



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

*Version 8.0.1*

*January 2022*



# Contents

---

- Sample Transfer Using MFT Platform Server Administrator . . . . . 6**
- Transfer Properties . . . . . 9**
  - Transfer Tab . . . . .9
  - Universal Fields . . . . . 10
  - File to File Tab . . . . .12
    - z/OS Options Panel . . . . . 15
      - Record Format Tab . . . . . 15
      - Allocation Tab . . . . . 17
      - Disk Tab . . . . .17
      - Other Tab . . . . . 18
  - Schedule Tab . . . . . 19
  - Notify Tab . . . . .20
  - Advanced Options Tab . . . . . 23
  - Expiration Tab . . . . . 27
  - Post Processing Action Tab . . . . . 28
    - Substitutable Parameters . . . . . 30
  - Accelerator Tab . . . . . 31
  - TCP/IP Tab . . . . . 33
- The Network View . . . . .34**
  - Buttons . . . . . 34
- Past Transactions . . . . .38**
- Notification . . . . .39**
- Server Properties . . . . .40**
  - General Tab . . . . . 40
  - Responder Tab . . . . . 43
  - Throttle . . . . . 45
  - Trace Tab . . . . . 45
  - Accelerator . . . . . 48
  - Service Control Manager . . . . . 49
  - View - Options . . . . . 50
    - Options - General Tab . . . . . 50
    - Options - Administrator Trace Tab . . . . . 51
- MFT Platform Server Monitor . . . . . 53**
- Command Line Interface . . . . . 55**
  - Command Line Format . . . . . 55
  - Specifying Command Line Parameters . . . . . 55

- File to File Transfers ..... 56
- File to Job Transfers ..... 57
- File to Print Transfers ..... 58
  - Specifying a Printer Name ..... 58
    - Printer Name Parameters ..... 58
- Remote Command Transfers ..... 61
- Parameters ..... 61
  - Optional Parameters ..... 61
- Use of Errorlevel with FTMSCMD ..... 85
- Extended Features ..... 87**
  - Access Control ..... 87
    - Access Control Parameters ..... 87
      - Directory Name Used in Request ..... 89
      - Continuation and Comments ..... 90
      - Default Entries ..... 90
      - Parameter Validation ..... 90
    - Sample of AccessControl.cfg File ..... 91
  - CFAlias ..... 91
    - CFAlias Parameters ..... 91
    - Substitutable Parameters ..... 92
    - Example of CFAlias Configuration ..... 93
    - Sample of CfAlias.cfg File ..... 93
  - CFINQ ..... 94
    - Log Files ..... 94
    - CFINQ Program ..... 94
    - CFINQ Parameters ..... 94
    - Example of Using CFINQ Utility ..... 97
  - Configured Post Processing ..... 100
    - Configuration Parameters ..... 101
    - Sample of CfgPostProc.cfg File ..... 102
    - Arguments for Substitution ..... 102
  - Custom Code Page Conversion ..... 102
    - ASCII to EBCDIC Conversion Table Example ..... 103
    - Definition of Your Own Tables ..... 106
    - Additional Information for Data Conversion ..... 108
  - Directory Named Initiation (DNI) GUI and Command Line Interface ..... 108
    - DNI GUI Interface ..... 108
      - Transfer Template ..... 109
        - Creating a transfer template ..... 109

|   |     |
|---|-----|
| Advanced TCP Template Definition .....                                  | 110 |
| Advanced Batch Template Definition .....                                | 110 |
| File Name Tokens .....  | 113 |
| The Initiation Directories Properties Sheet .....                       | 113 |
| Directory Initiation Property Page .....                                | 114 |
| Schedule Property Page .....  | 115 |
| DNI Command Line Interface (CLI) .....                                  | 117 |
| fusping Utility .....   | 117 |
| Format of fusping Commands .....  | 117 |
| Examples of Using fusping Utility .....                                 | 118 |
| fusutil Utility .....   | 118 |
| Format of fusutil Commands .....  | 119 |
| Examples of Using fusutil Utility .....                                 | 119 |
| Special Processing .....  | 120 |
| Return Codes .....  | 120 |
| Nodes, Profiles, and Distribution Lists .....                           | 120 |
| Node Definitions .....  | 121 |
| Node Parameters .....   | 122 |
| Examples of Using cfnode Utility .....                                  | 127 |
| Profile Definitions .....   | 130 |
| User Profiles .....   | 130 |
| Examples of Using cfprofile Utility .....                               | 130 |
| Responder Profile Parameters .....                                      | 131 |
| Examples of Using cfrprofile Utility .....                              | 132 |
| Distribution Lists .....  | 133 |
| Distribution Parameters .....   | 133 |
| TIBCO Accelerator .....   | 134 |
| TIBCO Accelerator Ports .....   | 134 |
| Usage of TIBCO Accelerator within MFT Platform Server .....             | 134 |
| Example 1: Windows to Windows Using TIBCO Accelerator for Windows ..... | 134 |
| Example 2: z/OS to UNIX Using TIBCO Accelerator for Windows .....       | 135 |
| SSL .....   | 139 |
| SSL or TLS Transfers .....  | 139 |
| SSL Utility .....   | 140 |
| Certificate Creation .....  | 140 |
| Certificate View .....  | 141 |
| SSL Configuration .....   | 142 |
| SSL Settings .....  | 143 |
| Using SSL/TLS Transfer .....  | 146 |

