOURCE	INITIATOR or RESPONDER
&STATUS	SUCCESS or FAILURE
&RC	Numeric return code (0 if successful)

&FILENAME or &DSN	Local file name
&PROCESS	Process name
&NODE	Node name (or NODE if no node is found)
&IPADDR	IP address
&TRN	Transaction number

4.5 Custom Code Page Conversion

This feature supports converting text files between various character-set specifications.

With MFT Platform Server, the following four conversion tables are provided:

Comtblg.classic	The old comtblg.dat file shipped with previous versions (before version 7.1).
Comtblg.cp037	Extended ASCII table that is based on IBM Code page 037.
Comtblg.cp1047	Extended ASCII table that is based on IBM Code page 1047.
Comtblg.dat	ASCII/EBCIDIC table used by the platform server at run time. (By default a copy of Comtblg.cp037)

Comtblg.dat contains the following information which converts data from ASCII to EBCDIC and vice versa:

```
00010203372D2E2F16050A0B0C0D0E0F
101112133C3D322618193F27221D351F
405A7F7B5B6C507D4D5D5C4E6B604B61
F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F
7CC1C2C3C4C5C6C7C8C9D1D2D3D4D5D6
D7D8D9E2E3E4E5E6E7E8E9BAE0BBB06D
798182838485868788899192939495 96
979899A2A3A4A5A6A7A8A9C04FD0A107
9F00000000000000000000000000000 0
00000000000000000000000000000000
41AA4AB100B26AB5BDB49A8A5FCAAFBC
908FEAFABEA0B6B39DDA9B8BB7B8B9AB
6465626663679E687471727378757677
AC69EDEEEBEFECBF80FDFEFBFCADAE59
4445424643479C485451525358555657
8C49CDCECBCFCCE170DDDEDBDC8D8EDF
002E2E2E2E2E2E2E2E2E0A2E2E0D2E2E
2E2E2E2E2E0A2E2E2E2E2E2E2E2E2E2E
2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E
2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E
20A0E2E4E0E1E3E5E7F1A22E3C282B7C
26E9EAEBE8EDEEEFECDF21242A293BAC
2D2FC2C4C0C1C3C5C7D1A62C255F3E3F
F8C9CACBC8CDCECFCC603A2340273D22
D861626364656667686 9ABBBF0FDFEB1
B06A6B6C6D6E6F707172AABAE6B8C680
B57E737475767778797AA1BFD0DDDEAE
5EA3A5B7A9A7B6BCBDBE5B5DAFA8B4D7
7B41424344454647484 9ADF4F6F2F3F5
7D4A4B4C4D4E4F505152B9FBFCF9FAFF
5CF7535455565758595AB2D4D6D2D3D5
303132333435363738 39B3DBDCD9DA2E
```

The first sixteen lines are the ASCII-EBCDIC translation table, and the next 16 lines are the EBCDIC-ASCII translation table.

4.5.1 ASCII to EBCDIC Conversion Table Example

Each ASCII or EBCDIC character is represented by 2 hexadecimal digits. For example, ASCII character "E" is hexadecimal 45 or X’45’. The following table is the ASCII to EBCDIC translation table. The first hexadecimal digit of ASCII character "E" is 4, so you can go down the table to the row marked 4x. The second hexadecimal digit is 5, so you can move across to the x5 column and in that box is X’C5’.

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	00	01	02	03	37	**2D**	2E	2F	16	05	0A	0B	0C	0D	0E	0F
1x	10	11	12	13	3C	**3D**	32	26	18	19	3F	27	22	1D	35	1F
2x	40	5A	7F	7B	5B	**6C**	50	7D	4D	5D	5C	4E	6B	60	4B	61
3x	F0	F1	F2	F3	F4	**F5**	F6	F7	F8	F9	7A	5E	4C	7E	6E	6F
4x	**7C**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**
5x	D7	D8	D9	E2	E3	**E4**	E5	E6	E7	E8	E9	AD	E0	BD	5F	6D
6x	79	81	82	83	84	**85**	86	87	88	89	91	92	93	94	95	96
7x	97	98	99	A2	A3	**A4**	A5	A6	A7	A8	A9	C0	6A	D0	A1	07
8x	9F	00	00	00	00	**00**	00	00	00	00	00	00	00	00	00	00
9x	00	00	00	00	00	**00**	00	00	00	00	00	00	00	00	00	00
Ax	41	AA	4A	B1	00	**B2**	6A	B5	BD	B4	9A	8A	5F	CA	AF	BC
Bx	90	8F	EA	FA	BE	**A0**	B6	B3	9D	DA	9B	8B	B7	B8	B9	AB
Cx	64	65	62	66	63	**67**	9E	68	74	71	72	73	78	75	76	77
Dx	AC	69	ED	EE	EB	**EF**	EC	BF	80	FD	FE	FB	FC	AD	AE	59
Ex	44	45	42	46	43	**47**	9C	48	54	51	52	53	58	55	56	57
Fx	8C	49	CD	CE	CB	**CF**	CC	E1	70	DD	DE	DB	DC	8D	8E	DF

An ASCII character "P" is X’50’. Go down the table to row 5x and move across to column x0 and in the box is X’D7’. Therefore, X’50’ is translated to X’D7’.

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	**00**	01	02	03	37	2D	2E	2F	16	05	0A	0B	0C	0D	0E	0F
1x	**10**	11	12	13	3C	3D	32	26	18	19	3F	27	22	1D	35	1F
2x	**40**	5A	7F	7B	5B	6C	50	7D	4D	5D	5C	4E	6B	60	4B	61
3x	**F0**	F1	F2	F3	F4	F5	F6	F7	F8	F9	7A	5E	4C	7E	6E	6F
4x	**7C**	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	D4	D5	D6
5x	**D7**	**D8**	**D9**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**	**E9**	**AD**	**E0**	**BD**	**5F**	**6D**
6x	**79**	81	82	83	84	85	86	87	88	89	91	92	93	94	95	96
7x	**97**	98	99	A2	A3	A4	A5	A6	A7	A8	A9	C0	6A	D0	A1	07
8x	**9F**	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
9x	**00**	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Ax	**41**	AA	4A	B1	00	B2	6A	B5	BD	B4	9A	8A	5F	CA	AF	BC
Bx	**90**	8F	EA	FA	BE	A0	B6	B3	9D	DA	9B	8B	B7	B8	B9	AB
Cx	**64**	65	62	66	63	67	9E	68	74	71	72	73	78	75	76	77
Dx	**AC**	69	ED	EE	EB	EF	EC	BF	80	FD	FE	FB	FC	AD	AE	59
Ex	**44**	45	42	46	43	47	9C	48	54	51	52	53	58	55	56	57
Fx	**8C**	49	CD	CE	CB	CF	CC	E1	70	DD	DE	DB	DC	8D	8E	DF

EBCDIC to ASCII translation works the same way, but uses the lower 16 lines of the comtblg.dat file.

EBCDIC character "Z" is X’E9’. Go down the table to row Ex and move across to column x9 and in the box is X’5A’. Therefore, X’E9’ is translated to X’5A’.

	X0	**X1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	00	2E	2E	2E	2E	2E	2E	2E	2E	**2E**	0A	2E	2E	0D	2E	2E
1x	2E	2E	2E	2E	2E	0A	2E	2E	2E	**2E**	2E	2E	2E	2E	2E	2E
2x	2E	2E	2E	2E	2E	2E	2E	2E	2E	**2E**	2E	2E	2E	2E	2E	2E
3x	2E	2E	2E	2E	2E	2E	2E	2E	2E	**2E**	2E	2E	2E	2E	2E	2E
4x	20	A0	E2	E4	E0	E1	E3	E5	E7	**F1**	A2	2E	3C	28	2B	7C
5x	26	E9	EA	EB	E8	ED	EE	EF	EC	**DF**	21	24	2A	29	3B	AC
6x	2D	2F	C2	C4	C0	C1	C3	C5	C7	**D1**	A6	2C	25	5F	3E	3F
7x	F8	C9	CA	CB	C8	CD	CE	CF	CC	**60**	3A	23	40	27	3D	22
8x	D8	61	62	63	64	65	66	67	68	**69**	AB	BB	F0	FD	FE	B1
9x	B0	6A	6B	6C	6D	6E	6F	70	71	**72**	AA	BA	E6	B8	C6	80
Ax	B5	7E	73	74	75	76	77	78	79	**7A**	A1	BF	D0	DD	DE	AE
Bx	5E	A3	A5	B7	A9	A7	B6	BC	BD	**BE**	5B	5D	AF	A8	B4	D7
Cx	7B	41	42	43	44	45	46	47	48	**49**	AD	F4	F6	F2	F3	F5
Dx	7D	4A	4B	4C	4D	4E	4F	50	51	**52**	B9	FB	FC	F9	FA	FF
Ex	**5C**	**00**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**5A**	**B2**	**D4**	**D6**	**D2**	**D3**	**D5**
Fx	30	31	32	33	34	35	36	37	38	**39**	B3	DB	DC	D9	DA	2E

EBCDIC character ")" is X’5D’. Go down the table to row 5x and move across to column xD and in the box is X’29’. Therefore, X’5D’ is translated to X’29’.

	X0	**X1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	00	2E	2E	2E	2E	2E	2E	2E	2E	2E	0A	2E	2E	**0D**	2E	2E
1x	2E	2E	2E	2E	2E	0A	2E	2E	2E	2E	2E	2E	2E	**2E**	2E	2E
2x	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	**2E**	2E	2E
3x	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	**2E**	2E	2E
4x	20	A0	E2	E4	E0	E1	E3	E5	E7	F1	A2	2E	3C	**28**	2B	7C
5x	**26**	**E9**	**EA**	**EB**	**E8**	**ED**	**EE**	**EF**	**EC**	**DF**	**21**	**24**	**2A**	**29**	**3B**	**AC**
6x	2D	2F	C2	C4	C0	C1	C3	C5	C7	D1	A6	2C	25	**5F**	3E	3F
7x	F8	C9	CA	CB	C8	CD	CE	CF	CC	60	3A	23	40	**27**	3D	22
8x	D8	61	62	63	64	65	66	67	68	69	AB	BB	F0	**FD**	FE	B1
9x	B0	6A	6B	6C	6D	6E	6F	70	71	72	AA	BA	E6	**B8**	C6	80
Ax	B5	7E	73	74	75	76	77	78	79	7A	A1	BF	D0	**DD**	DE	AE
Bx	5E	A3	A5	B7	A9	A7	B6	BC	BD	BE	5B	5D	AF	**A8**	B4	D7
Cx	7B	41	42	43	44	45	46	47	48	49	AD	F4	F6	**F2**	F3	F5
Dx	7D	4A	4B	4C	4D	4E	4F	50	51	52	B9	FB	FC	**F9**	FA	FF
Ex	5C	00	53	54	55	56	57	58	59	5A	B2	D4	D6	**D2**	D3	D5
Fx	30	31	32	33	34	35	36	37	38	39	B3	DB	DC	**D9**	DA	2E

4.5.2 Making Your Own Tables

For other conversions besides standard ASCII to EBCDIC, you can define new tables. The provided table can be altered, or a completely new table can be defined.

For example, if you want to change the EBCDIC to ASCII translation of X'DE' to X'A3'. In the bottom half of the default table this translates to X'FA'. After the change, the table is as follows:

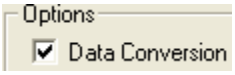
	X0	**X1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	00	2E	2E	2E	2E	2E	2E	2E	2E	2E	0A	2E	2E	0D	**2E**	2E
1x	2E	2E	2E	2E	2E	0A	2E	2E	2E	2E	2E	2E	2E	2E	**2E**	2E
2x	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	**2E**	2E
3x	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	**2E**	2E
4x	20	A0	E2	E4	E0	E1	E3	E5	E7	F1	A2	2E	3C	28	**2B**	7C
5x	26	E9	EA	EB	E8	ED	EE	EF	EC	DF	21	24	2A	29	**3B**	AC
6x	2D	2F	C2	C4	C0	C1	C3	C5	C7	D1	A6	2C	25	5F	**3E**	3F
7x	F8	C9	CA	CB	C8	CD	CE	CF	CC	60	3A	23	40	27	**3D**	22
8x	D8	61	62	63	64	65	66	67	68	69	AB	BB	F0	FD	**FE**	B1
9x	B0	6A	6B	6C	6D	6E	6F	70	71	72	AA	BA	E6	B8	**C6**	80
Ax	B5	7E	73	74	75	76	77	78	79	7A	A1	BF	D0	DD	**DE**	AE
Bx	5E	A3	A5	B7	A9	A7	B6	BC	BD	BE	5B	5D	AF	A8	**B4**	D7
Cx	7B	41	42	43	44	45	46	47	48	49	AD	F4	F6	F2	**F3**	F5
Dx	**7D**	**4A**	**4B**	**4C**	**4D**	**4E**	**4F**	**50**	**51**	**52**	**B9**	**FB**	**FC**	**F9**	**A3**	**FF**
Ex	5C	00	53	54	55	56	57	58	59	5A	B2	D4	D6	D2	**D3**	D5
Fx	30	31	32	33	34	35	36	37	38	39	B3	DB	DC	D9	**DA**	2E

If you also want the reverse conversion, you change the ASCII to EBCDIC section. Therefore, in the top half of the table, you can find row Ax and column x3 and change the value to X'DE'.

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	00	01	02	**03**	37	2D	2E	2F	16	05	0A	0B	0C	0D	0E	0F
1x	10	11	12	**13**	3C	3D	32	26	18	19	3F	27	22	1D	35	1F
2x	40	5A	7F	**7B**	5B	6C	50	7D	4D	5D	5C	4E	6B	60	4B	61
3x	F0	F1	F2	**F3**	F4	F5	F6	F7	F8	F9	7A	5E	4C	7E	6E	6F
4x	7C	C1	C2	**C3**	C4	C5	C6	C7	C8	C9	D1	D2	D3	D4	D5	D6
5x	D7	D8	D9	**E2**	E3	E4	E5	E6	E7	E8	E9	AD	E0	BD	5F	6D
6x	79	81	82	**83**	84	85	86	87	88	89	91	92	93	94	95	96
7x	97	98	99	**A2**	A3	A4	A5	A6	A7	A8	A9	C0	6A	D0	A1	07
8x	9F	00	00	**00**	00	00	00	00	00	00	00	00	00	00	00	00
9x	00	00	00	**00**	00	00	00	00	00	00	00	00	00	00	00	00
Ax	**41**	**AA**	**4A**	**DE**	**00**	**B2**	**6A**	**B5**	**BD**	**B4**	**9A**	**8A**	**5F**	**CA**	**AF**	**BC**
Bx	90	8F	EA	**FA**	BE	A0	B6	B3	9D	DA	9B	8B	B7	B8	B9	AB
Cx	64	65	62	**66**	63	67	9E	68	74	71	72	73	78	75	76	77
Dx	AC	69	ED	**EE**	EB	EF	EC	BF	80	FD	FE	FB	FC	AD	AE	59
Ex	44	45	42	**46**	43	47	9C	48	54	51	52	53	58	55	56	57
Fx	8C	49	CD	**CE**	CB	CF	CC	E1	70	DD	DE	DB	DC	8D	8E	DF

4.5.3 Additional Information

To activate the conversion tables, the Data Conversion box must be selected on the main transfer panel.



This uses the Comtblg.dat file for conversion. Comtblg.dat is located in the installation directory of MFT Platform Server. If the LocalCTFile and RemoteCTFile parameters are filled in under the Advanced Options tab, they are used instead. The platform server does not convert the file twice. The Comtblg.dat file is unaffected by the RemoteCTFile parameter.

In individual parameters, you can specify two conversion tables: one on the local side, and one on the remote side. In this way, you can have a standard character set to be used for transmission, without having a conversion table between every two possible character sets.

The local conversion table is specified with the LocalCTFile parameter in the GUI, and the LocalCTFile parameter on the command line. Similarly, the remote conversion table is specified with the RemoteCTFile parameter in GUI, and the RemoteCTFile parameter at the command line.

The maximum lengths of the LocalCTFile and RemoteCTFile parameters are 16 characters. However, they support file names relative to the current working directory on the local side. For Windows, the platform server looks in the MFT Platform Server working directory.

Nodes can also support both local and remote conversion tables. Unless the parameters are overridden on the command line, the associated conversion tables are used whenever that node is specified.

You must always replace a 2-digit hexadecimal number with a 2-digit hexadecimal number. If the table is invalid, conversion cannot be performed. The table consists of two sections with 16 lines each, therefore the entire file must have 32 lines across and 32 lines down. If it contains anything else, it does not work.

For all transfers, if the file is outgoing (a send transfer), the top half of the conversion table is used. If the file is incoming (a receive transfer), the bottom half of the conversion table is used. For example, in a send transfer, if both the LocalCTFile and RemoteCTFile parameters are used, the top half of the LocalCTFile is used on the local side, and the bottom half of the RemoteCTFile is used on the remote side. The reverse is true for a receive transfer.

To identify which table translates for send and which translates for receive, during editing, place a few lines between the two tables.

The ASCII character set in the default table supports the extended ASCII range which covers special characters outside the English alphabet. For standard ASCII support, you can use the comtblg.classic file. To replace the default table, rename the existing comtblg.dat file, and then rename the existing comtblg.classic file to become the new comtblg.dat file. The conversion tables currently available do not support wide or multi-byte character sets at present.

4.6 Directory Named Initiation (DNI) GUI and Command Line Interface

You can run a DNI job through both the DNI GUI interface and the command line interface.

4.6.1 DNI GUI Interface

By using Directory Named Initiation (DNI), you can transmit a file, print, or batch job by simply placing a file in a directory on a local drive or a network volume. By using the Directory Initiation property sheet, you can determine directory attributes and create and modify the DNI schedule. When the DNI entry dispatch is completed, you can leave the local file where it is, copy it to another directory, move it to another directory, or delete it. You can also retry a failed DNI job at the next time when the DNI job starts.

The following features and uses are supported by DNI:

- DNI can scan network volumes shared from Novell NetWare, UNIX, IBMi, and any network drive that can be viewed by using UNC (Universal Naming Convention). **Note**: Mapped drives are not supported.
- A DNI scan directory can be a single directory or a directory and all of its subdirectories.
- DNI directories are put on a flexible schedule. You can scan the directory at a time that you determine (for example, 5:00 p.m. on Fridays, the first day of every month, or every 30 seconds).
- DNI provides store and forward capabilities where the DNI scan directory is the destination of a platform server transfer. DNI scans the directory and forwards the file to another destination.
- DNI supports sequential distribution, where the copy or move disposition targets another DNI directory.
- When the DNI entry dispatch is completed and the disposition of the file is applied, a secondary Windows Event message indicates what happened to the original file.
- Up to 50 DNI scan directories are supported per platform server for Windows.

To use DNI, you must create a transfer template (see the following section on Transfer Templates), and then create a Directory Named Initiation entry related to that template (see the section on The Initiation Directories Properties Sheet).

Note:

- For optimum performance, an excessive number of DNI directories require another platform server for Windows. To improve performance, you must also adjust the dispatch time accordingly.
- It is best practice to define the DNI disposition as move or delete, because this reduces overhead associated with managing the leave file and clearly identifies which files are pending to be transferred and which file transfer failed.

4.6.1.1 Transfer Templates

A transfer template is a collection of any or all of the parameters required to perform a transfer.

Each DNI entry is associated with a transfer template. The transfer template describes the name of the remote system (with a DNS name or TCP/IP address) and DNI entry options, such as compression, check point restart, or character conversion, and dynamic allocation parameters for remote z/OS systems.

You can associate any number of DNI entries with a transfer template. Perform this if you want multiple DNI directories to communicate with the same remote system.

The transfer template also describes the name of the remote data set or file, but if the remote file name is not dynamic (for example, 'SYS1.USERDATA'), every DNI entry will overwrite the data of the previous transfer. TIBCO MFT Platform Server for Windows provides dynamic file name creation through the use of file name tokens (see File Name Tokens). Use this feature in the Remote File Name field of the template to create unique file names for files transmitted from a DNI directory.

To create a transfer template, complete the following steps:

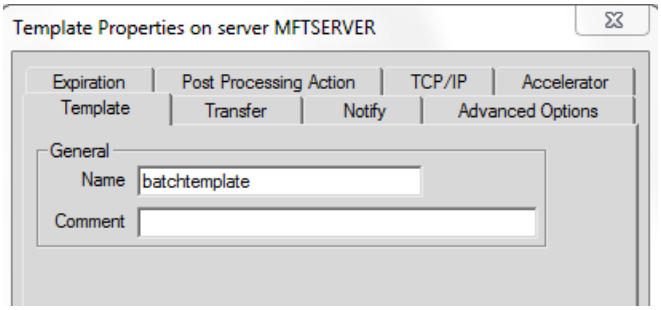
1. From the right-hand panel, click **Templates** to highlight it.
2. Right-click **Templates** and click **New > Advanced TCP Template**.
3. Complete any or all of the property pages for this transfer template as you will for any other transfer, and then complete the **Directory Initiation Properties** sheet.

You can define 2 different types of templates, TCP templates and Batch templates.

4.6.1.1.1 *Advanced TCP Template*

The TCP template property pages are similar with the transfer property pages.

1. **Template Tab**



- **Name**

  The name used to identify this template. Ensure not to use spaces.

- **Comment**

  This field is optional. A comment can be used to give more description to your template. The maximum length of the comment is 64 byte.

2. **Transfer Tab**
   See Transfer Tab section for more information on the parameters of this tab.

3. **Notify Tab**
   See Notify Tab section for more information on the parameters of this tab.

4. **Advanced Options Tab**
   See Advanced Options Tab section for more information on the parameters of this tab.

5. **Expiration Tab**
   See Expiration Tab for more information on the parameters of this tab.
   Note: The expiration date option is not usable for templates.

6. **Post Processing Action Tab**
   See Post Processing Action Tab section for more information on the parameters of this tab.

7. **TCP/IP Tab**
   See TCP/IP Tab section for more information on the parameters of this tab.

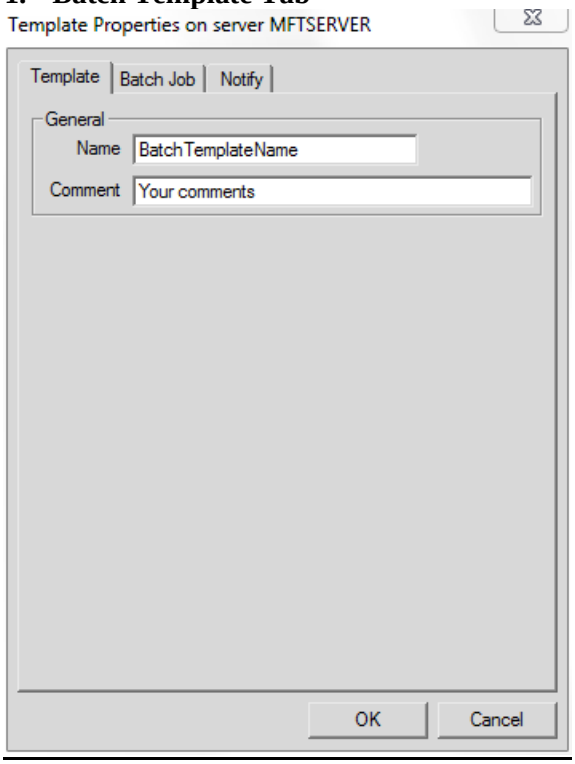
8. **Accelerator Tab**
   See Accelerator Tab section for more information on the parameters of this tab.

4.6.1.1.2 *Advanced Batch Template*

The Batch Template can be used to execute jobs on every modified file or a new file in the directory specified as the DNI directory. When the job specified in the Advanced Batch Template is executed, an email can be sent (if email notification is chosen). This email states that the Create Process is successful. This does not mean that the job is executed successfully. The results of the job executed are logged into the Event log. The output of the job is redirected to a file named "FtmsCp.trc" under the trace directory.

Note: The job is executed in the \WINDOWS\SYSTEM32 directory. Ensure that when writing your batch jobs in the event, you must change the directory in which the batch job is executed.

1. **Batch Template Tab**



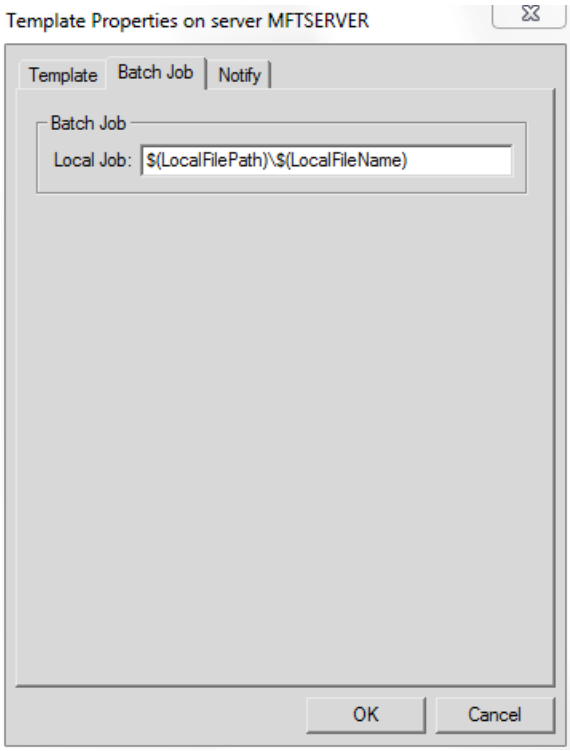
- **Name**

  The name used to identify this template.

- **Comment**

  This field is optional. A comment can be used to give more description to your template.

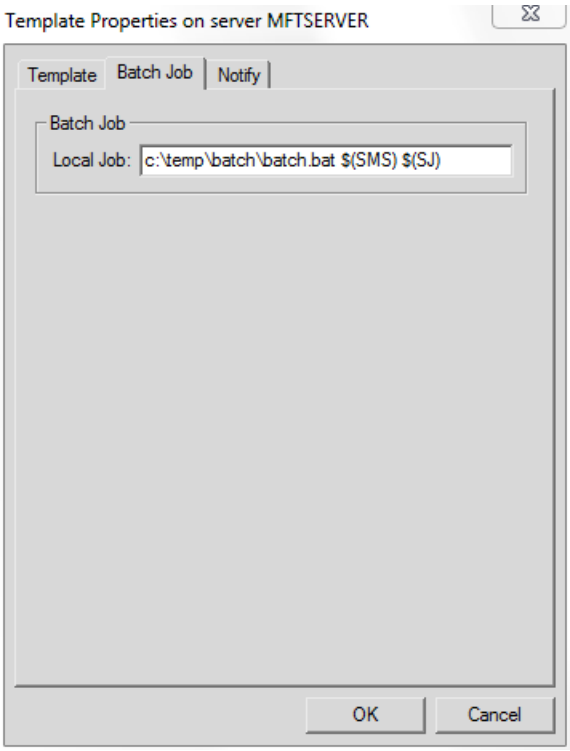
2. **Batch Job Tab**



- **Local Job**

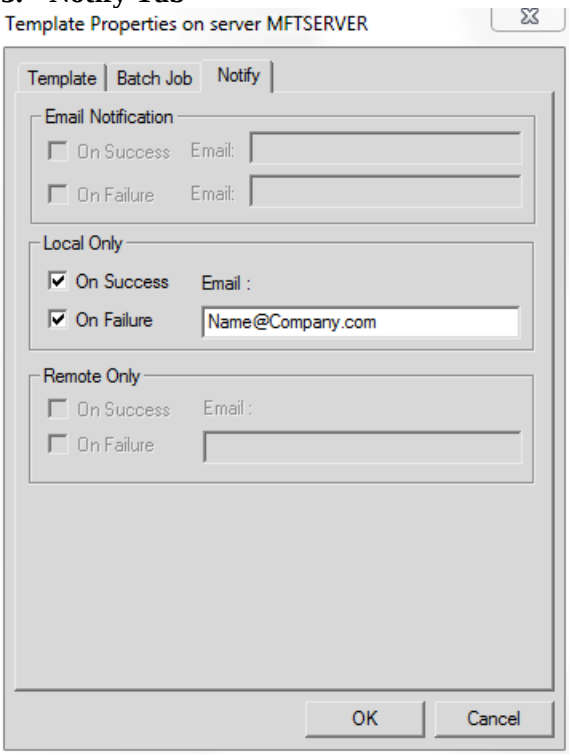
  This is the job that you want to execute when a file is placed in the defined DNI directory.

You must input the whole command line for a particular job. You can specify the file name tokens as input to the command line. The following figure shows an example of the Batch Job Template being used to execute a job with variable parameters. The parameters entered are MFT Platform Server substitutable tokens for milliseconds and Julian date.



When the input is specified as $(LocalFileName), only the file name without its path is used as input. If the whole path is required, specify $(LocalFilePath)\$(LocalFileName).

3. Notify Tab



- **Local Only**

  On Success

Select this check box to send notification to the local user when the transfer succeeds.

On Failure

Select this check box to send notification to the local user when the transfer fails.

Email

This is the email address to notify when a transaction is completed. It informs the user whether the transaction is successful or not.

4.6.1.1.3 *File Name Tokens*

You can create dynamic file names through the use of substitutable tokens (file name tokens) embedded within the name of the local or remote file names.

When creating a transfer template for sending a file from DNI, use the file name tokens in the remote file name field.

Example:

In this example, a remote file name is created by using file name tokens. The local file name is left blank because the name and path of the file in the DNI directory are substituted in the local file name field automatically. The entire remote file name is as follows:

```
prx0115.$(LocalFileBase).$(SMon)$(SDD)
```

The platform server resolves the file name tokens into a file name based on the base name of the DNI file, current month and day. For example, if the DNI file is named file001.dat and today's date is June 10, the generated file name is as follows:

```
prx0115.file001.Jun10
```

For a full list of file name tokens, see the File Name Tokens appendix.

4.6.1.2 The Initiation Directories Properties Sheet

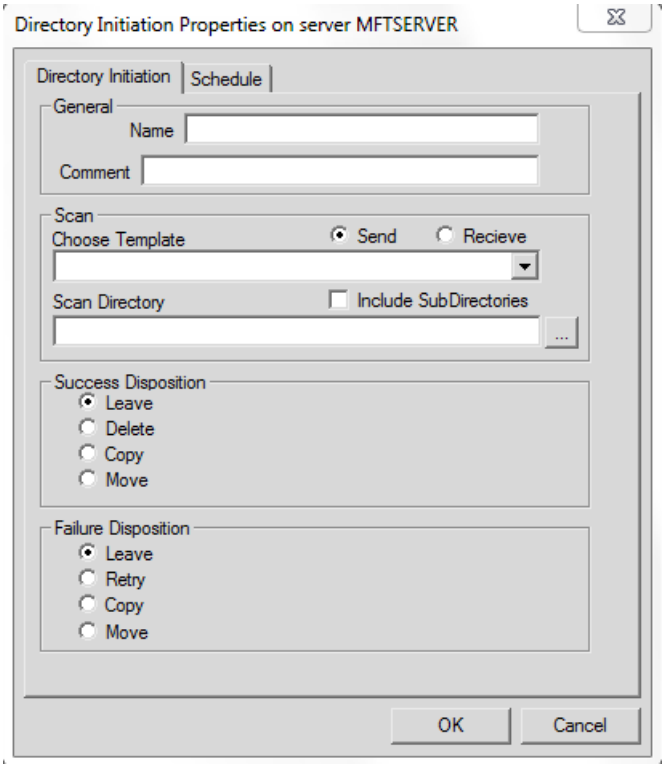
You can use the Initiation Directories Properties sheet to create and maintain DNI information. This sheet contains two property pages: Directory Initiation and Schedule.

You can create DNI entries with or without a schedule. DNI entries without schedules are dispatched every time the dispatcher of MFT Platform Server becomes active.

Note: Scans run against the directory are done by using the Service account of the platform server instead of the local user account that is defined in a template or presently logged into the system.

When creating a new DNI definition, if the scan directory does not exist, an error message will be displayed.

4.6.1.2.1 *Directory Initiation Property Page*



- **General**

  Name

  A character string which uniquely identifies the DNI entry.

  Comment

  A free text description of the entry. The maximum length of this field is 64 bytes.

- **Scan**

  Choose Template

  The name of the transfer template that is used to create the DNI entry. The template must exist on the same MFT Platform Server where the DNI entry is stored. The Choose Template drop-down list contains a list of the templates that are defined on that MFT Platform Server.

  - **Send**: displays all the send templates in the drop-down list.
  - **Receive**: displays all the receive templates in the drop-down list.

  Scan Directory

  The name of the directory to scan for files. This can be the local directory for a send transfer or a remote directory for a receive transfer.

  Note: File name tokens are not supported in this field. Do not end the path with a forward slash or backslash.

  Include SubDirectories

  When selected, the service scans the DNI directory for files as well as all the subdirectories beneath the DNI directory.

- **Success Disposition**

  In this section, you can define the operation to apply to the scanned file after the DNI entry is dispatched. You can select the following options:

  - **Leave** the file where it is
  - **Delete** the file
  - **Copy** the file to another directory
  - **Move** the file to another directory

  If the disposition is Copy or Move, the **Copy To/Move To Directory** field also becomes available.

  Note: It is best practice to define the DNI disposition as Move or Delete, as this reduces overhead associated with managing the leave file and clearly identifies which files are pending transfer versus failed.

  Copy To/Move To Directory

  This field is displayed when the Copy or Move disposition is selected. This field indicates the directory where the source file is placed when the DNI entry is dispatched. This is especially useful for DNI entries which are configured to receive files.

- **Failure Disposition**

  In this section, you can specify the operation to apply to the scanned file after the DNI entry is dispatched. You can select the following options:

  - **Leave** the file where it is.
  - **Retry** the file transfer when next time the DNI job runs.
  - **Copy** the file to another directory.
  - **Move** the file to another directory.

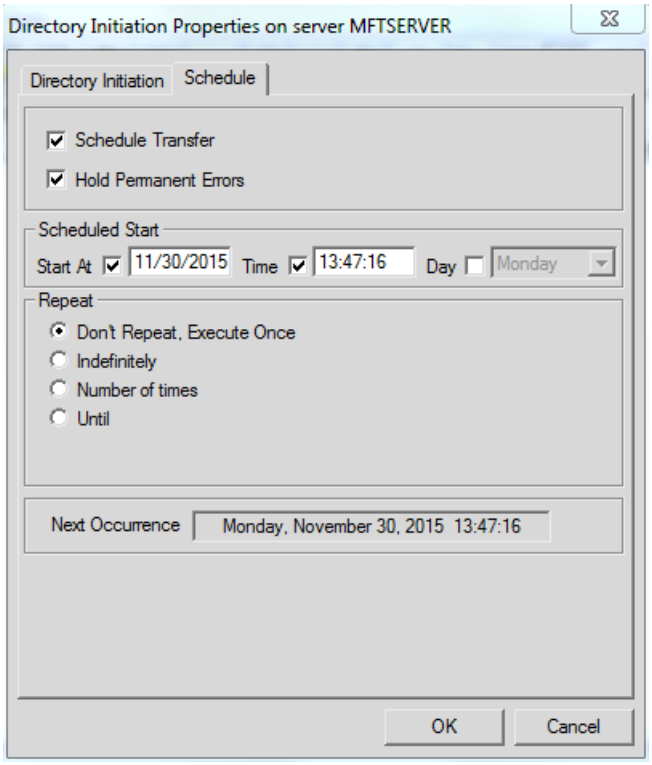
  If the disposition is Copy or Move, the **Copy To/Move To Directory** field also becomes available.

  Note: When the Retry disposition is used, if the file transfer fails for any reason, the transfer will be retried until it is successful.

  Copy To/Move To Directory

  This field is displayed when the Copy or Move disposition is selected. This field indicates the directory where the source file is placed when the DNI entry is dispatched. This is especially useful for DNI entries which are configured to receive files.

4.6.1.2.2 *Schedule Property Page*



You can use the Schedule property page to schedule DNI activity.

- **Schedule Transfer**

  Adds a schedule to the DNI entry.

- **Hold Permanent Errors**

  Puts a scheduled transfer on hold if a permanent error occurs. If this option is not selected, the system continues to try the transfer even after a permanent error occurred. Examples of permanent errors include the remote file not existing, bad user ID or password.

- **Scheduled Start**

  Scheduled Start provides the information necessary for dispatching a DNI entry in the future. This parameter includes the following three fields.

  Start At

  This field specifies the dispatch date for the DNI entry. This defaults to the current date. This entry is mutually exclusive with the value in the Day (day of week) field.

  Time

  This field specifies a particular time to dispatch the DNI entry. This defaults to the current time.

  Day

  This field specifies a particular day of the week to dispatch the DNI entry. This entry is mutually exclusive with the value in the Start At (date) field.

  Note: To apply the information in these fields to the DNI entry, you must select the check box at the left of each of the fields.

- **Repeat**

  Provides information relative to the future dispatching (if any) of a particular file transfer after the file transfer has already been executed once.

  Don’t Repeat, Execute Once

  When this option is selected, the system only attempts to dispatch this DNI entry for a single time.

  Indefinitely

  When this option is selected, the Interval field is displayed on the panel. The DNI entry is to be dispatched indefinitely (or until the current user or administrator deletes the job) and in accordance with the information specified in the Start At field and in the Interval field.

  Number of times

  This option specifies the number of times the DNI entry can be dispatched before it is removed from the queue. The range for this field is from 2 to 32767.

  Similar to the Indefinitely option, the Number of times option invokes the Interval field.

  Until

  You can specify the date and time or the day of the week until which the DNI entry is dispatched. When this option is selected, the fields (similar to the Start At field) where you can specify the required information are displayed in the panel.

  Interval

  This parameter is selectable if you specify a Repeat option (with the exception of Don’t Repeat, Execute Once). From the drop-down list, you can select Daily 7 (Sunday to Saturday), Weekly, Bi-Weekly, Monthly, Bi-Monthly, Quarterly, Semi-Annually, Annually, Bi-Annually, or Every.