|   |            |
|---|------------|
| SSL Authorization Parameters .....                    | 147        |
| <b>Event Logs .....</b>                               | <b>150</b> |
| Viewing the Event Log .....                           | 150        |
| Event IDs and Transaction IDs .....                   | 151        |
| Severity 1 Errors .....                               | 151        |
| Clearing an Event Log .....                           | 152        |
| Emptying the Current Log .....                        | 152        |
| Replacing the Old Event with a New Event .....        | 152        |
| <b>Cached Passwords .....</b>                         | <b>153</b> |
| <b>File Name Tokens .....</b>                         | <b>155</b> |
| File Name Tokens List .....                           | 155        |
| Examples of Using File Name Tokens .....              | 164        |
| Rules for Use .....                                   | 165        |
| PPA Tokens .....                                      | 166        |
| Directory Tokens .....                                | 167        |
| <b>TIBCO Documentation and Support Services .....</b> | <b>169</b> |
| <b>Legal and Third-Party Notices .....</b>            | <b>170</b> |

# Sample Transfer Using MFT Platform Server Administrator

You can use MFT Platform Server Administrator to do a simple file to file transfer.

When you start MFT Platform Server Administrator, you are automatically attached to your TIBCO Managed File Transfer (MFT) Platform Server for Windows. To do a transfer, configure related transfer properties in the Transfer Properties on server dialog. Required fields include:

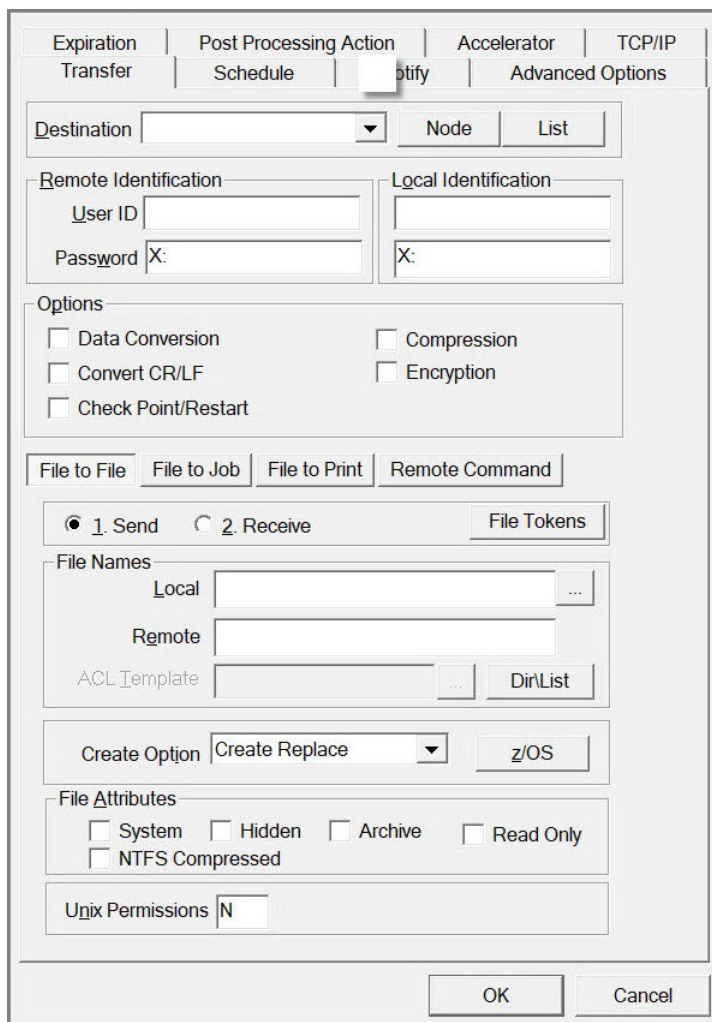
- **Destination** field
- Fields in the **Remote Identification** area
- **Local** field and the **Remote** field in the **File Names** area

Optional configurations include:

- **Data Conversion** check box
- **Check Point/Restart** check box
- **Compression** check box
- **Encryption** check box

You can also specify the direction of the file transfer by clicking **1. Send** or **2. Receive** in the **File to File** tab.

One way to get to the transfer GUI is to click the **New Transfer** icon  and then click **Advanced TCP Transfer**. Then the Transfer Properties on server dialog appears.



Expiration | Post Processing Action | Accelerator | TCP/IP

Transfer | Schedule | Notify | Advanced Options

Destination  Node  List

Remote Identification

User ID

Password  X:

Local Identification

X:

Options

☐ Data Conversion ☐ Compression

☐ Convert CR/LF ☐ Encryption

☐ Check Point/Restart

File to File | File to Job | File to Print | Remote Command

☒ 1. Send ☐ 2. Receive File Tokens

File Names

Local  ...

Remote

ACL Template  ... Dir/List

Create Option  z/OS

File Attributes

☐ System ☐ Hidden ☐ Archive ☐ Read Only

☐ NTFS Compressed

Unix Permissions

OK Cancel

For simplicity, take a file to file transfer from Windows to a mainframe as an example. You can conduct configurations in the **Transfer** tab as follows:

- **Destination:** You can use the **Destination** field to specify the remote system to which you send a file. The destination can be an IP name or address of the z/OS running on the mainframe. The specific field value depends on the remote system and the protocol that is used for the transfer. The value is kept in a drop-down list so that it can be used for future transfers.

You can also pre-define the remote system, which is referred to as a node. By clicking **Node** next to the **Destination** field, a list of all pre-defined nodes is available in the **Available Nodes** dialog. After you select a node from the list and click **OK**, the value in the **Destination** field is filled in automatically.


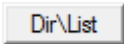
- **Local Identification:** In the Local Identification area, you can specify the user ID and password of a user on the local system. The transfer is submitted under the user and executes with the rights of the user. The entered password is shown as asterisks.
- **Remote Identification:** In the **Remote Identification** area, you can specify the user ID and password on the remote system. Therefore, if you send a file to the mainframe, place your mainframe user ID and password in this area. The entered password is shown as asterisks.
- **Options:** The **Options** area contains the following check boxes:
  - **Data Conversion** check box: has three additional fields so that data can be converted to or from ASCII or EBCDIC on a local or remote system.
  - **Check Point/Restart** check box: has an additional parameter called **Interval** in the **Check Point** area of the **Advanced Options** tab. You can use the **Interval** parameter to specify the interval at which MFT Platform Server for Windows takes a checkpoint
  - **Convert CR/LF** check box
  - **Compression** check box: has an additional parameter called **Type** in the **Compression** area of the **Advanced Options** tab.

The type of compression can be:

- RLE
- LZ
- ZLIB1 - ZLIB9
- Default from Node
- None
- **Encryption** check box: has an additional parameter called **Method** in the **Encryption** area of the **Advanced Options** tab.


The method of encryption can be:

- DES
- 3DES
- Blowfish
- Blowfish Long
- AES(Rijndael)
- Default
- None
- **File to File:** You need to conduct transfer configurations as required in the **File to File** tab.
  - **1. Send** or **2. Receive:** Click **1. Send** or **2. Receive** to specify the direction of a file to file transfer. In this example, click **1. Send** to initiate a Send transfer.

- **File Names:** Specify the local file name and remote file name in the **File Names** area.
- **Local:** Specify the name of a file to be sent on your local Windows machine in the **Local** field.  
  
You can also click  next to the **Local** field to select a file from your machine or network. This operation is particularly helpful in eliminating errors while typing the directory or file name.
- **Remote:** Specify a mainframe dataset name in the **Remote** field. The name can be an existing dataset or a new dataset name. In this example enter a new dataset name.
- **ACL Template:** The **ACL Template** field is available only when **2. Receive** is selected. This field allows you to have the same security attributes on a file that you are receiving as the file entered in the ACL Template field.
- **Dir\List:** MFT Platform Server for Windows has the ability to transfer entire directories. You can click  to check the options for directory transfers, such as the ability to scan subdirectories and stop on failure.
- **Create Option:** The **Create Option** parameter indicates whether a file exists. You have the option to create it if it does not exist, to replace it if it exists, to append to an existing file, and so on.
- **z/OS:** You can use the **z/OS** button next to the **Create Options** field to define attributes for the file that you send to the mainframe, such as the record format, record length, block size, allocation, and so on.
- **File Attributes:** You can use the check boxes in the **File Attributes** area to define attributes for the file that you receive to Windows.

You can specify all the information for a basic file transfer in the **Transfer** tab. However, more transfer options are in other tabs of the Transfer Properties on server dialog. The information includes scheduling, defining the compression to be used (when compression is selected on the main GUI panel), the port used with a TCP transfer.

After specifying all information for a file transfer, click **OK** to initiate the transfer. You can then click the

**New Transfer** icon  to check the progress of the file transfer.




# Transfer Properties

---

When you first initialize the Interactive Interface, the MFT Platform Server Administrator automatically connects to your TIBCO MFT Platform Server for Windows. You can configure the transfer properties on the Transfer Properties panel.

To open the Transfer Properties panel, you can choose either of the following two ways:

- Click  on the toolbar, and then select a protocol you want to use for the transfer.
- Right-click **Transfer** under the server name, and then click **new > Advanced TCP Transfer**.

## Transfer Tab

The Transfer tab contains two different halves. The top half of the panel has several fields that are universal to all transfer types. The lower half of the panel consists of four tabs.

Each tab represents a different type of transfer that is supported by MFT Platform Server for Windows: File to File, File to Job, File to Print and Remote Command. The GUI elements on the lower half panel vary depending on the different tabs.

Expiration

Post Processing Action

Accelerator

TCP/IP

Transfer

Schedule

Notify

Advanced Options

Destination

Node

List

Remote Identification

User ID

Password X:

Local Identification

X:

Options

☐ Data Conversion

☐ Compression

☐ Convert CR/LF

☐ Encryption

☐ Check Point/Restart

File to File

File to Job

File to Print

Remote Command

☒ 1. Send

☐ 2. Receive

File Tokens

File Names

Local

Remote

ACL Template

Dir/List

Create Option

Create Replace

z/OS

File Attributes

☐ System

☐ Hidden

☐ Archive

☐ Read Only

☐ NTFS Compressed

Unix Permissions

N



OK

Cancel

Universal Fields

The universal fields are the fields located on the top half of the transfer panel.