  When Every is selected, two additional fields that you can use to specify the frequency with which you want to dispatch the DNI entry are added to the Interval option. You can enter a number in the first field. The second field contains a drop-down list which contains seconds, minutes, hour(s), day(s), week(s), month(s), and year(s).

  Next Occurrence

  This read-only field indicates the next time the schedule will dispatch the DNI entry.

When new templates or new DNI are created, they are backed up to a file called "FTMSSVR.BAK" which is located in the directory where MFT Platform Server is installed. During the uninstallation process, if you choose not to remove all the application configuration data, the FTMSSVR.BAK file is not deleted; therefore, you can copy this file to some other directory, and then reinstall the product. If this file is then put into the MFT Platform Server installation directory and renamed to "FTMSSVR.PQF", all the previously defined templates and DNI can be restored.

4.6.2 DNI Command Line Interface (CLI)

Directory Named Initiation (DNI) supports detecting the existence of files that are placed within a directory and/or subdirectories and automatically transferring those files to one or more targeted MFT Platform Server remote systems.

For more information, see *TIBCO Perl Directory Named Initiation (DNI) Installation and Operations Guide* contained within the dni.tar file, which is located in the MFT Platform Server installation directory. Use a file expansion utility, such as WinZip or 7-Zip, to extract the dni.tar file.

Note: DNI processing is done by using a Perl script called dni. As such, to use DNI, your Windows systems must have a version of Perl installed. The Perl program directory must be defined in the Windows PATH environment variable. If you do not have Perl installed on your computer, it can be downloaded for free from the following website: www.perl.org.

You can also manage Perl DNI job through TIBCO MFT Command Center.

4.7 Directory Transfer and Wildcard Support

MFT Platform Server can send and receive directories. If the LocalFileName or RemoteFileName contains an asterisk (*) or question mark (?) after the last forward slash (/) in the file path, a directory transfer is performed.

4.7.1 Directory Transfer Parameters

ScanSubDir

This causes not only the directory from the file path to be scanned, but all subdirectories, as well. The default value for this is No.

StopOnFailure

If the current file transfer fails, the platform server does not try to transfer the rest of the files. The default value for this is Yes.

Test

You can display the local and remote file names to verify that the file names are correct rather than perform the actual transfers. The default value for this is No.

4.7.2 Tokens for Local and Remote File Names

$(SDIR)

This case-sensitive token can be used with a receive transfer as part of the LocalFileName path, and with a send transfer as part of the RemoteFileName path.

Example:

For a receive transfer, you can set as follows:

LocalFileName:

```
C:\johndoe\data\$(SDIR)\$(RemoteFileName)
```

RemoteFileName:

```
C:\TransferFiles\data\*
```

The text before this token is assumed to be a base directory.

If **ScanSubDir** is selected and files exist in both the remote directory (**C:\TransferFiles\data**) and in the remote subdirectories, the same subdirectories are created in the local directory (**C:\johndoe\data**) and local file names are given as **$(RemoteFileName)** token.

If this token is missing but **ScanSubDir** is selected, all the files from the remote directory and all subdirectories are located at the local base directory. Their names are given by the **$(RemoteFileName)** token.

SubDirectories are created with the same access rights as the base directory. If some of the directories do not exist at the base directory path (for example, directory **data** from **LocalFileName**), it is created with the same access as its base directory (**johndoe**), and all directories after it are created under it with the same access rights.

For a send transfer, **$(SDIR)** must be used as part of the **RemoteFileName** path, in the form of **C:\TransferFiles\data\$(SDIR)\$(LocalFileName)**.

If the remote side contains no subdirectory structures (as on z/OS), files from the remote side are placed in the local base directory and **$(SDIR)** is ignored.

$(MEMBER)

This token is used only for a receive transfer from a z/OS system. It is used for a similar purpose as the $(SDIR) token, but it is used because data set names work differently than directory names.

With this token, you can have file names on the local side that are the same as Member names on the z/OS side.

If the file name from the z/OS side contains no $(Member), this token is not used. For example, if the path is **C:\TransferFiles\$(MEMBER)\whatever**, it changes to **C:\TransferFiles\whatever**.

4.7.3 Wildcard Information

MFT Platform Server supports asterisk (*) and question mark (?) wildcards. They have exactly the same meaning on each platform as they do in the Operating System.

For Windows platforms, asterisk (*) means any number of any symbols in the file name, and question mark (?) means any one symbol in the file name. MFT Platform Server interprets these symbols if they are present in the file name after the last backslash (\). Any combination and amount of these symbols and alpha numeric characters can be used to narrow down the required files.

Only those files whose names satisfy the selection criteria are transferred.

For example, the name **c:\johndoe\r?t*** matches the files **c:\johndoe\returns** and **c:\johndoe\ratelist** but not the name **c:\johndoe\report.** To transfer an entire directory, a single * must be used.

4.7.4 General Information

- The file path can contain both backslash (\) and forward slash (/).
- Be aware of the **Creation Option** parameter. If no file with the same names specified in the receiving directory exists, the **Creation Option** must be **CreateReplace** or **CreateReplaceNew**.
- If subdirectories are transferred as well and file names that match the specified names in the receiving directory exist, the **Creation Option** must be **CreateReplaceNew**; if it is **CreateReplace**, the subdirectories are not created.
- Selected files are sent to or received from the remote side sequentially, each with a separate local and remote transaction numbers and thus separate entries in the log files. Only after a file transfer is completed will the next file be transferred.
- With the exception of the new parameters listed in the Directory Transfer Parameters section of this document, all parameters are applicable per file transfer.
- **Try Count** will be the same for all files in the directory and the next file will not be transferred until all attempts to transfer the previous one are made.
- **CheckPointRestart** works for the directory/wildcard transfers as well. If something occurs during the directory transfer and the MFT Platform Server Service is stopped, when the MFT Platform Server Service is restarted, it will finish the checkpointed file, and then continue with the rest of the files specified in the directory transfer.

4.8 fusping Utility

The fusping utility is used to find the status of a platform server running on a remote system.

4.8.1 Format of fusping Commands

The following example shows the usage of the fusping command:

```
usage: fusping parameters:[values]
[parameters]:
h: or Host and Port: - h:[IpAddress]:[PortNumber] or h:[IpName]:[PortNumber]
?: - Help
```

4.8.2 Examples of Using fusping Utility

The following examples show how to use the fusping utility to check whether MFT Platform Server is running on the remote system as well as the version of MFT Platform Server.

This example checks a remote mainframe platform:

```
fusping h:[11.22.33.55]:[46464]
```

Output:

```
Host:             11.22.33.55
Port:             46464
System Name:      Name=A390,STC=CFUSN65,CPUType=1234,CPUID=5555
Key Expiration:   20160516
Version:          MFT Platform Server z/OS,Version=720 ,PTFLevel=CZ01977:720
```

This example checks a remote Windows platform:

```
fusping h:[11.22.33.44]:[46464]
```

Output:

```
Host:             11.22.33.44
Port:             46464
System Name:      WIN44
Key Expiration:   Unknown
Version:
     Ftms32.DLL, Version 7.2 (Build 8 UNICODE)
     FtmsDni.DLL, Version 7.2 (Build 8 UNICODE)
     FtmsTcpS.DLL, Version 7.2 (Build 8 UNICODE)
     FtmsVer.DLL, Version 7.2 (Build 8 UNICODE)
     FusionMs.DLL, Version 7.2 (Build 8 UNICODE)
     HoLib.DLL, Version 7.2 (Build 8 UNICODE)
     HOTrace.DLL, Version 7.2 (Build 8 UNICODE)
     SMTPDll.DLL, Version 7.2 (Build 8)
     FtmsMgr.EXE, Version 7.2 (Build 8 UNICODE)
     FtmsCmd.EXE, Version 7.2 (Build 8 UNICODE)
     FtmsMon.EXE, Version 7.2 (Build 8 UNICODE)
     FtmsSvr.EXE, Version 7.2 (Build 8 UNICODE)
     FusionVer.EXE, Version 7.2 (Build 8 UNICODE)
```

4.9 fusutil Utility

When a file transfer is completed, you might want to perform some action such as renaming or deleting a file. All of the platforms have different commands to rename or delete a file. With this utility, you can use a common interface to rename or delete a file or directory, and to verify whether a file or directory exists on a remote platform.

The fusutil utility provides the following three functions:

- Delete a file or directory.
- Rename a file or directory.
- Verify whether a file or directory exists.

When a fusutil request is received by the platform server, the request must be converted to the proper request for that operating system. The following table shows the relationship between the fusutil command and the operating system command.

Function	Shortcut	Windows Equivalent Command
RENAME	R	move
DELETE	D	erase
EXIST	E	N/A

Note: One or more spaces must be added between the command parameters.

4.9.1 Format of fusutil Commands

The fusutil command must be configured as a post processing action by using the COMMAND option.

The first parameter after the COMMAND option is required and is the command name: fusutil.

The second parameter is required and is the function type:

Function	Shortcut	Description
RENAME	R	Renames a file.
DELETE	D	Deletes a file.
EXIST	E	Verifies whether a file exists.
RENAMEDIR	RDIR	Renames a directory.
DELETEDIR	DDIR	Deletes a non-empty directory recursively.
REMOVEDIR	RMDIR	Removes an empty directory only.
EXISTSDIR	EDIR	Verifies whether a directory exists.

The following examples show the syntax of using the fusutil utility as a post processing action:

```
Post_Action1: S,L,COMMAND,fusutil DELETE <filename>
or
Post_Action1: S,L,COMMAND,fusutil D <filename>

Post_Action2: F,R,COMMAND,fusutil RENAME <old_filename> <new_filename>
or
Post_Action2: F,R,COMMAND,fusutil R <old_filename> <new_filename>

Post_Action3: S,R,COMMAND,fusutil EXIST <filename>
or
Post_Action3: S,R,COMMAND,fusutil E <filename>

Post_Action4: S,L,COMMAND,fusutil DELETEDIR <directoryname>
or
Post_Action4: S,L,COMMAND,fusutil DDIR <directoryname>
```

```
Post_Action5: S,L,COMMAND,fusutil REMOVEDIR <directoryname>
or
Post_Action5: S,L,COMMAND,fusutil RMDIR <directoryname>

Post_Action6: S,R,COMMAND,fusutil EXISTSDIR <directoryname>
or
Post_Action6: S,R,COMMAND,fusutil EDIR <directoryname>

Post_Action7: F,R,COMMAND,fusutil RENAMEDIR <old_directoryname>
<new_directoryname>
or
Post_Action7: F,R,COMMAND,fusutil RDIR <old_directoryname> <new_directoryname>
```

Note: File names with embedded spaces must be enclosed in double quotation marks.

4.9.2 Special Processing

When processing the EXIST function, the platform server also checks whether the file is available for use. This is done on all platforms except UNIX, because no standard call is available to accomplish this on UNIX.

4.9.3 Return Codes

When the function is successful, the return code is set as 0, and any output data is returned to the caller, in the same way as any other command.

When the function is unsuccessful, the return code is set to a non-zero value, and a send error is returned to the caller along with a message indicating the cause of the failure (if possible).

4.10 Nodes, Profiles, and Distribution Lists

Nodes, user profiles, and distribution lists are used to define all information required to interact with a single or multiple MFT Platform Server remote systems. Therefore, you do not have to constantly provide information to MFT Platform Server when conducting transfers with remote systems.

Node definitions are used to define information about a remote system (node). They are stored in a file named cfnode.cfg located in the MFT Platform Server directory. The MFT Platform Server cfnode command, located in the MFT Platform Server System directory, is used to add and update node definitions to the cfnode.cfg file.

Local User Profile definitions are used to define a remote user name and remote password that can be used by a local user. Local User Profiles are stored in a file named cfprofile.cfg located in the MFT Platform Server directory. Passwords are stored in encrypted format to ensure maximum security. The MFT Platform Server cfprofile command is to be used to add and update profile definitions in the cfprofile.cfg file.

Responder Profiles define a local user name and password that are used in place of the incoming user name and password. By using responder profiles, a remote MFT Platform Server installation does not have to know an actual user name and password on your local machine to initiate a transfer.

Distribution Lists are used to conduct send transfers to multiple nodes at one time (**Note**: Distribution Lists support send requests only). The MFT Platform Server configuration file cflist.cfg is located in the MFT Platform Server installation directory.

4.10.1 Node Definitions

Node definitions define default parameters required by the platform server to interact with a remote system (node). The following information is included:

- Node name
- System type
- IP address or host name
- Port number
- (Optional) Security compliance level
- (Optional) Netmask for remote IP address
- (Optional) Netmask6 for remote IPv6 address
- (Optional) Use of SSL for secure communications
- (Optional) Default compression type
- (Optional) Default encryption type
- (Optional) Default local translation file
- (Optional) Default remote translation file
- (Optional) Whether responder profiles are used
- (Optional) Whether verified users are accepted
- (Optional) Text description for the node
- (Optional) Supported Command Center functions
- (Optional) Maximum initiator transfers

After a node definition is created, you can specify the name of the node to be used when executing a transfer. The platform server consults the definition for the specified node to obtain the parameters required to execute a transfer.

Node definitions are stored in a file named cfnode.cfg located in the MFT Platform Server directory. You must use the cfnode command to update the cfnode.cfg file. MFT Command Center can also be used to update the cfnode.cfg file. Before cfnode updates any information in cfnode.cfg, a backup of this file is created called cfnode.bak.

The following example shows a sample node definition created by using the cfnode command:

```
[dataServerA]
   SystemType                = Windows
   Protocol                  = tcpip
   RemoteLocation            = HostName
   HostName                  = 111.222.33.55
   Compression               = RLE
   Encryption                = NO
   RemoteCTFile              = rmttrans.txt
   Description               = This is a sample windows node definition

[dataServerB]
   SystemType                = Linux
   Protocol                  = tcpip
   HostName                  = 111.222.33.44
   Server                    = 56565
   SSL                       = Y
   Compression               = No
   Encryption                = No
   SecurityPolicy            = None
   ResponderProfile          = N
   Description               = Sample TCP node
   CommandSupport            = PING
   Winners                   = 2
```

4.10.1.1 Node Parameters

If you do not specify the required parameters, and the prompt:YES parameter is specified in the cfnode command line, you are prompted for all information required to successfully execute the cfnode utility.

See the following table for the required node parameters.

Note: If the required parameters are not supplied and the prompt:NO parameter is specified in the cfnode command line, the cfnode command fails.

Required Parameter (Shortcut)	Description

node (n)	The node parameter is used to specify the name of the node to be added or updated to the cfnode.cfg file. The node name can be up to 256 characters long and cannot contain any spaces.
	Examples:
	`node:dataserver`
	`n:dataserver`

systemType (s)	The systemType parameter is used to specify the type of system represented by this node definition. The valid system types are as follows: ◆ HPUX ◆ SUN/SOLARIS ◆ AIX ◆ LINUX ◆ Windows ◆ IBMi

	◆ z/OS
	◆ Command_Center
	◆ Other
	
	Examples:
	`systemType:Windows`
	`s:Windows`

hostname (h)	The hostName parameter is used to specify the IP address of the node. This value can be the dotted IP address of the remote machine or a resolvable host name.
	
	Examples:
	`hostname:11.22.33.44`
	`h:computer.domain.com`

port (p)	The port parameter is used to specify the port number on which the remote node is listening.
	
	Examples:
	`port:46464`
	`p:46464`

See the following table for the optional node parameters.

Note: The cfnode command does not require the optional parameters to be defined if the prompt:NO parameter is supplied.

Optional Parameter (Shortcut)	Description
Action (a)	The action parameter is used to specify the action to be taken. The valid values are: Delete, List, and Add. Example: `action:delete n:192.168.20.53`

netmask (net)	This is the netMask for the remote IPv4 address.
	
	Examples:
	`netMask:255.255.255.0`
	`net:255.255.255.128`

netMask6 (net6)	This is the netMask for the remote IPv6 address. This is a number between 8 and 128 and a multiple of 8.
ssl	The ssl parameter is used to specify whether SSL is used for TCP/IP communications. Example: `ssl:Y`
compress (c)	The compress parameter is used to specify the default compression type for all transfers with this node. The valid values are as follows: ◆ LZ ◆ RLE ◆ ZLIB1 - ZLIB9 ◆ NO – No default compression

	◆ NEVER – Never use compression
	
	Note: NEVER is the only option that cannot be overridden by options on the command line.
	
	Examples:
	`compress:LZ`
	`c:NEVER`

encrypt (e)	The encrypt parameter is used to specify the default encryption type to use for all transfers with this node. The valid values are as follows:
	
	◆ DES
	◆ 3DES - Triple DES
	◆ BF - Blowfish Encryption
	◆ BFL - Blowfish Long
	◆ RIJN(AES) - Rijndael
	◆ NO - No encryption
	◆ NEVER - Never use encryption
	
	Note: NEVER is the only option that cannot be overridden by options on the command line.
	
	Examples:
	`encrypt:DES`
	`e:NEVER`

security (sl)	This parameter determines whether the node is HIPAA or FIPS-140 compliant. If set, only HIPAA or FIPS-140 compliant encryption types are listed. See General Properties Page, System Configurations: Security Policy for more information. Example: `security:HIPAA`
lct	The name of the Local Conversion Table (also referred to as the Local Translation File), which is used to translate the data on the local side. Example: `lct:convert.txt`
rct	The name of the Remote Conversion Table (also referred to as the Remote Translation File), which is used to translate the data on the remote side. Example: `rct:convert.txt`
responder (r)	This parameter defines whether a responder profile is used for this node. The valid values are Yes, No, and Dual. The value D (Dual) means that the substitution of a real user ID occurs only if the responder profile exists and a match is found. If no match is found, the platform server attempts to log in with the remote user ID and password, rather than generate an error message that responder profile does not exist or the information does not match. On the other hand, a value of Yes means that the platform server does not try to log in with the remote user ID and password, and a value of No means that the platform server does not check the responder

	profiles.
	
	Example:
	`responder:Yes`
	`r:Y`

description (d)	The description parameter is used to specify a text description of the node definition. The description can be up to 256 characters and can contain spaces. If the description contains spaces, it must be enclosed in double quotation marks. The cfnode command does not require the description parameter defined if the prompt:NO parameter is supplied.
	
	Examples:
	`description: “This is a sample description”`
	`d: “This is a sample description”`

commandsupport (ccc)	It defines the actions that Command Center can perform on this node. The valid values are as follows:
	
	◆ ALL - NODE, PROFILE, AUDIT, ALTER, PING, and TRANSFER are supported on this node.
	◆ NONE - No Command Center function is supported on this node. This is the default setting if the parameter is not defined.
	◆ AUDIT - This node supports requests that inquire on the MFT Platform Server audit file.
	◆ NODE - Node List and Update functions are supported on this node.
	◆ PING - MFT Platform Server fusping requests are supported on this node.
	◆ PROFILE – Profile list and update functions are supported on this node.
	◆ TRANSFER - This node supports the Command Center Transfer function that initiates file transfers.
	