| Elements    | Description   |
|-------------|---|
| Destination | <div>The address of the remote system.</div> <div><ul style="list-style-type: none"><li>This is the TCP/IP DNS Name, IPV4 Address (for example 251.250.41.5) of IPV6 Address (for example ::1)</li></ul></div> <div>This field has a drop-down list that is designed to keep a list of the remote systems used in the past. A pre-defined Node can be used in the this field.</div> |

| Elements              |          | Description   |
|-----------------------|----------|---|
| Node                  |          | <p>The name of the remote system defined using the <code>cfnode</code> program that is provided with MFT Platform Server for Windows.</p> <p>If a Profile is associated with the Node, the Remote Identification area is filled in with <code>Default</code> from node. If no profiles are found, the area is blank. When you type a Node in the <b>Destination</b> field, the user ID and password will be picked up from the profile definition to corresponding fields if it is defined.</p> <p>If any of the transfer settings conflict with the node settings, a notification is displayed. The transfer can be modified by clicking <b>OK</b> or <b>Cancel</b>.</p>   |
| List                  |          | <p>Displays a list of distribution defined in the <code>cflist.cfg</code> file, which is located in the MFT for Windows installation directory.</p> <div>  <p>The distribution lists are supported for SEND transactions only.</p> </div>  |
| Remote Identification | User ID  | <p>The user ID for the remote system, or the name by which the issuer is known to the remote system. The user ID is up to 36 characters in length which includes fifteen characters for a machine name or domain, a slash and up to 20 characters for the ID. It is generally not case sensitive, unless on a UNIX system.</p> <p>The User ID defaults to the last Issuer ID entered in this field. If a Node is selected and there is a Profile associated with the Node, this field is filled in with <code>Default</code> from node.</p>   |
|                       | Password | <p>The remote password is up to 20 characters in length and case sensitive.</p> <p>For security reasons, this field is not saved in the registry as other values. It remains in the panel during the Transfer Properties GUI execution but need to be reset at the next startup of the Transfer Properties GUI.</p> <p>A feature called <code>cached passwords</code> allows you to specify a password for a particular remote Windows User ID and store the password in the Windows registry on the remote system. You can perform MFT Platform transfers to that Windows system without having to specify the password. For more information on this feature, see <a href="#">Cached Passwords</a>.</p> <div>  <p>If your password on a remote z/OS system has expired, you will be unable to access a z/OS file from MFT Platform Server Administrator. In order to change the password, specify both the old password and the new password in this field, separated by a slash. For example, <code>old/new</code>. This changes the z/OS password to the new one specified.</p> </div> |
| Local Identification  | User ID  | <p>The user ID of the local authentication credentials for transfers. It is up to 36 characters in length which includes fifteen characters for a machine name or domain, a slash and up to 20 characters for the ID.</p>   |

| Elements |                       | Description   |
|----------|-----------------------|---|
|          | Password              | <p>The Local Identification is set to the user ID of the logged on user. The default value for the password is X:, which causes the MFT 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 enter a password or any of the other cached password keys: X:password, X:DELETE, X:DELETEALL, or X:.</p> <p>For more information on cached passwords, see <a href="#">Cached Passwords</a>.</p>   |
| Options  | Data Conversion       | Converts data between ASCII and EBCDIC. Transfers can be either binary or text. If this checkbox is cleared, the transfer is a binary transfer. Otherwise, it is a text transfer. There are additional parameters under the <b>Advance Options</b> tab. If you want to use this feature, select the check box and give details under the <b>Advanced Options</b> tab. See <a href="#">Advanced Options Tab</a> .  |
|          | Convert CR/LF         | Inserts an end-of-line character when you are receiving a file from the z/OS and removes those characters when you are sending a file to the z/OS.  |
|          | Check Point / Restart | Allows packets of data to 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 checkpoint of the sender 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 <b>Advanced Options</b> tab. See <a href="#">Advanced Options Tab</a> . |
|          | Compression           | <p>Allows to specify that compression used for this transfer. Select the check box to turn compression on, and then select the type of compression to be used for the transfer under the <b>Advanced Options</b> tab.</p> <p>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.</p>  |
|          | Encryption            | Allows to turn encryption on and off. Select the check box to turn encryption on, and then select the Method of Encryption to be used for the transfer under the <b>Advanced Options</b> tab.   |

## File to File Tab

The **File to File** tab stores the contents of the file transfer in a file.

The elements under the tab are shown below.

| Elements    |              | Description  |
|-------------|--------------|--|
| 1. Send     |              | Initiates the send request to the remote system.   |
| 2. Receive  |              | Initiates the receive request from the remote system.  |
| File Tokens |              | Displays a list of file tokens that are supported. Tokens can be copied from this page into the local and remote file names.   |
| File Names  | Local        | The name by which a file is known at the local side.<br><br>Click  to browse and select the file. MFT Platform Server for Windows supports the standard file names as well as UNC file names.  |
|             | Remote       | The name by which a file is known on the remote side.  |
|             | ACL Template | <p>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.</p> <p>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 created must reside on an NTFS drive.</p> <p>The browse button  is available when the direction of the transfer is Receive.</p> |

| Elements        |                    | Description  |
|-----------------|--------------------|--|
|                 | Dir\List           | <p>MFT Platform Server for Windows has the ability to transfer entire directories as well as send to a distribution list. The Dir/List button gives the options for a directory transfer or transfer sent to a distribution list the ability to stop on failure.</p> <ul style="list-style-type: none"> <li>• StopOnFailure: When this check box is selected, it will not try to transfer the rest of files if the current file transfer fails</li> <li>• ScanSubDir: When this check box is selected, it causes not only the directory from the file path to be scanned, but all subdirectories as well. (Not available for List transfers.)</li> <li>• Test: When this check box is selected, it allows to display the Local and Remote File Names rather than do the actual transfers as a means of verifying that the file names are correct.</li> </ul> |
| Create Option   | Create             | Creates a file on the remote system 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 on the remote system, 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     | If the file does not exist on the system, it is created. If the file does exist, replace its contents with the contents of the source file.  |
|                 | Create Append      | If the file does not exist on the system, it is created. If the file does exist, append the contents of the source file to the end of the destination file.  |
|                 | Create Replace New | Creates new files, replaces existing files, and if the path to the new file does not exist, creates the path as part of the transfer.  |
|                 | z/OS               | This button is only available on the File to File tab. Click this button to select the z/OS file creation options when sending files to MFT Platform Server for z/OS partners. After clicking it, the z/OS Options Panel is displayed. See <a href="#">z/OS Options Panel</a> for more information.  |
| File Attributes | System             | Indicates that the file is a system file and can only be viewed by the operating system and not by the user.   |

| Elements         | Description   |
|------------------|---|
| Hidden           | A file that cannot be seen by the user.   |
| Archive          | Marks a file that has changed since it was last backed up.  |
| Read Only        | Indicates that 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, MFT Platform Server for Windows has the ability to set the UNIX Permissions on the file. UNIX permissions are defined by a three digit number such as 777 (the same as for the <b>chmod</b> command). The default value for this parameter is the file permissions of the file being sent or received. |

### z/OS Options Panel

The z/OS Dynamic Allocation Parameters window contains four tabs are necessary to specify when the user is sending files to an MFT Platform Server for z/OS partner.

### Record Format Tab

The **Record Format** tab contains the following elements:


| Elements   | Description  |
|------------|--|
| Format     | <p>Determines the logical record length (LRECL). Select one of the following format:</p> <ul style="list-style-type: none"> <li>Fixed: Each string contains exactly this number of characters.</li> <li>Fixed ASA: Each string contains exactly the number of characters and the use of ASA characters on z/OS.</li> <li>Fixed Block: All blocks and all logical record are fixed in size. One or more logical records reside in each block.</li> <li>Fixed Block ASA: All blocks and all logical record are fixed in size. One or more logical records reside in each block and the use of ASA characters on z/OS.</li> <li>Fixed Block MACHINE: All blocks and all logical record are fixed in size. One or more logical records reside in each block and the use of MACHINE characters on z/OS.</li> <li>Fixed MACHINE: Each string contains exactly the number of characters defined by the string length parameter and the use of MACHINE characters on z/OS.</li> <li>Variable: The length of each string is less than or equal to this number.</li> <li>Variable ASA: The length of each string is less than or equal to this number and the use of ASA characters on z/OS.</li> <li>Variable Block: Blocks, as well as logical record length, can be any size. One or more logical records reside in each block.</li> <li>Variable Block ASA: Blocks, as well as logical record length, can be any size. One or more logical records reside in each block and the use of ASA characters on z/OS.</li> <li>Variable Block MACHINE: Blocks, as well as logical record length, can be any size. One or more logical records reside in each block and the use of MACHINE characters on z/OS.</li> <li>Variable MACHINE: The length of each string is less than or equal to the string length parameter and the use of MACHINE characters on z/OS.</li> <li>Undefined: Blocks are of variable size. There are no logical records. The logical record length is zero. This record format is usually only used in load libraries. Block size must be used if you are specifying Undefined.</li> </ul> |
| Length     | <p>Record length is the maximum number of characters in a string or record of the file. The maximum number is 32760.</p>   |
| Block Size | <p>Specifies the size of the block. For FB the block size must be a multiple of record length, and for VB the record length can be any size up to the block size minus four. The maximum number is 32760.</p>  |



Allocation Tab

The **Allocation** tab contains the following elements:



| Elements  | Description  |
|-----------|--|
| Type      | <div>The valid value can be Tracks, Cylinders, Megabytes, and Kilobytes.</div> <div> The default is Kilobytes with zero Primary and zero Secondary space. This default configuration picks up the size of the file sent to the z/OS system and allocates the appropriate space.</div> |
| Primary   | Used by the z/OS partner when creating datasets as the initial number of units of TRACKS, CYLINDERS, and so on to allocate.  |
| Secondary | Used by the z/OS partner when creating datasets as the next number of units of TRACKS, CYLINDERS, and so on to allocate once the initial space in the dataset has been exhausted.  |

Disk Tab

The **Disk** tab contains the following elements:



| Elements | Description  |
|----------|--|
| Volume   | This is the 1–6 character volume name of the disk drive on which the z/OS data set is to be allocated. |
| Unit     | This is the 1–8 character name of the type of Unit where the host dataset is to be allocated.          |

| Elements     | Description   |
|--------------|---|
| Availability | Indicates when the remote file is available. The two valid values are Immediate (Disk) and Deferred (Tape). |

### Other Tab

The **Other** tab contains the following elements:

| Elements                        | Description   |
|---------------------------------|---|
| Truncate                        | <p>Defines the action to be taken on z/OS when the record length is greater than the LRECL. Valid values are:</p> <ul style="list-style-type: none"> <li>No: The transfer is terminated. This is the default value.</li> <li>Yes: The record is truncated to the record length.</li> <li>Wrap: The record is truncated and the truncated data is written to the next record.</li> </ul> |
| MaintainRDW                     | Defines that the data is an RDW format and the RDW format is maintained by z/OS.  |
| MaintainBDW                     | Defines that the data is an a BDW format and the BDW format will be maintained by z/OS.   |
| Remove Trailing Spaces          | Select this check box to remove all spaces or binary zeros at the end of a record when transferred from the z/OS platform.  |
| RetentionPeriod_Expiration Date | Defines the number of days or provides the yyyy/ddd format. Example 1: 30 Example 2: 2016/264   |

## Schedule Tab

You can schedule a transfer activity under the **Schedule** tab.