	Examples:
	`ccc:PING`
	`commandsupport:TRANSFER`

winners (win)	The Winners parameter is used to specify the maximum number of initiator transfers for the node. The valid values are from 1 to 50. If the Winners parameter is not defined in the node definition, the default value 5 is used. **Note**: This parameter only works for transfers submitted to the the platform server. It does not work for command line transfers. Example: `Winners:2`
prompt	The prompt parameter is used to put cfnode into an interactive mode. If prompt:YES is supplied, cfnode prompts you for all information required to create a node. You will also be prompted whether you want to create cfnode.cfg if it cannot be found. Prompt is turned on by default. If you do not want to be prompted, use prompt:NO.

-?	The -? parameter is used to display the online help for cfnode.

See the following output of online help:

```
usage: cfnode [required-parameters] [optional-parameters]
[required-parameters]:
   n: or node:            - Name of Node
   s: or systemType:      - Type of system (ie. Windows, UNIX, SUN, etc.)
   h: or hostName:        - Network address of remote node. This may be
                                host name or a dotted IP address.
                                (This parameter is only required for
                                TCP/IP transfers.)
   p: or port:            - Port number that remote node is listening on.
                                (This parameter is only required for
                                TCP/IP transfers.)

[optional-parameters]:
   a: or action:          - Following values are allowed:
                                : Delete (Nodename is required)
                                : List (Nodename is optional)
                                : Add (Default value)
   net: or netMask:       - NetMask for remote IpAddress. Valid value:
                                netmask.
   net6: or netMask6:     - NetMask for remote IpV6 Address. Valid value:
                             A number between 8 and 128 and a multiple of 8.
   ssl:                   - Either Yes or No depending on whether remote
                                node requires an ssl connection.
   c: or compress:        - Type of default compression to use during
                                transfers. Valid compression types:
                                (LZ | RLE | NO | NEVER | ZLIB1 - ZLIB9)
   e: or encrypt:         - Type of default encryption to use during
                                transfers. Valid encryption types:
                                (DES | 3DES | BF | BFL | RIJN(AES) | NO | NEVER)
   sl: or security:         - Security Compliance level to use during
                                transfers. Valid security types:
                                (Default | None | HIPAA)
   lct:                   - Local translation file. Valid value:
                                file path.
   rct:                   - Remote translation file. Valid value:
                                file path.
   r: or responder:       - Either Yes, No or Dual depending on whether or not
                                to use ResponderProfiles with this node. If
                                responder profiles are to be allowed as well as
                                regular logins, enter Dual.
```

```
   d: or description:     - Text description of the following node
                                 definition. Note: the definition must be
                                 encapsulated in " ".
   ccc: or commandsupport: - The actions this node will allow MFT Command Center to
perform:
                                 (ALL, NONE, NODE, PROFILE, AUDIT, PING, TRANSFER)
   win: or winners:        - Max Initiator Transfers per node. Default value is 5
   prompt:                 - Prompts the user for corrections when errors
                                 are found.
                                 Valid values: (YES | NO). Default is YES.
   -?                      - Online help.
```

4.10.1.2 Examples of Using cfnode Utility

The following sample shows how cfnode is used with the command line options.

At the command prompt the following operations are performed:

```
C:\>cd Program Files\TIBCO\MFT Platform Server\System
C:\Program Files\TIBCO\MFT Platform Server\System>cfnode n:dataServerA s:Windows
h:111.222.33.55 p:46464 c:RLE e:NO rct:rmttrans.txt d:”This is a sample node definition”
prompt:NO
```

The following example shows a sample of cfnode by using the prompt parameter:

```
C:\Program Files\TIBCO\MFT Platform Server\System>cfnode prompt:YES
Enter a valid node name: dataServerB
Enter a System Type for Node[dataServerB]:
1: HPUX
2: SUNOS/SOLARIS
3: AIX
4: LINUX
5: Windows
6: IBMi
7: z/OS
8: Command_Center
9: Other
: 5
Enter a valid IP address for Node [dataServerB]: 111.222.33.44
Would you like to specify netmask for remote IpAddress:
1: Yes
2: No
: 2
Enter the port for which Node [dataServerB] is configured to
use:46464
Enter the Security Compliance level for file transfers:
1: Default ( use Security Policy from Server Property )
2: None
3: HIPAA
: 1
Should SSL be used:
1: Yes
2: No
: 2
What should be the default encryption used:
1: DES
2: 3DES
3: BF
4: BFL
5: RIJN(AES)
6: No default encryption
7: Never use encryption
: 6
```

```
What should be the default compression used:
1: LZ
2: RLE
3: ZLIB
4: No default compression
5: Never use compression
: 2
Would you like to specify local translation file:
1: Yes
2: No
3: None < Caution! If uncertain, refer to User Guide. >
: 2
Would you like to specify remote translation file:
1: Yes
2: No
: 1
Please enter remote translation file:
: remotetrans.txt
Use Responder Profiles for this node?
1: Yes
2: No
3: Dual
4: Do not define
: 3
Would you like to add a description:
1: Yes
2: No
: 1
Please enter a description:
: Sample TCP node
Enter the Command Center parameters this node will support:
1: All
2: None
3: Audit
4: Node
5: Ping
6: Profile
7: Transfer
: 5
Enter the Command Center parameters this node will support:
1: All
2: None
3: Audit
4: Node
6: Profile
7: Transfer
99: No more parameters
: 99
Enter Winners number for this node: 2

A Node definition was created for:
[dataServerB]
   SystemType                = Windows
   Protocol                  = tcpip
   HostName                  = 111.222.33.44
   Server                    = 46464
   SSL                       = N
   Compression               = RLE
   Encryption                = NO
   SecurityPolicy            = Default
   RemoteCTFile              = remotetrans.txt
   ResponderProfile          = D
   Description               = Sample TCP node
   CommandCenterSupport      = PING
   Winners                   = 2
```

By using the cfnode command, the preceding samples update a cfnode.cfg file with the following contents:

```
[dataServerA]
   SystemType         = Windows
   Protocol           = tcpip
```

```
   HostName           = 111.222.33.55
   Server             = 46464
   Compression        = RLE
   Encryption         = NO
   SecurityPolicy     = None
   RemoteCTFile       = rmttrans.txt
   Description        = This is a sample node definition
   CommandSupport     = NONE
   Winners            = 5

[dataServerB]
   SystemType         = Windows
   Protocol           = tcpip
   HostName           = 111.222.33.44
   Server             = 46464
   SSL                = N
   Compression        = RLE
   Encryption         = NO
   SecurityPolicy     = Default
   RemoteCTFile       = remotetrans.txt
   ResponderProfile   = D
   Description        = Sample TCP node
   CommandSupport     = PING
   Winners            = 2
```

Node definitions can be deleted or listed by using the “action” parameter “delete” or “list”. The following sample shows how to list nodes:

```
C:\Program Files\TIBCO\MFT Platform Server\System>cfnode a:list
[dataServerA]
   SystemType         = Windows
   Protocol           = tcpip
   HostName           = 111.222.33.55
   Server             = 46464
   Compression        = RLE
   Encryption         = NO
   SecurityPolicy     = None
   RemoteCTFile       = rmttrans.txt
   Description        = This is a sample node definition
   CommandSupport     = NONE
   Winners            = 5

[dataServerB]
   SystemType         = Windows
   Protocol           = tcpip
   HostName           = 111.222.33.44
   Server             = 46464
   SSL                = N
   Compression        = RLE
   Encryption         = NO
   SecurityPolicy     = Default
   RemoteCTFile       = remotetrans.txt
   ResponderProfile   = D
   Description        = Sample TCP node
   CommandSupport     = PING
   Winners            = 2
```

4.10.2 Local User Profiles

Local user profiles define a remote user ID and password that can be used by a local user to log on to a remote node. When a node is supplied in a transfer, a user profile is chosen for the node based on the current logon user and the information in that user profile is used to log on to the remote system. A local user profile contains the following information:

- Node with which the Local User Profile is associated.
- Local user name who can use this profile.
- Remote user name to use to log on to the node.

- Remote password to use to log on to the node (in encrypted format).

Profile definitions are stored in a file named cfprofile.cfg located in the MFT Platform Server directory. Local User Profiles can be added and updated by using the MFT Platform Server cfprofile command. Before cfprofile updates any information in cfprofile.cfg, a backup of this file is created called cfprofile.bak.

4.10.2.1 Local User Profile Parameters

You can specify the following parameters in the cfprofile command line. If you do not specify these parameters, and the prompt:YES parameter is specified, you are prompted for all information required to successfully execute the cfprofile command.

See the following table for the required local user profile parameters.

Note: If the required parameters are not supplied and the prompt:NO parameter is specified in the cfprofile command line, the cfprofile command fails.

Required Parameter (Shortcut)	Description

node (n)	The node parameter is used to specify the name of the node with which the user profile is associated.
	
	The node name can be up to 256 characters long and cannot contain any spaces. A node must already exist in cfnode.cfg to successfully add or update a user profile.
	
	Example:
	`node:dataserverA`
	`n:dataserverB`

password (p)	The password parameter is used to specify a password to be used to log on to the remote node. This parameter is case sensitive.
	
	Example:
	`password:apple`
	`p:computer`

user (u)	The user parameter is used to specify the user name to be used to log on to the remote system.
	
	The user ID can be up to 31 characters in length which includes fifteen characters for a machine name, a slash, and up to fifteen characters for the user ID. The user ID is generally not case sensitive, except in a UNIX system. If the remote node is a Windows system, the domain must also be specified by using either of these formats: `domain\\username` or `domain/username`.
	
	Example:
	`user:kenny`
	`u:bob`

See the following table for the required local user profile parameters.

Note: The cfprofile command does not require the optional parameters to be defined if the prompt:NO parameter is supplied.

Optional Parameter (Shortcut)	Description

action (a)	The action parameter is used to specify the action to be taken. The valid values are Delete, List, and Add. The cfprofile command does not require the action parameter to be defined if the prompt:NO parameter is supplied.
	
	Example:
	`action:delete`
	`a:delete`

localUser (l)	The localUser parameter is used to define the identity of a different local user on the local system. With this, userA can add a User Profile for userB without having to be logged in as userB.
	
	Only an administrator or a member of the cfadmin group can use this option. When this parameter is not defined, the **prompt** parameter is set to YES, and you log on as the root account or a member of the cfadmin group, you are prompted whether you want to define another local user.
	
	The administrator or a member of the cfadmin group can use the localUser option to create a user profile that can be used by all local users who want to command transfers with a particular node. In this case, the localUser option must be coupled with the *ALL option. If no user profile is available for the current user on a given node but an *ALL entry is defined, the platform server uses the *ALL user profile for transfers.
	
	Example:
	`localUser:john`
	`l:*ALL`

prompt	The prompt parameter is used to put cfprofile into an interactive mode. If prompt:YES is supplied, cfprofile prompts you for all information required to create or update a user profile. Using the prompt:YES parameter, you will be asked whether to create the cfprofile.cfg file if it cannot be found. Prompt is turned on by default. If you do not want to be prompted, supply prompt:NO.
-?	The -? parameter is used to display the online help for cfprofile.

See the following output of online help:

```
usage cfprofile [required-parameters] [optional-parameters]
[required-parameters]:
   n: or node:            - Name of Node
   u: or user:            - User ID to be used on the remote node. If the remote node
                            is a Windows machine the domain must also be specified
                            using the following format: domain\\userID or
                            domain/userID
   p: or password:        - Password to be used on the remote node.
[optional-parameters]:
   a: or action:          - Following values are allowed:
                                  : Delete  Nodename is required
                                            localUser is Admin option
                                  : List    Nodename is optional
                                            localUser is Admin option
```

```
                                  : Add (Default value)
   l: or localUser:       - Specifies the use of a local user name other than the one
                            currently logged in. Only Administrator or member of the
                            cfadmin group may use this parameter.
   prompt:                - Prompts the user for corrections when errors are found.
                            Valid values: (YES | NO). Default is YES.
   -?                     - Online help.
```

4.10.2.2 Examples of Using cfprofile Utility

The following sample shows how cfprofile can be used on a command line with short commands:

```
C:\>cd Program Files\TIBCO\MFT Platform Server\System
C:\Program Files\TIBCO\MFT Platform Server\System>cfprofile n:dataserverA u:kenny
p:apple

Profile added.
```

The following example shows a sample cfprofile by using the prompt parameter:

```
C:\Program Files\TIBCO\MFT Platform Server\System>cfprofile prompt:YES

Enter a valid Node Name: dataserverB

Add profile as local user Admin?
1: Yes
2: No
: 2
Enter new local user: *ALL

Enter a valid Remote User: bob
Enter a valid Remote Password:
Re-enter Remote Password:

Profile added for..
Local User            = *ALL
Remote User           = bob
Remote Password       = ******
```

The previous sample cfprofile commands update a cfprofile.cfg file with the following contents:

```
[dataserverA]
   Admin = Secure       kenny
8eb26af8131f0634820482c79c83ff1b68584c8aa2f549eb10e984155eef


[dataserverB]
   *ALL    = Secure            bob
84e053ab10463b6ea6c105e2c9bdbaadebc11b1ab9ba58774343702fbff
```

The local user profiles can be listed or deleted by using the action parameter. The following sample shows how to list profiles:

```
cfprofile a:list

[dataserverA]
 Local User        = root
 Remote User       = kenny

[dataserverB]
 Local User        = *ALL
 Remote User       = bob
```

4.10.3 Responder Profiles

Responder Profiles define a local user name and password that are used in place of the incoming user name and password. By using responder profiles, a remote MFT Platform Server does not have to know an actual user name and password on your local machine to initiate a transfer. A responder profile contains the following information:

- Node: the remote system with which the responder profile is associated.
- Remote User Name: the user ID supplied by the remote system initiating the transfer. (Does not have to be a valid user name on the local system.)
- Remote Password: the password supplied by the remote system initiating the transfer. If the remote user is a verified user, this parameter must be set to *VER.
- Local User Name: the user ID used by MFT Platform Server when processing a transfer on your local machine from the specified remote user.
- Local Password: the local password associated with the local user ID.

Responder profile definitions are stored in the cfrprofile.cfg file located in the MFT Platform Server System directory. In the cfrprofile.cfg file, all password information is encrypted. Responder profiles can be added or updated by using the MFT Platform Server cfrprofile command. Before cfrprofile updates any information in cfrprofile.cfg, a backup of this file is created called cfrprofile.bak.

4.10.3.1 Responder Profile Parameters

You can specify the following parameters in the cfrprofile command line. If you do not specify these parameters, and the prompt:YES parameter is specified, you are prompted for all information required to successfully execute cfrprofile command.

See the following table for the required responder profile parameters.

Note: If the required parameters are not supplied and the prompt:NO parameter is specified in the cfrprofile command line, the cfrprofile command fails.

Required Parameter (Shortcut)	Description

node (n)	The node parameter is used to specify the name of the node with which the responder profile is associated.
	
	The node name can be up to 256 characters long and cannot contain any spaces. A node must already exist in cfnode.cfg to successfully add or update a responder profile.
	
	Example:
	`node:dataserverA`
	`n:dataserverB`

lPass (lp)	The lPass parameter is used to specify the local password associated with the local user ID. This must be a valid user name on the local system.
	
	Example:
	`password:apple`
	`p:computer`

lUser (l)	The lUser parameter is used to specify the local user name to be mapped to the incoming remote user name. This must be a valid user name on the local system.

	Example:
	`lUser:john`
	`l:john`

rPass (rp)	The rPass parameter is used to specify the remote password that is sent by the remote MFT Platform Server initiating the transfer.
	Note: If this responder profile is to be in conjunction with an already verified user, rPass must be set to *VER.
	Example:
	`rPass:apple`
	`rp:*VER`

rUser (r)	The rUser parameter is used to specify the remote user name that is sent by the remote MFT Platform Server installation initiating the transfer.
	Example:
	`rUser:kenny`
	`r:kenny`

See the following table for the required responder profile parameters.

Note: The cfrprofile command does not require the optional parameters to be defined if the prompt:NO parameter is supplied.

Optional Parameter (Shortcut)	Description

action (a)	The action parameter is used to specify the action to be taken. The valid values are Delete, List, and Add.
	Example:
	`action:delete`
	`a:delete`

prompt	The prompt parameter is used to put cfrprofile into an interactive mode. If prompt:YES is supplied, cfrprofile prompts you for all information required to create or update a responder profile. Using the prompt:YES parameter, you will be asked whether to create the cfrprofile.cfg file if it cannot be found. Prompt is turned on by default. If you do not want to be prompted, supply prompt:NO.
-?	The -? parameter is used to display the online help for cfrprofile.

See the following output of online help:

```
usage cfrprofile [required-parameters] [optional-parameters]
[required-parameters]:
   n:  or node:           - Name of Node
   r:  or rUser:          - Remote User ID
   rp: or rPass:          - Remote User's password. If remote user is intended
                            to be a verified user enter '*VER' as the remote
                            password.
   l:  or lUser:          - Local User ID to be used. If the local
                            system is a Windows machine
                            the domain must also be specified using
                            the following format: domain\\userID or
                            domain/userID
   lp: or lPass:          - Local password to be used.
[optional-parameters]:
   a: or action:          - Following values are allowed:
                                  : Delete  Nodename is required
```

```
                                        localUser is Admin option
                                : List  Nodename is optional
                                        localUser is Admin option
                                : Add (Default value)
   prompt:                - Prompts the user for corrections when errors are
                            found.
                            Valid values: (YES | NO). Default is YES.
   -?                     - Online help.
```

4.10.3.2 Examples of Using cfrprofile Utility

The following sample shows how cfrprofile can be used on a command line with short commands:

```
C:\>cd Program Files\TIBCO\MFT Platform Server\System
C:\Program Files\TIBCO\MFT Platform Server\System>cfrprofile n:dataServerA r:kenny
rp:apple l:john lp:orange prompt:NO

Responder Profile added for...
Remote User           = kenny
Remote Password       = ****************
Local User            = john
Local Password        = ****************
```

The following example shows a sample of cfrprofile by using the prompt parameter:

```
C:\Program Files\TIBCO\MFT Platform Server\System>cfrprofile prompt:YES

Enter a valid Node Name:   dataServerA
Enter a valid Remote User: kenny
Enter a valid Remote Password:
Re-enter Remote Password:
Enter a valid Local User:  john
Enter a valid Local Password:
Re-enter Local Password:

Responder Profile updated for...
Remote User                = kenny
Remote Password                   = ****************
Local User                 = john
Local Password                    = ****************
```

The above cfrprofile commands update a cfrprofile.cfg file with the following contents:

```
[dataServerA]
      RemoteUser=kenny
            RemotePassword= 24c89e105efee2f3d2d84988a4140652b45d7345
       LocalUser=john
          LocalPassword= 40562eb4d4fd437ab7d7b256221267b6c43da8fb8
```

The responder profiles can be listed or deleted by using the action parameter. The following sample shows how to list responder profiles:

```
cfrprofile a:list

[dataserverA]
 Local User        = john
 Remote User       = kenny
```

4.10.4 Distribution Lists

Distribution lists define multiple nodes and a default directory to which you can perform send transfers. You can configure the distribution list name, the nodes to be used, and the distribution directory in the cflist.cfg file located in the MFT Platform Server installation directory. Because there is no maintenance program for distribution lists, you must use a text editor to update the cflist.cfg file.