The screenshot shows the 'Schedule' tab of a configuration window. It includes checkboxes for 'Schedule Transfer' and 'Hold Permanent Errors'. The 'Scheduled Start' section contains fields for 'Start At' (01/18/2018), 'Time' (03:12:20), and 'Day' (Thursday). The 'Repeat' section has radio buttons for 'Don't Repeat, Execute Once' (selected), 'Indefinitely', 'Number of times', and 'Until'. A 'Next Occurrence' field displays 'Thursday, January 18, 2018 03:12:20'. The bottom of the window features 'OK' and 'Cancel' buttons.

| Elements              | Description  |
|-----------------------|--|
| Schedule Transfer     | Adds (select) or deletes (clear) schedules for the transfer. If a transfer is scheduled, it takes precedence over the Check Point/Restart option under the <b>Transfer</b> tab and the input under the <b>Expiration</b> tab.  |
| Hold Permanent Errors | Puts a scheduled transfer on hold if a permanent error occurs. If this check box is cleared, the transfer continues to be attempted even after a permanent error occurred. Examples of permanent errors can be the remote file not existing, bad user id or password, and expired license key. |

| Elements        | Description   |
|-----------------|---|
| Scheduled Start | <p>Indicates when you want a file transfer to execute. This parameter has three fields.</p> <ul style="list-style-type: none"> <li>Start At: Specifies the date that the transfer is eligible. This defaults to the current date. This entry is mutually exclusive with the <b>Day</b> (day of week) field.</li> <li>Time: Specifies a particular time that the transfer is eligible. This defaults to the current time.</li> <li>Day: Specifies a particular day of the week that the transfer is eligible. This entry is mutually exclusive with the <b>Start At</b> (date) field.</li> </ul>   |
| Repeat          | <p>Provides information relative to the future execution of the particular file transfer after it has been executed once. This parameter has the following radio buttons:</p> <ul style="list-style-type: none"> <li>Don't Repeat, Execute Once: When this option is selected, the file transfer is executed once, and then no longer attempted.</li> <li>Indefinitely: When this option is selected, The transfer is to be executed indefinitely (or until the current user or administrator deletes the job) and in accordance with the information specified in the <b>Start At</b> field and in the <b>Interval</b> field.</li> <li>Number of times: This option specifies the number of times the file transfer can be executed before it is removed from the queue. Valid values between 2 and 32767. Default is 2.</li> <li>Until: This option specifies the date, time and the day of the week until when you want to execute the file transfer.</li> <li>Interval: If you specify a Repeat option (with the exception of Don't Repeat, Execute Once), this option is displayed. There is a drop-down list that provides the following selections: Daily 7 (Sunday to Saturday), Weekly, Bi-Weekly, Monthly, Bi-Monthly, Quarterly, Semi-Annually, Annually, Bi-Annually, and Every.</li> </ul> <p>The panel changes if the option Every is selected. The Interval parameter adds two additional fields that you can use to indicate the frequency with which you want to repeat the transfer. The first field allows you to 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).</p> |



If your scheduled transfer fails during transmission for any reason, the transfer will be executed at the next Scheduled date and time, it will NOT be executed as soon as the problem that caused the failure is resolved.

## Notify Tab

You can set notification to be received at the end of a transaction under the **Notify** tab.

There are three types of notifications: Email Notification, Local Only, and Remote Only. You can specify notification for success or failure by selecting the corresponding check boxes in each type of notifications.

|            |                        |             |                  |
|------------|------------------------|-------------|------------------|
| Expiration | Post Processing Action | Accelerator | TCP/IP           |
| Transfer   | Schedule               | Notify      | Advanced Options |

**Email Notification**

☐ On Success    Email:

☐ On Failure    Email:

**Local Only**

☐ On Success    Email:

☐ On Failure

**Remote Only**

☐ On Success    Email:

☐ On Failure

| Notification Type   | Fields     | Description   |
|---------------------|------------|---|
| Remote Notification | On Success | <p>The emails of the user to notify when a transaction is completed. It notifies the user whether the transaction is successful or not.</p> <p>The check boxes allow you to define whether an email should be sent on Success and/or Failure.</p> <p>If specifying email in this field, ensure that you have completed the <b>SMTP Server</b> field under the <b>General</b> tab in the MFT Platform Server Properties panel.</p> |
|                     | On Failure |   |

| Notification Type  | Fields                   | Description   |
|--------------------|--------------------------|---|
| Local Notification | On Success<br>On Failure | <p>The emails of the user to notify when a transaction is completed. It notifies the user whether the transaction is successful or not.</p> <p>The check boxes allow you to define whether an email should be sent on Success and/or Failure.</p> <p>If specifying email in this field, ensure that you have completed the <b>SMTP Server</b> field under the <b>General</b> tab in the MFT Platform Server Properties panel.</p> |
| Email Notification | On Success<br>On Failure | <p>The emails of the user to notify when a transaction is completed. It notifies the user whether the transaction is successful or not.</p> <p>The check boxes allow you to define whether an email should be sent on Success and/or Failure.</p> <p>Ensure that you have completed the <b>SMTP Server</b> field under the <b>General</b> tab in the MFT Platform Server Properties panel.</p>                                    |

# Advanced Options Tab

You can set some advanced features under the **Advanced Options** tab.

Expiration

Post Processing Action

Accelerator

TCP/IP

Transfer

Schedule

Notify

Advanced Options

Transfer Description

Process Name

Name

User Data

Thread Priority

Level

Normal

Check Point

Interval

5

(min.)

Compression

Type

None

Encryption

Method

None

Custom Code Page Conversion

LocalCTFile

...

RemoteCTFile

UTF8BOM

OK


Cancel

| Elements             |              | Description   |
|----------------------|--------------|---|
| Transfer Description | Process Name | <p>This eight-character field describes the application which is initiating 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.</p> <p>This field can be used for automating transactions from the Host. See Appendix C Automated Operations of the <i>MFT Platform Server for z/OS User's Guide</i>.</p> |

| Elements                   | Description  |
|----------------------------|--|
| User Data                  | <p>Any alpha, numeric or national characters of up to 25 characters that are logged into the history files that descriptive information on the transfer. This field is optional.</p> <p>This field can be used for automating transactions from the Host. See Appendix C Automated Operations of the <i>MFT Platform Server for z/OS User's Guide</i>.</p>   |
| Thread Priority<br>(Level) | <p>Assigns priority to transactions that are executed simultaneously and are competing for resources. This is the priority is assigned when creating the transfer thread. This is not the priority used in the work queue.</p> <p>The levels of priority that can be assigned are as follows: highest, above normal, normal, below normal, lowest, and idle.</p>   |
| Check Point<br>Interval    | <p>Checks 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.</p> <p>The MFT Platform Server for Windows checkpoint uses a time interval to determine when to send a checkpoint. Since Check Point is time-based, the checkpoint always occurs at a regular interval.</p> <p>Check Point Interval is specified in minutes and is a valid range 1 to 90 minutes.</p> |



| Elements    | Description  |
|-------------|--|
| Compression | <p>Compresses data on the sender side of the transfer and decompresses the data on the receiver side of the transfer. This will result in fewer packets being sent between systems, and reduce network traffic. The compression provided by MFT Platform Server for Windows is Smart compression because it removes a level of complexity from the user.</p> <p>When you compress certain types of data, the compressed data is larger than the original data. Smart Compression solves this problem by transmitting only the data packets which are smaller than the original. This saves the increased network bandwidth of the larger compressed packet and saves the CPU cycles on the receiving side.</p> <p>This field provides the following compression algorithms:</p> <ul style="list-style-type: none"> <li>• LZ (Lempel-Zev): Provides better compression ratios and compresses a wider variety of different data types than RLE. Choose LZ if you need better compression ratios and can spare CPU cycles.</li> <li>• RLE (Run Length Encoding): More data-dependent than LZ. That is, the compression ratio varies 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 need to save CPU cycles.</li> <li>• ZLIB1 through ZLIB9: Refers to levels of ZLIB compression. Level 1 is very fast but hardly compresses. Levels 7 to 9 yield the best compression but is much slower. Level 2 (ZLIB2) typically offers the best compromise of compression and speed. We suggest using ZLIB2 unless there is a specific need for higher compression and CPU utilization is not an issue.</li> <li>• None: No compression is used for this transfer.</li> <li>• Default: If Default is chosen, the type of compression is taken from the Node setting or set to None for non-Node transfers.</li> </ul> |

| Elements                          | Description   |
|-----------------------------------|---|
| Encryption                        | <p>Turns encryption on and off. The encryption contains the following methods:</p> <ul style="list-style-type: none"> <li>• DES (56 bit encryption): Data Encryption Standard (DES) 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.</li> <li>• Triple DES (112 bit encryption): Triple DES is just DES done three times with two secret keys applied in a particular order giving you 112 bit encryption.</li> <li>• Blowfish (56 bit encryption): Blowfish is a block encryption algorithm that can use keys from 40 to 448 bits long. The MFT Platform Server implementation of Blowfish uses a 56 bit encryption key.</li> <li>• Blowfish Long (448 bit encryption): This Blowfish block encryption algorithm uses a key 448 bits long (AKA. Blowfish Long encryption). It is very fast, about six times faster than DES, and about fifteen times fast than 3DES.</li> <li>• 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. When encrypting data, we suggest using this encryption algorithm.</li> <li>• None: No encryption used for this transfer.</li> <li>• Default: The type of encryption is taken from a Node that had been configured or it will be set to None for non-Node transfers.</li> </ul> |
| Custom Code<br>Page<br>Conversion | <div>LocalCTFile</div> <div>Translates on the local side. This parameter need to contain the name of the file.</div>  |
|                                   | <div>RemoteCTFile</div> <div>Translates on the remote side. This parameter need to contain the name of the file.</div> <div>  <p>When defining the RemoteCTFile, you need to set the LocalCTFile to Null in order that no translation takes place locally.</p> </div>  |
|                                   | <div>UTF8BOM</div> <div>Defines whether the UTF8BOM is added or removed by z/OS. Valid Options are:</div> <ul style="list-style-type: none"> <li>• None - no UTF8BOM processing takes place</li> <li>• Add - The UTF8BOM is added by z/OS</li> <li>• Remove - The UTF8BOM is removed by z/OS</li> <li>• Both - The UTF8BOM is added and removed by z/OS</li> </ul>  |


## Expiration Tab

You can set expirations for transfer under this tab.