When a distribution list is selected from the Transfer window by using the List button, the destination information is pulled from your node configurations.

Note: Distribution lists can only be used for a send transfer. When you perform a receive request, the List button is grayed out.

You can find the following examples in the cflist.cfg file:

```
[AccList]
# Distribution list : AcctList
Node=NYAcct,LAACCT,chiacct

[Stores]
# Distribution list : Stores
Node= Store1, Store2,
Directory = /tmp/prod/data
Node=Store5
```

4.10.4.1 Distribution Parameters

See the following table for the supported parameters for distribution lists.

Parameter	Description
distribution_list_name	This is a required parameter. The distribution list name can be from 1 to 32 characters and cannot contain any spaces. Specify the distribution list name between square brackets. Any names longer than 32 characters are truncated.
Node	This is a required parameter. The Node parameter is used to specify either a single or multiple nodes to conduct transfer requests with when this distribution list is used. Multiple nodes defined on one line must be delimited by a comma. Up to 4096 characters can be entered on a single line.
Directory	The Directory parameter is used to define the default destination directory for the nodes. If the parameter is not defined, the directory defined in the transfer window or on the command line is used. However, if a directory is defined in the distribution list, it overrides a directory that is defined in the transfer window or on the command line.

4.11 TIBCO Accelerator

You can use TIBCO Accelerator technology to improve data transfer speed over IP network connections (high bandwidth, high latency).

TIBCO Accelerator technology is added to TIBCO MFT Platform Server to provide a faster way to send files to remote destinations, where high latency is a problem with long distance connections.

TIBCO Accelerator technology uses its own version of User Datagram Protocol (UDP), and it offers a parallel implementation of Transmission Control Protocol (TCP), called Parallel Delivery Protocol (PDP).

4.11.1 TIBCO Accelerator Ports

By default, the TIBCO Accelerator listens on port 9000 for incoming TCP or UDP requests and listens on port 9002 for incoming PDP requests. The platform server uses port 9099 to connect to TIBCO Accelerator.

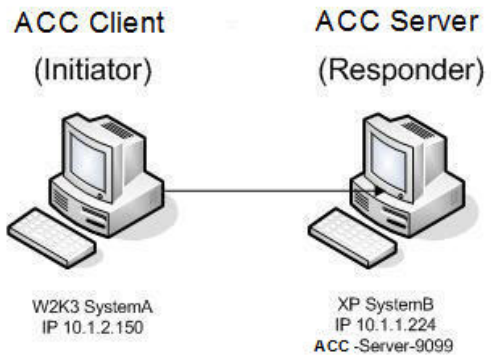
When a request is received, TIBCO Accelerator Client sets a port number for the data transmission between TIBCO Accelerator Server and Responder in the range of 9100 - 9199.

Note: Ports 9000, 9002, and 9100 - 9199 must be opened in the firewall to allow TIBCO Accelerator Client to access TIBCO Accelerator Server. If requests are initiated from an external computer, these ports must be opened on the firewall for incoming traffic. If requests are initiated from an internal computer, these ports must be opened on the firewall for outgoing traffic.

4.11.2 Using TIBCO Accelerator within MFT Platform Server

TIBCO Accelerator technology is available in TIBCO MFT Platform Server for Windows. Your Windows MFT Platform Server can act both as a TIBCO Accelerator Client and/or a TIBCO Accelerator Server. You can send and receive files from z/OS and UNIX platforms (System i can only act as a responder), but only when they pass through the Windows MFT Platform Servers running the TIBCO Accelerator service (RsTunnel.exe). Two example diagrams and configuration instructions are presented as follows.

4.11.2.1 Example 1: Windows to Windows Using TIBCO Accelerator for Windows



This example describes a file sent from a Windows MFT Platform Server (SystemA) to a Windows MFT Platform Server (SystemB) using TIBCO Accelerator. This is the simplest TIBCO Accelerator transfer to configure.

First, verify SystemB has the TIBCO Accelerator service running and is listening on the default port 9099. To verify this, open the MFT Platform Server Administrator window on SystemB, and then display the server properties, click the Accelerator tab as seen in Accelerator of this manual.

Next, on SystemA, you can set an advanced TCP transfer by filling in the necessary transfer detail information in the various Transfer Property tabs as seen in Transfer Properties of this manual, except for when you get to the Accelerator tab as seen below:

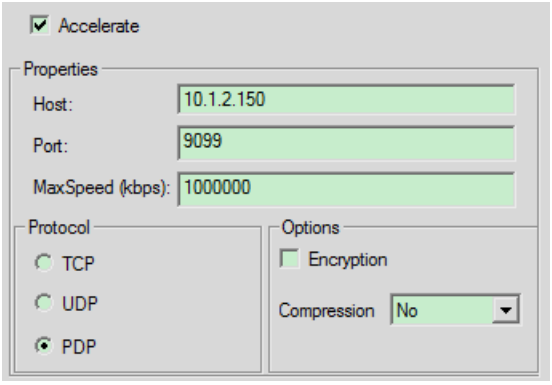
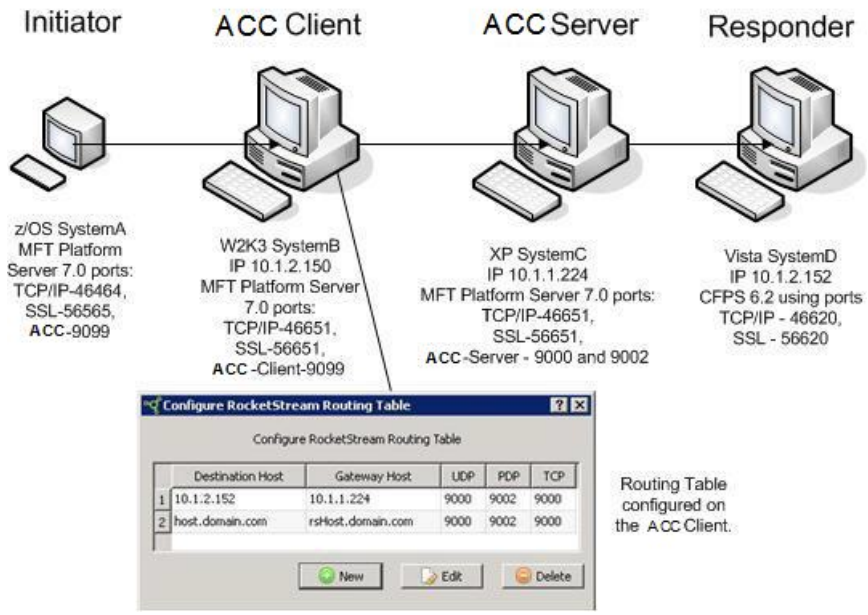


Figure 4.11.a

By default, the TIBCO Accelerator parameters are grayed out. To have this transfer be sent via TIBCO Accelerator, you must select the Accelerate check box. Only then can you configure the TIBCO Accelerator parameters. In the screenshot above, the transfer is defined to go through the local TIBCO Accelerator Client (SystemA). You can read more about the TIBCO Accelerator parameters in Accelerator Tab of this manual. In this example, the default values are used for all other fields on this screen.

After your transfer details are completed, click the OK button on the bottom of your Transfer Properties window. Your file is now sent by using TIBCO Accelerator.

4.11.2.2 Example 2: z/OS to UNIX Using TIBCO Accelerator for Windows



As you can see in Example 2, more operations are performed than in Example 1. This diagram demonstrates sending a file from a z/OS MFT Platform Server (SystemA) to a Linux MFT Platform Server system (SystemD). Both of these servers do not have the TIBCO Accelerator technology contained in them and therefore must pass the transfer to TIBCO MFT Platform Server for Windows server running the TIBCO Accelerator service.

When conducting TIBCO Accelerator transfers of this kind, you must configure a TIBCO Accelerator Routing Table on the TIBCO Accelerator Client that the Platform Server Initiator connects to. The TIBCO Accelerator Client needs the connectivity information for the destination TIBCO Accelerator Server that connects to the Platform Server Responder.

Note: If your final destination is the TIBCO Accelerator Server itself, no routing table entry is needed. It is only required when the Platform Server Responder is a different machine than the TIBCO Accelerator Server.

Example 1 shows the Platform Server Responder on the same machine as the TIBCO Accelerator Server; therefore, no routing table updates are needed. Example 2 shows the Platform Server Responder on a different machine as the TIBCO Accelerator Server; therefore, the routing table must be updated.

To configure the routing table, open Windows Explorer and navigate to the following folder:

<PlatformServer_Install>\TIBCO\MFT Platform Server\RSTunnel\

Double-click the file **RSTunnelConfig.exe**. The following window is displayed:

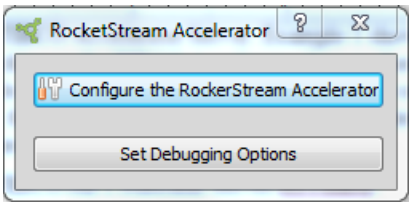


Figure 4.11.b

Click **Configure the RocketStream Accelerator** (Set Debugging Options can only be used when instructed by TIBCO Technical Support.)

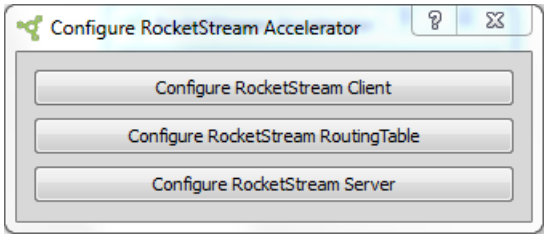


Figure 4.11.c

Click the **Configure TIBCO Accelerator Routing Table** button. You can see an example setup.

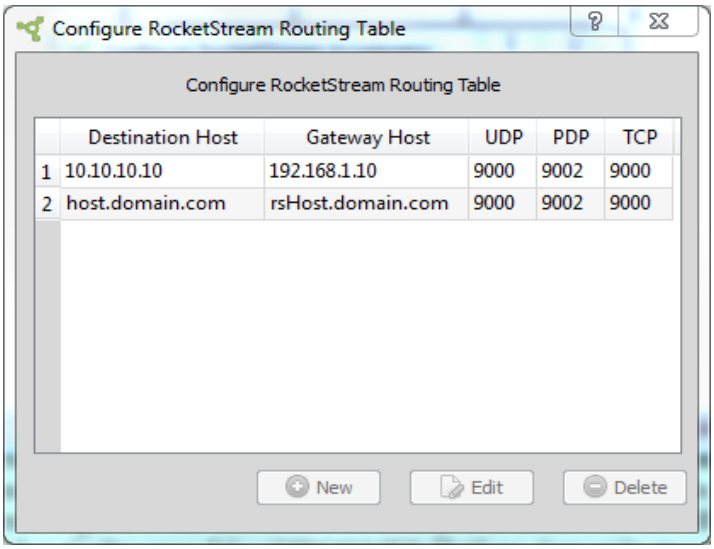


Figure 4.11.d

Click the first line to highlight it and then click the Edit button. The following window is displayed:

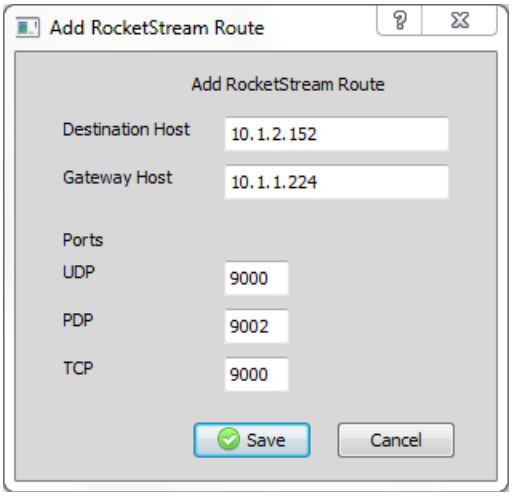


Figure 4.11.e

In this window, you define the Destination Host (the IP of the server that is the final destination for your file being transferred: the Responder) and the Gateway Host (the remote TIBCO Accelerator Server that is initially receiving your file transfer before passing it off to the Destination Host). You can also edit the default ports used for the various protocols TIBCO Accelerator provides. As shown in the screenshot above, the Routing Table is configured with the IP addresses of SystemC and SystemD. When you are done, click the Save button.

You are then presented with a warning that the RocketStream Tunnel service must be restarted for the changes to take effect. Stop and start TIBCO Accelerator from your Server Properties window or you can open your Services window and restart MFT Platform Server. You can close the **Configure RocketStream Routing Table** window.

The routing table is configured on the TIBCO Accelerator Client defining what ports is used when sending files with the various protocols TIBCO Accelerator offers. At this time, you can also define what port and IP address your Client binds to if multiple network cards are available. By default, the client listens on port 9099. If you must change the default port number, click the

Configure RocketStream Client button as shown in Figure 4.11.c. The following window is displayed:

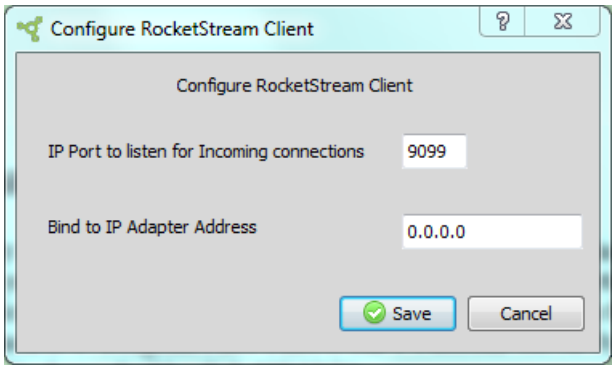


Figure 4.11.f

If your server has multiple network cards, you can define the IP adapter address you want the TIBCO Accelerator Client to bind to; otherwise, you can leave the **Bind to IP Adapter Address** field alone. When you are done, click the Save button and close the window.

You must define all the above configurations for a TIBCO Accelerator Client. However, if your TIBCO Accelerator Client would ever be switching roles and acting as a TIBCO Accelerator Server in the future, you can also configure the TIBCO Accelerator Server ports and IP address to bind to at this time by clicking the **Configure RocketStream Server** button as shown in Figure 4.11.c. The following window is displayed:

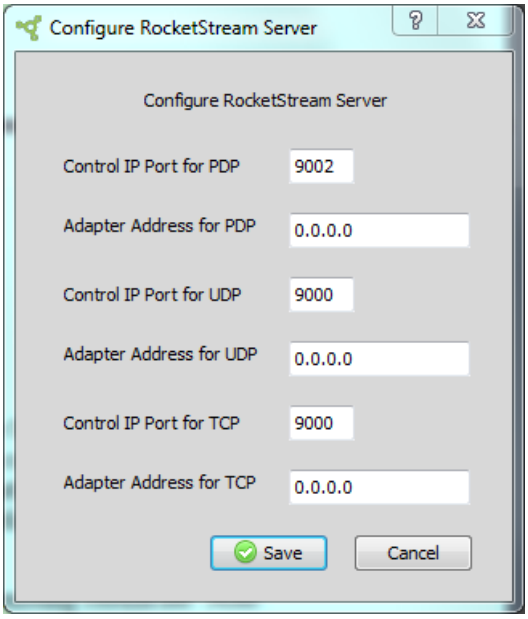


Figure 4.11.g

Unless you want to change the default ports being used by the server or you must bind to a specific IP address because multiple network cards are installed on the system, you can leave these settings alone.

This completes configuring TIBCO Accelerator Client and TIBCO Accelerator Server. For more information on how to initiate file transfers on a platform server for z/OS or UNIX, see TIBCO Managed File Transfer Platform Server for z/OS, and TIBCO Managed File Transfer Platform Server for UNIX documentation.

A TIBCO Accelerator Server can send a file to any MFT Platform Server responder with version 7.0 or lower. This includes TIBCO MFT Platform Server for Windows, UNIX, z/OS, and AS/400 (System i) servers.

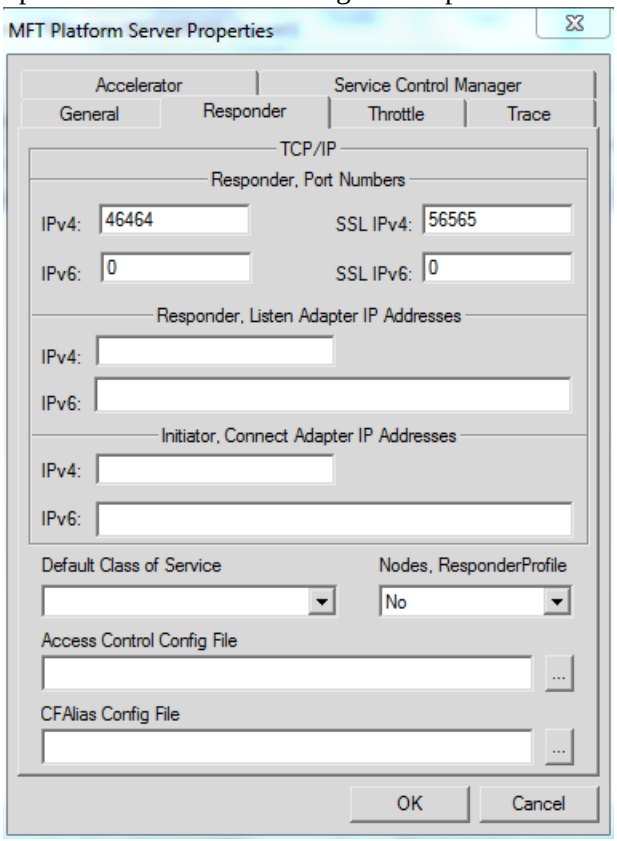
This concludes configuring the necessary steps needed for our Example 2 diagram. If you need further assistance for this example, contact TIBCO Technical Support.

4.12 SSL

An additional layer of security can be configured for MFT Platform Server transfers by enabling transfers over SSL. To properly configure SSL, each platform server must have a public and private key. To facilitate the certificate procurement, MFT Platform Server includes an SSL utility, SSLutility.exe (See section SSL Utility), which generates a private key and a Certificate Signing Request (CSR) file. A public key is then obtained by forwarding the CSR file to a Certificate Authority (CA) for authorization. When authorized, the CA returns a public certificate that has been signed by the CA and can be used by MFT Platform Server. This section describes the installation, configuration, and usage of SSL on MFT Platform Server.

4.12.1 SSL Installation

All SSL transfers must be performed on a port specifically identified for this purpose only. It is not the same port as the TCP/IP port that MFT Platform Server listens on for incoming requests. The SSL port is optional. Entering 0 as the SSL port number disables SSL.
The SSL port number can be configured at the time of installation by following the Install Shield steps. However, if it is not entered at the time of installation, you can set the port by opening the Server Properties window and clicking the Responder tab as follows:



Choose a port within the range of 1025 - 65535 that you want to use for SSL transfers. It is good practice to use port 56565. Then click the **OK** button. To invoke this change or addition made to the registry, stop and start the MFT Platform Server service.

4.12.2 SSL Utility

If you already have an SSL Private Key and Certificate in base64 format for the machine you have MFT Platform Server installed on, you can use it for SSL transfers. If you do not have an SSL certificate, you can use the SSLUtility.exe utility to issue a certificate request to Certificate Authority. It is located in the MFT Platform Server System directory, which is C:\Program Files\TIBCO\MFT Platform Server\System by default. To execute this program on Windows, double-click SSLUtility.exe.

You can use the SSLUtility.exe utility to create certificate requests and private keys, and view an existing certificate. **Note:** The bit strength must meet the requirements of CA.

4.12.2.1 Creating Certificates

The following screenshot depicts the menu of choices available when SSLUtility.exe is executed.

```
SSL Utilities Menu
1. Generate a Certificate Request
2. View a Certificate
3. Exit

Please enter your choice:
```

Selecting choice 1 to generate a certificate request prompts you to enter the following required fields to create the distinguished name of the certificate:

Parameter	Description
Certificate Holder's Name	The person for whom the certificate is made.
Organization	Group or company with which the certificate holder is associated.
Organizational Unit	Department within the organization.
City	City of certificate holder.
State	State of certificate holder.
Country	Country of certificate holder.
Email address	Email address of the holder of the certificate.
Certificate Request File	Fully qualified file name for the new certificate request.
Private Key File	Fully qualified file name for the new private key.
Private Key Password	Password that is required to access the private key. The maximum value is 20.

The utility then creates a certificate request and private key and places them in the files that you specified. These files can be forwarded to a certificate authority to request a certificate.

Note: The names of the file and directory for the Certificate Request File and the Private Key File cannot contain any spaces; otherwise, the files cannot be created properly.

4.12.2.2 Viewing a Certificate

To view a certificate, select 2 from the SSL Utilities menu.

```
SSL Utilities Menu
1. Generate a Certificate Request
2. View a Certificate
3. Exit
Please enter your choice: 2

View Certificate Menu

Please enter the Certificate Filename:
c:\MFT Platform Server\sslcert
```

When prompted to enter the certificate file name, enter the fully qualified file name.

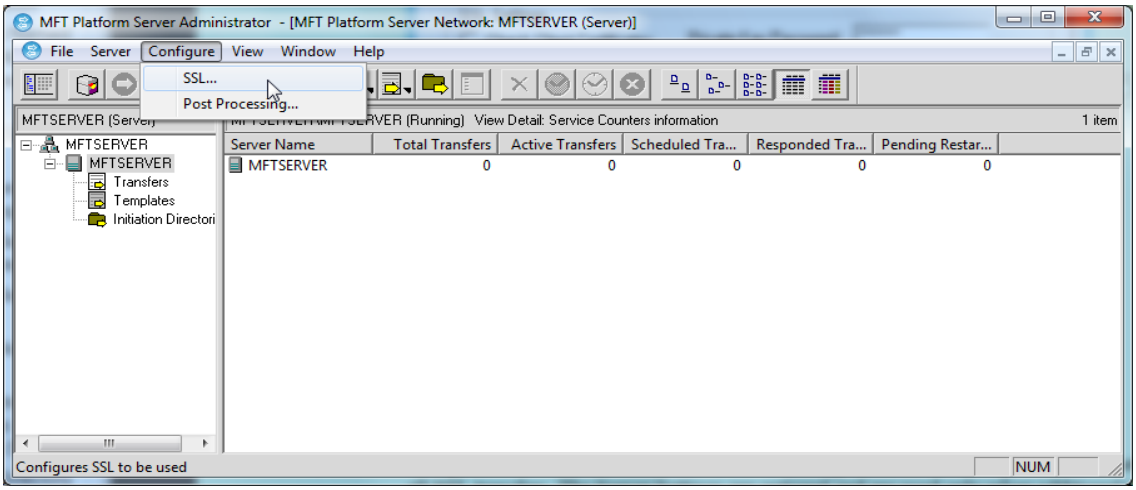
The following example shows a sample output:

```
Please enter the Certificate Filename:
c:\MFT Platform Server\sslcert
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 7 (0x7)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=TIBCO, OU=TIBCO Local CertAuth
        Validity
            Not Before: Aug 13 00:00:00 2005 GMT
            Not After : Aug 13 23:59:59 2006 GMT
        Subject: C=US, ST=NY, L=Garden City, O=TIBCO Software Inc.,
OU=Technical Support, CN=Joleen/Email=jbarker@tibco.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:ae:6a:25:45:19:e0:ec:d1:13:b7:a6:9c:fc:f4:
                    39:b6:a3:74:b2:98:4c:02:77:74:37:69:2f:08:f1:
                    3f:3e:95:68:1d:e8:93:09:90:8a:ec:16:8e:50:62:
                    82:57:31:8e:a5:6f:db:1c:72:79:c0:d3:de:83:e4:
                    f6:da:e1:ee:e0:d4:2f:26:05:77:f0:94:e9:70:20:
                    75:42:0d:64:eb:8f:36:a2:04:67:a9:e5:e0:ab:a3:
                    f9:a8:22:5d:75:b1:60:6e:82:ea:6f:5a:cf:61:d6:
                    2e:f7:36:b9:76:9e:4e:6d:f5
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            Netscape Comment:
                .<Generated by the SecureWay Security Server for z/OS (RACF)
            X509v3 Subject Key Identifier:
2C:C4:0E:E4:AC:E2:2D:9F:E3:EC:5F:32:67:53:B0:6A:D4:EB:36:F3
            X509v3 Authority Key Identifier:
keyid:42:77:A2:C7:AE:3D:A5:47:5C:30:FF:4F:51:B8:CF:ED:AC:D1:9C:3A
    Signature Algorithm: sha1WithRSAEncryption
        9f:7d:bd:66:f1:d5:2c:cf:5d:c5:cc:aa:16:16:e5:52:ae:04:
        89:51:66:c6:c5:03:0a:19:66:c1:d2:c9:30:4d:a4:85:c9:91:
        79:79:b0:61:bf:88:61:44:3e:21:fa:2d:98:85:b8:df:c5:77:
        ea:ee:c5:8b:7f:c3:27:56:69:3d:42:8b:c2:4a:89:2e:6f:85:
        fe:62:9c:fe:45:a0:3b:07:9b:1f:7b:f8:c0:35:89:af:be:72:
        8a:0c:a2:37:a5:fc:70:58:48:99:4f:40:ae:95:21:1e:4b:90:
        30:36
-----BEGIN CERTIFICATE-----
DXMxCzAJBgNVBAgTAm55QMowCAYDVQQHEwFnMQowCAYDVQQKEwFwMQowCAYDVQQLEwFxMQowCAYDVQQ
DEwFqMRAwDgYJKoZIhvcANQkBFgFqMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCuaiVFGeDs0R
O3ppz89Dm2o3SymEwCd3Q3aS8I8T8+lWgd6JMJkIrsFo5QYoJBGep5eCro/moIl11sWBugupvWs9h1i
73Nrl2nk5t9QIDAQABo4GQMIGNMEsGCWCGSAGG+
9yIE9TLzM5MCAoUkFDRikwHQYDVR0OBBYEFCzEDuSs4i2f4+xfMmdTsGrU6zbzMB8GA1UdIwQYMBaAF
EJ3oseuPaVHXDD/T1G4z+2s0Zw6MA0GCSqGSIb3DQEBBQUAA4GBAJ99vWbxXSzPXcXMqhYW5VKuBIlR
ZsbFAwoZZsHSyTBNpIXJkXl5sGH7iGFEPiH60piFuN/Fd+ruxoMojel/HCFSJlPQK6VIR5LkDA2
-----END CERTIFICATE-----

SSL Utilities Menu
1. Generate a Certificate Request
2. View a Certificate
3. Exit
Please enter your choice:
```

4.12.3 SSL Configuration

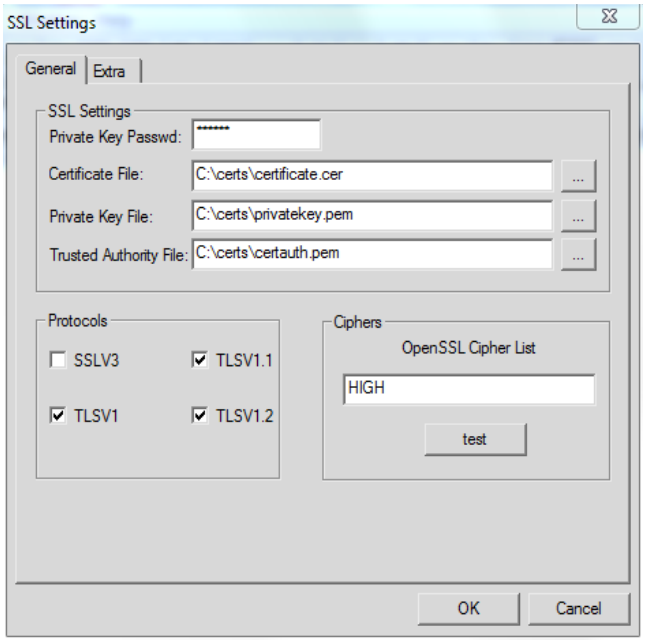
To configure TIBCO MFT Platform Server for SSL, open the SSL Settings dialog by selecting **Configure > SSL…** from the menu bar.



4.12.3.1 SSL Settings

SSL Settings have two tabs: General tab and Extra tab. The General tab settings are required for all SSL transfers. The Extra tab settings are optional and are used only when additional tracing or certificate authorization is required.

- **General Tab**



Private Key Password

The password or passphrase must be entered for MFT Platform Server to access the private key file for data encryption or decryption. Asterisks are displayed in the box as the password is entered to ensure the security of the private key file.

Certificate File

In the Certificate File text box, enter the drive, path, and file name of the base64 encoded certificate to be used by MFT Platform Server. This certificate is presented when MFT Platform Server is acting as the client. A browse button is provided to the right of the text box to facilitate this process.

Private Key File

In the Private Key File text box, enter the drive, path, and file name of the base64 encoded private key to be used when MFT Platform Server is decrypting received data. A browse button is provided to the right of the text box to facilitate this process.

Trusted Authority File

In the Trusted Authority File text box, enter the drive, path, and file name of the base64 encoded file containing the trusted authority certificates of CA, which recognizes all the certificates used in the platform server deployment that MFT Platform Server can accept from clients. A browse button is provided to the right of the text box to facilitate this process.

Protocols: SSLV3, TLSV1, TLSV1.1, TLSV1.2

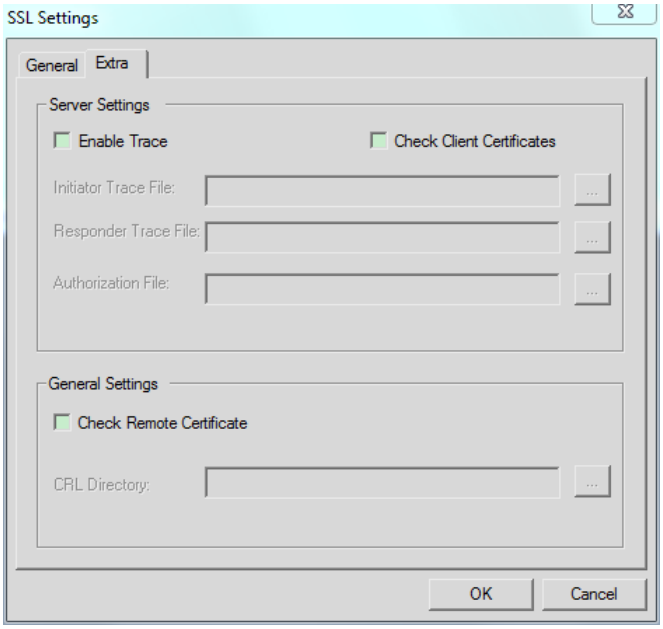
To define the protocols accepted for SSL transfers, select the check box to the left of the protocol.

Ciphers: OpenSSL Cipher List

In the OpenSSL Cipher List text box, enter the cipher suite name used in the Client and Server TLS negotiation. When not defined, the default OPENSSL TLS ciphers will be used. You can use the Test button to validate the cipher name.

Note: To perform SSL transfer successfully, you must use the same cipher suite for the server and client.

- **Extra Tab**



Enable Trace

Select this check box to enable tracing. When this check box is selected, the other fields in this section become available. Although SSL tracing is optional, when it is selected, the Initiator Trace File and Responder Trace File fields are required. Tracing should only be turned on at the request of TIBCO Technical Support.

Check Client Certificates

Select the Check Client Certificates check box if you want to perform client authentication in addition to server authentication. If this check box is not selected, only server authentication is performed. Selecting the Check Client Certificates check box also enables the Authorization File text box in the Server Settings section of this panel. An authorization file can be entered for additional security if Check Client Certificates is selected.

Initiator Trace File

In the Initiator Trace File text box, enter the drive, path, and file name of the file to be used for tracing information when acting as the initiator of the transfer. A browse button is provided to the right of the text box to facilitate this process.

Responder Trace File

In the Responder Trace File text box, enter the drive, path, and file name of the file to be used for tracing information when acting as the responder of the transfer. A browse button is provided to the right of the text box to facilitate this process.

Authorization File

To enter an authorization file, select the Check Client Certificates check box in the Server Settings section. In the Authorization File text box, enter the drive, path, and file name of the file to be used for additional certificate checking. A browse button is provided to the right of the text box to facilitate this process. The authorization file supports you to exclude and include certificates based on components of the distinguished name (namely the user name, company, division, serial number, and so on) as well as by date and time. This is an optional component of SSL transfers, and can only be implemented if client authentication is performed (namely the Check Client Certificates check box is selected).

Check Remote Certificate

Select the Check Remote Certificate box if you want to have the platform server check the published Certificate Revocation List (CRL). A CRL list is a list of digital certificates, more specifically of serial numbers for certificates that have been revoked. Therefore, the SSL transfers based on revoked certificates are no longer performed. For more information on CRL, see http://www.ietf.org/rfc/rfc3280.txt.

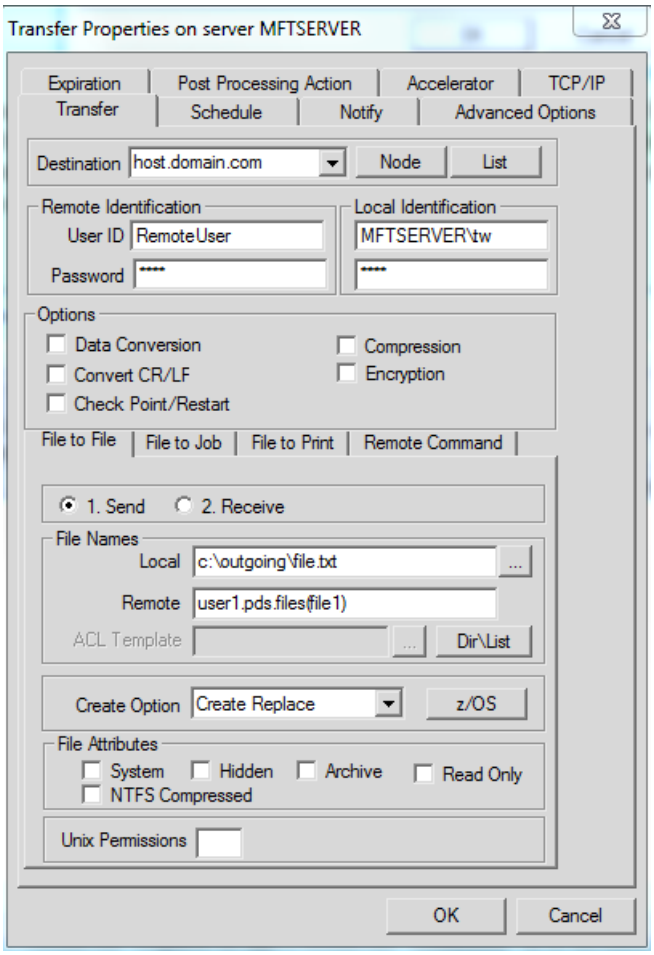
Note: If the Check Remote Certificate field is selected, the Check Client Certificate field must also be selected.

CRL Directory

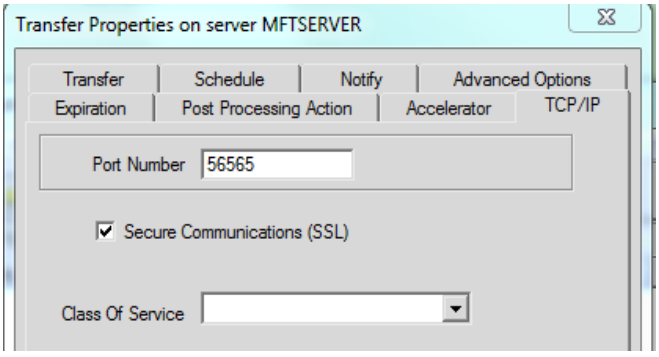
Defines the path where the CRL checking looks for the hashed file names.

4.12.4 SSL Transfer

1) Open your Advanced TCP transfer window.
2) Set your transfer. See the following example:



3) After completing the Transfer configuration, go to the TCP/IP tab.
4) Configure the SSL port used by the remote server of MFT Platform Server and enable SSL.



5) After the transfer configurations are completed, click the **OK** button and the transfer request runs.

4.12.5 SSL Authorization Parameters

MFT Platform Server supports an extension to the standard SSL processing, with which the system administrator can determine which certificates to accept and which to reject. This is done by the creation of an SSLAUTH file. This feature is supported on all MFT Platform Servers. The format of the file is the same on all platforms, but the way in which the file is defined is dependent on each platform.

See the following table for the name of the SSL authorization file on each platform.

Platform	Default Location	File Name
z/OS	SAMPLIB	SSLAUTH
Windows	C:\Program Files\TIBCO\MFT Platform Server	SslAuth
UNIX	/PlatformServer/samples	SSLAUTH

Note: The authorization file checking is in addition to the SSL authorization checking. Only when a certificate is accepted by SSL can the authorization file checking be performed.

The authorization file is compared against the certificate that is received by MFT Platform Server. The authorization file is not used on the client. The components of the Distinguished Name (DN) of the certificate are compared to the parameter in the authorization file to determine whether a certificate can be accepted. On many of the parameters, a generic character is supported. A generic character is defined in a parameter by an asterisk (*). When a generic character is defined, all characters from that point on are assumed to be a match.

If no authorization file is defined, or a match is not found in the authorization file, the request is accepted. If you want to reject all requests unless defined by the authorization file, you must insert the following statement as the last entry in the authorization file:

REVOKE

The authorization file supports the following two request types:

ACCEPT Accept an SSL request
REVOKE| REJECT Do not accept an SSL request

All of these requests accept a variety of parameters. If a parameter is not defined, it is assumed that the parameter is a match. Parameters can be defined on a single line or they can be continued over multiple lines. If the input record ends with a comma (,), the input record is continued on the next record. All parameter data is case sensitive. Be very careful when entering the values when using mixed case fields.

The following parameters are supported in the authorization file. These parameters must be defined in uppercase.

/CN Defines the Common Name defined in the Certificate. This is usually the name of the person who is requesting the certificate. Generic entries are supported.

/OU Defines the Organization Unit defined in the Certificate. This is also known as the Department. Generic entries are supported.

/O Defines the Organization defined in the Certificate. This is also known as the Company. Generic entries are supported.

/L Defines the Locality defined in the Certificate. This is also known as the City. Generic entries are supported.

/ST Defines the State/Province defined in the Certificate. Generic entries are supported.

/C Defines the Country defined in the Certificate. Generic entries are supported.

/SN Defines the Serial Number defined in the certificate. Generic entries are not supported.

/SDATE Defines the Start date for the certificate in the format of *ccyymmdd*. Generic entries are not supported. The start date is compared against the date that the transfer request is received by the platform server. If the start date is before the current date, SSLAUTH processing checks the next parameter. If the start date is after the current date, the transfer request is terminated and an error is sent to the remote system.

/STIME Defines the Start time for the certificate in the format of *hhmm*. Generic entries are not supported. The start time is only checked if the SDATE parameter exactly matches the current date. The start time is compared against the time that the transfer request is received by the platform server. If the start time is before the current time, SSLAUTH processing checks the next parameter. If the start time is after the current time, the transfer request is terminated and an error is sent to the remote system.

/EDATE Defines the End date for the certificate in the format of *ccyymmdd*. Generic entries are not supported. The end date is compared against the date that the transfer request is received by the platform server. If the end date is after the current date, SSLAUTH processing checks the next parameter. If the end date is before the current date, the transfer request is terminated and an error is sent to the remote system.

/ETIME Defines the End time for the certificate in the format of *hhmm*. Generic entries are not supported. The end time is only checked if the EDATE parameter exactly matches the current date. The end time is compared against the time that the transfer request is received by the platform server. If the end time is after the current time, SSLAUTH processing checks the next parameter. If the end time is before the current time, the transfer request is terminated and an error is sent to the remote system.

/USER This parameter is supported only by the z/OS system. It supports the administrator to define a user ID that must be used when an SSL certificate is accepted. This user ID overrides the user ID associated with the file transfer. By using this option, the remote user does not have to have any knowledge of a user ID or password on the z/OS system.

The following examples show how authorization file processing works:

Accept /OU=Marketing/O=TIBCO
revoke

MFT Platform Server accepts all certificates defined with an Organization of TIBCO and an Organization Unit of Marketing. It rejects all other certificates.

REVOKE /SN=987654
REVOKE /SN=12:34:56
ACCEPT

MFT Platform Server rejects any certificates with a serial number of 987654 or 123456. It accepts all other certificates.

Accept /OU=ACCT*/O=ACME
revoke

MFT Platform Server accepts all certificates defined with an Organization of ACME and an Organization Unit starting with ACCT. It rejects all other certificates.

Accept /CN=Joe*,

/L=New York,

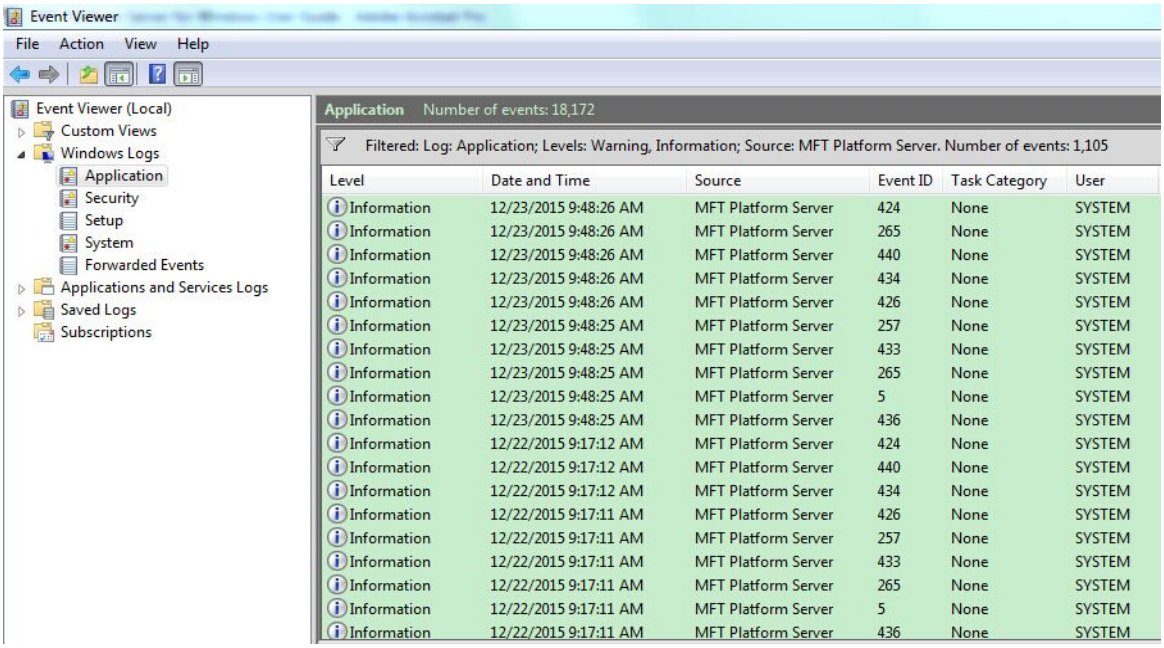
/ST=NY,
/C=US,
/OU=Dept1,
/O=ACME,
/SDATE=20051201,
/EDATE=20061130
revoke

MFT Platform Server accepts all certificates that match the information defined by the /CN, /L, /ST, /C, /OU, and /O parameters. The certificate is valid from 1 December 2005 until 30 November 2006. If the certificate is received before 1 December 2005 or after 30 November 2006, the request is rejected. All other certificates not matching these criteria are rejected.

Appendix A. The Event Log

You can use the Event Viewer to monitor events in your system. You can view and manage system, security, and application event logs. The event logging service starts automatically when you run Windows. To terminate the service, use the Services tool in Control Panel.

The Event Viewer is located in the Administrative Tools panel in Program Manager. To view the log, double-click the Event Viewer icon. The following figure shows a sample Application log.



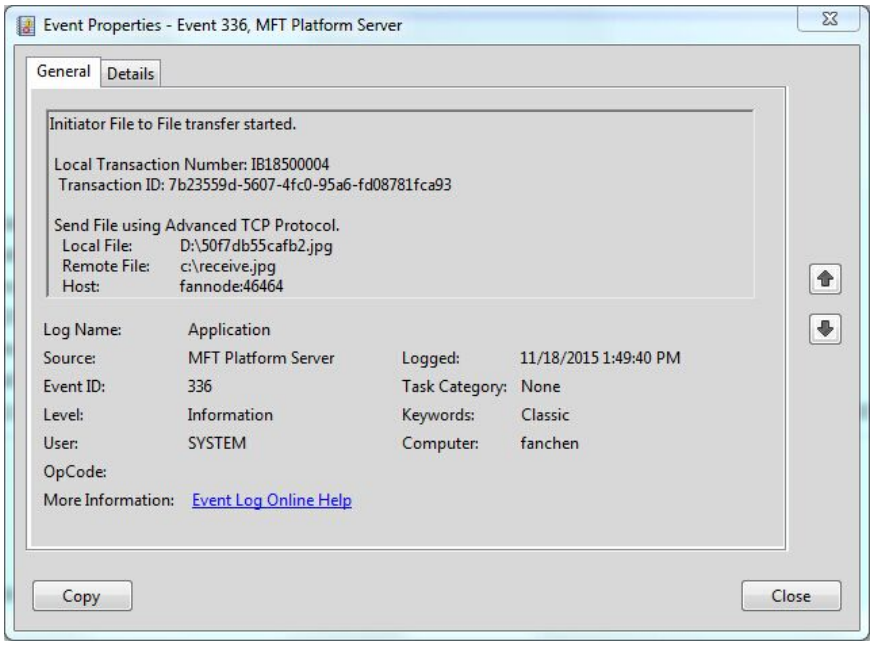
A.1 Using the Event Log

You can view three types of logs: application, system, and security. To select the log to view, select a log type from the Log pull-down menu.

Events displayed in Event Viewer are listed in sequence by date and time of occurrence. You can choose to view the events from newest to oldest (the default) or from oldest to newest.

The platform server writes the event to the event log in both successful and unsuccessful cases. The platform server also writes an informational event when the transfer begins. An informational event is also logged when the TIBCO MFT Platform Server for Windows starts. If the MFT Platform Server service is stopped, no messages for any active transfers are written to the Event Log on the machine where the service is stopped.

To view a more detailed description of an event, double-click it. The Event Properties panel is displayed.



A.1.1 Event IDs and Transaction IDs

When MFT Platform Server writes to the Windows Event Log, it provides an Event ID.

MFT Platform Server also writes transaction IDs for each of the transfers in the Windows Event log. The transaction IDs are broken down into two categories: local and remote. A transaction is assigned to one of these two categories by the MFT Platform Server initiator at the earliest possible time during the transfer. This transaction ID assigned is unique for all machines.

If a transaction is displayed in the event log before it is issued a transaction ID, the transaction does not have an ID number in the Event Log. For example, a transaction ID will not be assigned if a failure occurs before a connection to the remote system is established. The transaction is not assigned an ID by the remote system because it never actually gets to the remote system.

In addition to the transfer ID, three additional types of information are provided on the panel: message specific, error severity, and retry information.

Message specific information provides you with the details of the particular transfer ID that you are viewing at that time. This information includes remote file name, local file name, the transfer direction, and so on. Under this message information is information on the severity of the error. If the transfer fails with a severe error, this is indicated in the message. If the error is anything other than a severe error, the platform server retries the transfer if the Try Count is set to a value greater than one. If the platform server retries the transfer more than once, the retry information states the number of times that the transfer is attempted before it is completed successfully or fails.

A.1.2 Severity 1 Errors

Though the platform server can retry scheduled transfers that failed, it does not retry a severity 1 error. Severe errors would fail repeatedly.

The following errors are classified as Severity 1:

- Cannot open the source file.
    - The name is incorrectly formatted.
    - The volume name is incorrectly formatted.
    - The path is non-existent.
- Cannot open the destination ACL Template.
    - The name is incorrectly formatted.
    - The volume name is incorrectly formatted.
    - The path is non-existent.
- The destination printer name is invalid.
- Logon failure.
- File compression failed.
    - Not an NTFS formatted drive.
- Destination incorrect.
    - The IP address is incorrect.

A.2 Clearing the Event Log

When you receive a message indicating that the event log is full, you must clear the log. You can use one of the following two methods:

Method 1. Empty the current log.

1. Switch to the log whose events you want to clear.
2. From the **Action** menu, select **Clear Log**.

   You are given the option to save the current logged events.

   - If you choose to archive the events, you must select a file name and choose the directory path in which you want to store the log.
   - If you choose to not save the events, the Event Viewer empties the current log.

Method 2. Each new event replaces the oldest event.

1. From the **Action** menu, select **Log Properties**.

   The Event Log Settings panel is displayed.
2. Select **Overwrite Events as Needed**.
   When you select this option, each new event replaces the oldest event, even if the log is full.

Appendix B. Cached Passwords

When initiating transfers, you must enter credentials for local and remote users. You can cache the local and remote passwords so that you do not need to re-enter the passwords each time you execute a transfer.

The passwords are stored in a restricted area of the Windows registry. To enable this feature, you use a special set of tokens in the local or remote password fields on the initiating MFT Platform Server partner. The following four types of tokens are supported:

1. X: password
2. X:
3. X:DELETE
4. X:DELETEALL.

The tokens are case sensitive. For example, x:password (note the lowercase x) is interpreted as the user's password and not as the token (with the uppercase X) to set the cached password.

1. X:*password*

Use this token to set the local or remote passwords. As part of a file transfer, put "X:" in front of your password in the local or remote password field. For the local password, this is the password on your Windows computer. For the remote password, it is the password on the destination platform server.

When TIBCO MFT Platform Server for Windows receives this token, it strips off *password* and uses it with your user ID to log in to the Windows system. If successful, the password is encrypted and saved to a secure area of the Windows registry. After the password is saved in the registry, the transfer is executed.

2. X:

Use this token to instruct TIBCO MFT Platform Server for Windows to look up the password in the registry based upon your user ID. If the password is found, it is decrypted and used when the transfer is executed.

3. X:DELETE

Use this token to instruct the platform server to retrieve the cached password saved from a former transaction for your user ID, decrypt it, log in to Windows to conduct a transaction, and then delete the cached password from the registry. For any future transactions, the remote user must either specify a password at logon time or utilize the X:*password* to set a cached password on the Windows system.

4. X:DELETEALL

Use this token to instruct the platform server to retrieve the cached password saved from a former transaction for your user ID, decrypt it, log in to Windows to conduct a transaction, and then delete all the cached passwords from the registry.

Use *X:password* to set or change the cached password on the Windows system. If the user's Windows password changes, you must delete the old password to create a new one. Simply use the *X:newpassword* token again to overwrite the old cached password.

Note: The local cached password feature is only supported on Windows. The remote password can be used with any destination server.

The restrictions of this feature are as follows:

1. The service must be running with System Authority.
2. Because the X: token is contained within the password field, MFT Platform Server, which normally supports 20-character remote passwords, can only accept 18-character passwords.
3. Passwords that can otherwise contain X:, X:text…, X:DELETE, or X:DELETEALL are accepted as triggers to the feature and not as legitimate Windows passwords.
4. Because the passwords are saved in a restricted area of the registry, the uninstallation program cannot delete them. You must use X:DELETEALL to remove the passwords before the uninstallation. If you do not, the \\HKEY_LOCAL_MACHINE\SOFTWARE\TIBCO registry key is not removed.

Example:

This example shows how to create a batch transfer to a Windows remote system without everyone knowing the password.
In this example, the X: password token can be used to set the cached password. The following batch program invokes the cached password.

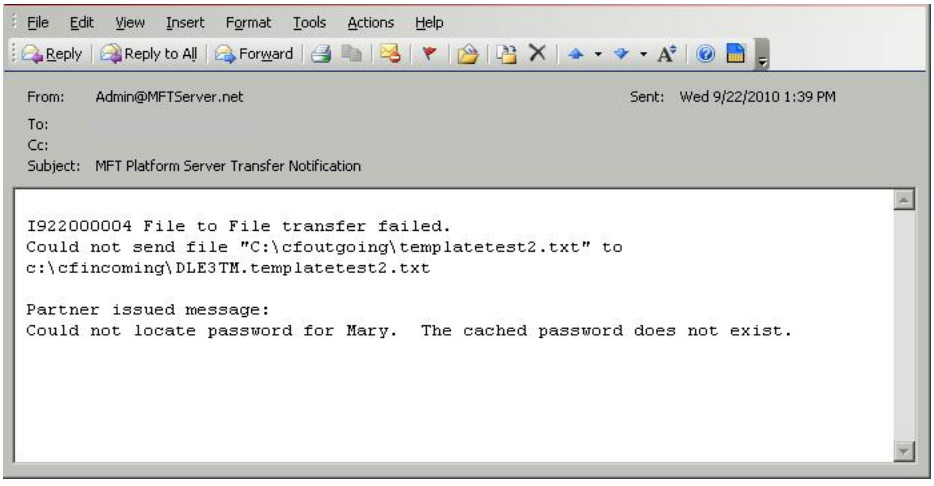
```
SET HOST=Fusion
SET PORT=46464
SET REMOTE_USER_ID=MARY
SET REMOTE_PASSWORD=X:pswdmary
SET PROCESS_NAME=MFTCMD
ftmscmd /send /file c:\abc.doc d:\abc.doc
```

For all future transfers, the remote user does not have to specify a password. Instead, the remote user can use the X: token. The remote user can use the following batch program for future transfers.

```
SET HOST=Fusion
SET PORT=46464
SET REMOTE_USER_ID=MARY
SET REMOTE_PASSWORD=X:
SET PROCESS_NAME=FTMS
```

Note: The password field and the tokens are case sensitive. If the password is in lowercase, the remote user must type “X:pswdmary”.

If the password is not yet cached, the remote user can also receive the following message:



Appendix C. File Name Tokens

Given a string of tokens, characters containing a mixture of literal and substitutable parameters, the platform server generates a formatted file name that you can use to create or read file names based on date, time, user, and file transfer information.

Instead of entering a standard file name, you enter a name that consists of tokens. You can use this feature whenever you use TIBCO MFT Platform Server for Windows.

- Section C.1 lists the available tokens.
- Section C.2 provides examples that demonstrate how to use File Name Tokens.
- Section C.3 lists the rules that you must follow when using the File Name Tokens.
- Section C.4 lists the substitutable parameters that are supported for PPA.
- Section C.5 lists the special tokens can be used for directory transfers.

C.1 File Name Tokens List

The following table lists the File Name Tokens, their respective definitions, and their generated values.

Token	Definition	Generated Value (Examples)
SYYYY	Year	0000 - 9999
SYYY	Year	000 - 999 (last 3 digits of year)
SYY	Year	00 - 99 (last 2 digits of year)
SY	Year	0 - 9 (last 1 digit of year)
SMON	Month of Year	JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
SMon	Month of Year	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
Smon	Month of Year	jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec
SMONTH	Month of Year	JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER
SMonth	Month of Year	January, February, March, April, May, June, July, August, September, October, November, December
Smonth	Month of Year	january, february, march, april, may, june, july, august, september, october, november, december
SMM	Month of Year	01 - 12
SM	Month of Year	1 - C
Sm	Month of Year	1 - c
SDD	Day of Month	01 - 31
SD	Day of Month	1 - 9, A - V
Sd	Day of Month	1 - 9, a - v
SJ	Julian Day of Year	001 - 366
SHH24	24 Hour	00 - 23
SH24	24 Hour	0 - 9, A - N
Sh24	24 Hour	0 - 9, a - n
SHH12	12 Hour	01 - 12
SH12	12 Hour	1 - C
Sh12	12 Hour	1 - c
SMI	Minute of Hour	00 - 59
SSS	Second of Minute	00 - 59
SMS	Milliseconds of Second	000 - 999
SAP	AM/PM	AM, PM
SAp	AM/PM	Am, Pm
Sap	AM/PM	am, pm
SWWW	Weekday	SUN, MON, TUE, WED, THU, FRI, SAT
SWww	Weekday	Sun, Mon, Tue, Wed, Thu, Fri, Sat
Swww	Weekday	sun, mon, tue, wed, thu, fri, sat
SWEEKDAY	Weekday	SUNDAY, MONDAY, TUESDAY,

		WEDNESDAY, THURSDAY, FRIDAY, SATURDAY
SWeekday	Weekday	Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
SW1	Weekday 1 based	1 - 7
SW0	Weekday 0 based	0 - 6

Token	Definition	Generated Value (Examples)
AllocationPrimary	Primary allocation size used in the file transfer.	Local file: c:\source\testfile1.txt Remote file: CFUSR.F$(AllocationPrimary).TEST Token resolved to: CFUSR.F800.TEST
AllocationSecondary	Secondary allocation size used in the file transfer.	Local file: c:\source\testfile1.txt Remote file: CFUSR.F$(AllocationSecondary).TEST Token resolved to: CFUSR.F500.TEST
AllocationType	Allocation type used in the file transfer.	Resolves to: Tracks, Blocks, Cylinders, Megabytes, Kilobytes
BlockSize	Block size used in the file transfer.	Local file: c:\source\testfile1.txt Remote file: CFUSR.F$(BlockSize).TEST Token resolved to: CFUSR.F6,160.TEST
CheckPointInterval	Check point used in the file transfer.	Local file: c:\source\testfile1.txt Remote file: d:\target\test$(CheckPointInterval).txt Token resolved to: d:\target\test5 minutes.txt
Compression	Compression used in the file transfer.	LZ, RLE, or NO
ComputerName	Initiator computer name.	Local file: c:\source\testfile1.txt Remote file: d:\target\$(ComputerName).txt Token resolved to: d:\target\SYSTEM3032.txt
CrLf	If CRLF is used in the file transfer.	TRUE, FALSE
DataClass	Data class used in file transfer to z/OS.	Local file: c:\source\directory\testfile1.txt Remote file: PRJOE.$(DataClass).FILE1 Token resolved to: PRJOE.DTCLS3.FILE1
DataType	Data type used in the file transfer.	BINARY, EBCDIC
Date1	The date formatted as *YYYYMMDD*.	Local file: c:\source\test.txt Remote file: d:\target\$(Date1)\test.txt Token resolved to: d:\target\20160809\test.txt
Date2	The date formatted as *MMDDYYYY*.	Local file: c:\source\test.txt Remote file: d:\target\$(Date2)\test.txt Token resolved to: d:\target\08092016\test.txt
Date3	The date formatted as *DDMMYYYY*.	Local file: c:\source\test.txt Remote file: d:\target\$(Date3)\test.txt

Token	Definition	Generated Value (Examples)
		Token resolved to: d:\target\09082016\test.txt
DateUS	The date formatted as *MMDDYYYY*.	Local file: c:\source\test.txt Remote file: d:\target\$(DateUS)\test.txt Token resolved to: d:\target\08092016\test.txt
Destination	IP or host name of the final destination for the file being transferred.	Local file: c:\source\testfile1.txt Remote file: d:\target\file1.$(Destination).txt Token resolved to: d:\target\file1.192.168.10.1.txt
FileAvailability	File availability used in file transfer.	IMMEDIATE, DEFERRED
LocalDomain	Local domain	Remote file name contains the local domain name in it.
LocalFile	Complete local file path.	Local file: c:\source\testfile1.txt Remote file: $(LocalFile) Token resolved to: c:\source\testfile1.txt
LocalFileBase	The local file name only.	Local file: c:\source\directory\testfile1.txt Remote file: $(LocalFileBase) Token resolved to: testfile1 (File transferred to the MFT Platform Server Windows directory unless a path is configured.)
LocalFileExt	Only the extension of the local file is used.	Local file: c:\source\directory\testfile1.txt Remote file: $(LocalFileExt) Token resolved to: txt (File transferred to the MFT Platform Server Windows directory unless a path is configured.)
LocalFileName	The local file name including the extension is used.	Local file: c:\source\directory\testfile1.txt Remote file: $(LocalFileName) Token resolved to: testfile1.txt (File transferred to the MFT Platform Server Windows directory unless a path is configured.)
LocalFilePath	The local file path without the file name is used.	Local file: c:\source\directory\testfile1.txt Remote file: $(LocalFilePath) Token resolved to: c:\source\directory
LocalPathWODrive	Local file path without the drive letter or file name is used.	Local file: c:\source\directory\testfile1.txt Remote file: $(LocalPathWODrive) Token resolved to: source\directory (File transferred to the MFT Platform Server Windows directory unless a drive letter is configured.)
LocalUserId	Local user ID used in the file transfer.	Local user ID: TESTLAB\cfuser1 Local file: c:\source\directory\testfile1.log Remote file: d:\target\file1$(LocalUserId).txt Token resolved to: d:\target\file1cfuser1.txt
MgmtClass	The management class to	Local file: c:\source\directory\testfile1.txt

Token	Definition	Generated Value (Examples)
	be used when transferring to a z/OS system.	Remote file: PRJOE.$(MgmtClass).FILE1 Token resolved to: PRJOE.MGCLS12.FILE1
NoLocalFileBase	The base name of the local file is not used in the file name on a send transfer.	Local file: c:\source\directory\a.b.c.txt Remote file: c:\target\$(NoLocalFileBase) Token resolved to: b.c.txt
NoLocalFileExt	The extension name of the local file is not used in the file name on a send transfer.	Local file: c:\source\directory\a.b.c.txt Remote file: c:\target\$(NoLocalFileExt) Token resolved to: a.b.c
NoRemoteFileBase	The base name of the remote file is not used in the file name on a receive transfer.	Local file: c:\target\$(NoRemoteFileBase) Remote file: c:\source\directory\a.b.c.txt Token resolved to: b.c.txt
NoRemoteFileExt	The extension name of the remote file is not used in the file name on a receive transfer.	Local file: c:\target\$(NoRemoteFileExt) Remote file: c:\source\directory\a.b.c.txt Token resolved to: b.c.txt
NotifyUser	The remote user name configured to be notified in the file transfer.	Local file: c:\source\directory\testfile1.txt Remote file d:\target\file1$(NotifyUser).txt Token resolved to: d:\target\file1JohnD.txt
NotifyUserType	The type of notification used for the remote user notification in the file transfer.	Local file: c:\source\directory\testfile1.txt Remote file: d:\target\file1$(NotifyUserType).txt Token resolved to: d:\target\file1Windows.txt (None, Email)
PortNumber	The port number used in the file transfer.	Local file: c:\source\directory\testfile1.txt Remote file: d:\target\file1$(PortNumber).txt Token resolved to: d:\target\file146,464.txt
PrinterName	The printer name used in File to Print.	<text>
Priority	The priority set in the file transfer.	Local file: c:\source\directory\testfile1.txt Remote file: d:\target\file1$(Priority).txt Token resolveds to: d:\target\file1Normal.txt
ProcessName	The process name configured in the file transfer.	Local file: c:\source\directory\testfile1.txt Remote file: d:\target\file1$(ProcessName).txt Token resolved to: d:\target\file1CyberFus.txt
RecordFormat	Record format used in the file transfer.	FIXED, BLOCKED, FIXED BLOCKED, VARIABLE, VARIABLE BLOCKED, UNDEFINED
RecordLength	Record length used in the file transfer.	Local file: c:\source\testfile1.txt Remote file: CFUSR.F$(RecordLength).TEST

Token	Definition	Generated Value (Examples)
		Token resolved to: CFUSR.F80.TEST
RemoteDomain	Remote domain used in the file transfer.	Remote file name contains the remote domain name within it.
RemoteFile (Token used when doing a receive)	Complete remote file path.	Local file: $(RemoteFile) Remote file: c:\source\testfile1.txt Token resolved to: c:\source\testfile1.txt
RemoteFileBase (Token used when doing a receive)	The remote file name only.	Local file: $(RemoteFileBase) Remote File: c:\source\directory\testfile1.txt Token resolved to: testfile1 (File transferred to the MFT Platform Server Windows Directory unless a path is configured.)
RemoteFileExt (Token used when doing a receive)	Only the extension of the remote file is used.	Local file: $(RemoteFileExt) Remote file: c:\source\directory\testfile1.txt Token resolved to: txt (File transferred to the MFT Platform Server Windows Directory unless a path is configured.)
RemoteFileName (Token used when doing a receive)	The remote file name including the extension is used.	Local file: $(RemoteFileName) Remote file: c:\source\directory\testfile1.txt Token resolved to: testfile1.txt (File transferred to the MFT Platform Server Windows Directory unless a path is configured.)
RemoteFilePath (Token used when doing a receive)	The remote file path without the file name is used.	Local file: $(RemoteFilePath) Remote file: c:\source\directory\testfile1.txt Token resolved to: c:\source\directory
RemotePathWODrive (Token used when doing a receive)	Remote file path without the drive letter or file name is used.	Local file: $(RemotePathWODrive) Remote File: c:\source\directory\testfile1.txt Token resolved to: source\directory (File transferred to the MFT Platform Server Windows directory unless a drive letter is configured.)
RemoteTransactionNumber	Remote transaction number used in the file transfer.	Local file: d:\fn\$(RemoteTransactionNumber).txt Remote file: c:\source\directory\testfile1.txt Token resolved to: d:\fn\IC21500000.txt
RemoteUserId	Remote user ID used in the file transfer.	Remote user ID: TEST\cfuser1 Local file: c:\fn\file1.$(RemoteUserId).txt Remote file: c:\source\directory\testfile.txt Token resolved to: c:\fn\file1.cfuser1.txt
StorageClass	Storage Class used when performing a file transfer to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote file: PRJOE.$(StorageClass).FILE1 Token resolved to: PRJOE.STANDARD.FILE1
SysoutClass	The SYSOUT class used when performing a File to Print to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote file: PRJOE.$(SysoutClass).FILE1 Token resolved to: PRJOE.A.FILE1
SysoutCopies	The amount of SYSOUT copies used when	Local file: c:\source\directory\testfile1.txt Remote file:

Token	Definition	Generated Value (Examples)
	performing a File to Print to a z/OS system.	PRJOE.TS$(SysoutCopies).FILE1 Token resolved to: PRJOE.TS2.FILE1
SysoutDestination	The SYSOUT destination used when performing a File to Print to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote file: HST.$(SysoutDestination).FILE1 Token resolved to: HST.NYPRINTER.FILE1
SysoutFcb	The SYSOUT FCB used when performing a File to Print to a z/OS system	Local file: c:\source\directory\testfile1.txt Remote file: PRJOE.$(SysoutFcb).FILE1 Token resolved to: PRJOE.STD2.FILE1
SysoutForms	The SYSOUT forms used when performing a File to Print to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote file: PRJOE.$(SysoutForms).FILE1 Token resolved to: PRJOE.INVC.FILE1
SysoutUserId	The SYSOUT user name used when performing a File to Print to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote file: PRJOE.$(SysoutUserId).FILE1 Token resolved to: PRJOE.MVSUSER1.FILE1
SysoutWriter	The SYSOUT writer used when performing a File to Print to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote file: PRJOE.$(SysoutWriter).FILE1 Token resolved to: PRJOE.WRITER1.FILE1
TransactionNumber	Local transaction number used in the file transfer	Local file: c:\source\directory\testfile1.txt Remote file: d:\target\f$(TransactionNumber).txt Token resolved to: d:\target\f1331600053.txt
TransferFunction	The transfer function used in the file transfer.	SEND, RECEIVE
TransferId	The transfer ID assigned to the file transfer.	Local file: c:\source\directory\testfile1.txt Remote file: d:\target\file1.$(TransferId).txt Token resolved to: d:\target\file1.d1544fd2-5fb7-4ce6-a717-ac8907697e4f.txt
TransferWork	The type of transfer being done. For example, File to File, File to Job, and so on.	F-FILE, J-JOB, P-PRINT
TryCount	Try count used in transfer.	Local file: c:\source\directory\testfile1.txt Remote file: d:\target\file1$(TryCount).txt Token resolved to: d:\target\file13 Times.txt
Unit	Unit used for transfer to and from a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote file: PRJOE.$(Unit).FILE1 Token resolved to: PRJOE.SYSDA.FILE1
UserData	The user data name used in the file transfer.	Local file: c:\source\directory\testfile1.txt Remote file: d:\target\file1$(UserData).txt Token resolved to: d:\target\file1MyUserData.txt
VolSer	Volume used for transfer to and from a z/OS	Local file: c:\source\directory\testfile1.txt Remote file: PRJOE.$(VolSer).FILE1

Token	Definition	Generated Value (Examples)
	system.	Token resolved to: PRJOE.CFP101.FILE1
WriteMode	The write mode used in the file transfer.	Create, Replace, Append, Create Replace, Create Append, Create Replace New

C.2 Examples of Using the File Name Tokens

When transferring a file, you can type the name of the file using file name tokens instead of a regular file name. These examples use the following sample system date/time:

```
Wednesday, April 25, 1996 5:03:45.061 PM
```

- In this example, instead of entering a standard file name, a string of file name tokens are entered. MFT Platform Server or Responder resolves the string into the directory name and file name.

File name	`C:\directory\$(SDD)$(SMON)$(SYYYY)\$(SHH24)$(SMI)$(SSS).dat`
Resolved file name	`C:\directory\25APR1996\170345.dat`

- In this example, the file name tokens are used to generate a resolved file name that has dashes between the date and time fields.

File name	`C:\directory\$(SDD)-$(SMON)-$(SYYYY)\$(SHH24)-$(SMI)-$(SSS).dat`
Resolved file name	`C:\directory\25-APR-1996\17-03-45.dat`

- In this example, MFT Platform Server or Responder resolves the tokens in the file name into a long file name using uppercase and lowercase letters.

File name	`\\Server\Volume\$(SMonth)\projectX\$(SWeekday)\products.xls`
Resolved file name	`\\Server\Volume\April\projectX\Wednesday\products.xls`

- In this example, the template is used to create a DOS 8.3 formatted file name whose 3-character extension contains an encoded representation of the date. The number of the day of the week is also used as part of the file name. In this case, the 0-based version is used. The 1-based day of week is also provided.

File name	`C:\DOS\SHORTNM$(SW0).$(SM)$(SD)$(SY)`
Resolved file name	`C:\DOS\SHORTNM4.4P6`

- In this example, DNI and file name tokens are used. The sample.txt file is placed in the DNI directory and the file name tokens are used to designate the directory and file name of the transferred file.

File name	`C:\$(RemoteUserId)\$(LocalFileName)\$(LocalPathWODrive)`
Resolved file name	`C:\pat\sample.txt\$(RemotePathWODrive)`

Note:

- Using $(LocalPathWODrive) or $(RemotePathWODrive) takes the path specified in the file name and transfers the file to the same directory, but different drive.
- The various time tokens available are resolved at the beginning of a file transfer from the Initiator. As a result, if a file transfer fails and goes into retries, the initial file name that is set does not change even though the transfer can be done at a later time because of retries.

C.3 Rules for Use

When you create a file name that uses file name tokens, you must follow the following rules:

- Substitution parameters are enclosed in $(*tokenname*).
- Each $(*tokenname*) can contain one token only.
- Any text in the remote file name which is not a substitution parameter is embedded as is into the generated name.
- Tokens can be used anywhere within the remote file name (as the file name or directory name, share name, or server name).
- Space permitting, any number of substitution parameters can be embedded within the file name.
- If the resolved remote file name length is greater than the maximum file name length allowed by MFT Platform Server on Windows (255 characters), it is truncated.
- If the transfer type is initiator send, the remote file name resolves to the destination file for the transfer.
- If the transfer type is initiator receive, the remote file name resolves to the source file for the transfer.
- Capitalization of the substitution parameters effects the capitalization of the output.
- If a formatted name which contains an invalid substitution parameter is given, the transfer fails with an error stating that a substitution parameter is invalid.
- The feature is designed to work with DOS 8.3 and Win32 long file names. You must ensure that the generated name is valid for the target system. Be careful when using forward slash (/), backslash (\), or colon (:) to delimit dates and times because these signs contain special meaning to the operating system.
- For remote systems which support long file names, embedded spaces are valid for the generated file name; however, TIBCO MFT Platform Server for z/OS currently does not support embedded spaces in remote file names.

C.4 PPA Tokens

The PPA substitutable parameters use the percent sign (%) as the escape character instead of the dollar sign ($) that file tokens use. The following table shows the substitutable parameters that are supported for PPA.

In this example, the file is called C:\a\b\c\d\config.txt.

Token	Description	Resolved Name Example
%DIR	The directory without the file name or drive	a\b\c\d sharename\a\b\c\
%DRIVE	Drive name	C \\server\
%NODRIVE	File name without drive	a\b\c\d\config.txt \sharename\a\b\config.txt
%SDIR	The lowest-level directory	d
%HDIR	The high level directory	a
%NOSDIR	Directory name without the lowest directory	a\b\c
%NOHDIR	Directory name without the high level directory	b\c\d
%FILE	File name without the directory	config.txt
%LFILE	File name with the directory	C:\a\b\c\d\config.txt \\server\sharename\a\test.txt
%LLQ	Low level qualifier of the file (data after the last period (**.**))	txt
%HLQ	High level qualifier of the file	config
%TRN	Transaction number	I824500001
%PROC	Process name	ABC123
%UDATA	User data	USRDATAABC123
%JDATE	Julian date (*YYDDD*)	05236
%JDATEC	Julian date with century (*CCYYDDD*)	2005236
%TIME	Time (*hhmmss*)	165030
%GDATE	Gregorian date (*yymmdd*)	050824
%GDATEC	Gregorian date with century (*ccyymmdd*)	20050824

You can use multiple PPA parameters within a single PPA data field. Each substitutable parameter must be processed one at a time before going onto the next byte of PPA data.

Some fields do not make sense such as %DRIVE in a UNIX environment. If a field does not make sense in the environment where PPA is used, the substitutable data is the text in the name of the parameter without the percent sign (%). If UNIX detects the %DRIVE parameter, the value DRIVE is used as substitution. Similarly, %PROC becomes PROC and %UDATA becomes UDATA if not interacting with a z/OS system.

C.5 Directory Tokens

You can use two special tokens in directory transfers.

$(SDIR)

This case-sensitive token can be used as part of the LocalFileName path in a receive transfer, and as part of the RemoteFileName path in a send transfer.

Example:

For a receive transfer, set as follows:

LocalFileName:

```
C:\johndoe\data\$(SDIR)\$(RemoteFileName)
```

RemoteFileName:

```
C:\MFT Platform Server\data\*
```

The text before this token is assumed to be a base directory.

If **ScanSubDir** is selected and files exist in both the remote directory (C:\MFT Platform Server\data) and in the remote subdirectories, the same subdirectories are created in the local directory (C:\johndoe\data) and local file names are given as the **$(RemoteFileName)** token.

If this token is missing but **ScanSubDir** is selected, all the files from the remote directory and all subdirectories are located at the local base directory. Their names are given by the **$(RemoteFileName)** token.

Subdirectories are created with the same access rights as the base directory. If some of the directories do not exist at the base directory path (for example, the directory data from LocalFileName), it is created with the same access rights as its base directory (johndoe), and all directories after it are created under it with the same access rights.

For a send transfer, **$(SDIR)** can be used as part of the **RemoteFileName** path, in the form of **C:\MFT Platform Server\data\$(SDIR)\$(LocalFileName)**.

If the remote side (as on z/OS) has no subdirectory structures, files from the remote side are placed in the local base directory and **$(SDIR)** is ignored.

$(MEMBER)

This token can only be used for a receive transfer from a z/OS system. It is used for a similar purpose as the **$(SDIR)** token, but this token is used because data set names work differently than directory names.

By using this token, you can have file names on the local side that are the same as Member names on the z/OS side.

If the file name from the z/OS side does not contain $(Member), this token is not used. For example, if the path is C:\MFT Platform Server\$(MEMBER)\whatever, it becomes C:\MFT Platform Server\whatever.