The screenshot shows a dialog box with a tabbed interface. The 'Expiration' tab is selected. The dialog has a title bar and a standard Windows-style border. The tabs are: Transfer, Schedule, Notify, Advanced Options, Expiration, Post Processing Action, RocketStream, and TCP/IP. The 'Expiration' tab contains the following fields:

- Expiration Date:** A group box containing three fields: 'At' with a date picker showing '04/09/2009', 'Time' with a time picker showing '15:10:00', and 'Day' with a dropdown menu showing 'Thursday'.
- Retention Period:** A group box containing a 'Retention' field with a spinner box set to '0' and the text '(days)'.
- Attempt Transfer:** A group box containing a 'Try Count' field with a spinner box set to '1'.
- Timeout:** A group box containing a 'Timeout' field with a spinner box set to '120' and the text '(min)'.

At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

| Elements                      | Description  |
|-------------------------------|--|
| Expiration Date               | <p>Specifies the exact date and time when a transfer is expired. However, if this transfer was scheduled, that will take precedence over expiration. If Expiration and Retention are used, whichever value occurs first takes precedence.</p> <p>In the first field, specify the date on which you want the transfer to expire. In the second field, specify the time at which you want the transfer to expire. In the third field, indicate the day of the week on which you want the transfer to expire.</p> <ul style="list-style-type: none"> <li>• At: 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 <b>Day</b> (day of week) field.</li> <li>• Time: Specifies a particular time at which you want the transfer to expire. This defaults to the current time.</li> <li>• Day: Specifies a particular day of the week on which you want the transfer to expire. This entry is mutually exclusive with the start <b>At</b> (date) field.</li> </ul> |
| Retention Period              | <p>Specifies the number of days that should pass from the start of the transfer to the point it is expired. If Expiration and Retention are used, whichever value occurs first takes precedence.</p>   |
| Attempt Transfer<br>Try Count | <p>Specifies the number of times that MFT Platform Server for Windows attempts the transfer. When the Try Count is reached, 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. Max number is 9998. Number 0 represents Unlimited feature, which is actually 9999 tries.</p>   |
| Timeout                       | <p>Specifies the amount of time (minutes) a connection stays open while waiting for a response from the remote side. Once the time is reached the connection is ended.</p> <div>  <p>This parameter takes precedence over the Initiator Timeout on the Server Properties window. See Timeout: Initiator parameter in the General tab of <a href="#">Server Properties</a>.</p> </div>   |

## Post Processing Action Tab

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

The screenshot shows the 'Post Processing Action' configuration window. It features a tabbed interface with the following tabs: Transfer, Schedule, Notify, Advanced Options, Expiration, Post Processing Action (selected), RocketStream, and TCP/IP. The 'Post Processing Action' tab contains four identical sections, each labeled 'Post Action 1' through 'Post Action 4'. Each section includes three dropdown menus and a 'Data' text field. The first dropdown menu in each section is currently set to 'Off'. At the bottom of the window are three buttons: 'OK', 'Cancel', and 'Help'.

| Elements    |         | Description  |
|-------------|---------|--|
| Post Action | Field 1 | The values for this field are <i>Off</i> , <i>Success</i> , or <i>Failure</i> . This Post Action is to be executed based on the completion status of the transfer. |
|             | Field 2 | The values for this field are <i>Initiator</i> or <i>Responder</i> . This Post Action is to be executed base on the source of the file transfer.                   |
|             | Field 3 | The values for this field are <i>Command</i> , <i>Call Program</i> , <i>Call JCL</i> , and <i>Submit</i> . This is the type of the action to be executed.          |

| Elements | Description   |
|----------|---|
| Data     | <p>Defines the file to be executed.</p> <p>Append a # sign to the end of the data entered to have MFT Platform Server for Windows launch the PPA and have it wait for the return code of the action.</p> <p>Append a &amp; sign to the end of the data entered to have MFT Platform Server for Windows launch the PPA and not wait for the action to finish. The default behavior is the same as appending a &amp; sign to the data entered.</p> <p>For example,</p> <pre>C:\MyAction1.exe arg1=true #</pre> <pre>C:\MyAction2.exe arg1=false &amp;</pre> |

## Substitutable Parameters

MFT Platform Server supports Substitutable Parameters to allow you to take full advantage of the 256 character maximum on the command data. When using substitutable parameters, you do not have to copy the filename from the LocalFileName or RemoteFileName parameters.

MFT Platform Server does not support file name tokens within PPA, 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 character (%) as the escape character instead of the \$ that tokens use.

The following is a list of the substitutable parameters that are supported. In the examples of the list, assume that there is a file named C:\a\b\c\d\config.txt.

| Substitutable Parameter | Description  | Resolved Name Example |
|-------------------------|--|-----------------------|
| %DIR                    | Remote File Name directory without the 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 lowest directory.                   | a\b\c                 |
| %NOHDIR                 | Directory name w/o high level directory.                   | b\c\d                 |
| %FILE                   | The file name without the 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            |

| Substitutable Parameter | Description                            | Resolved Name Example |
|-------------------------|--|-----------------------|
| %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 used, the substitutable data is the text in the name of the parameter without the % sign. If UNIX detects the %DRIVE parameter, the value DRIVE should be used as substitution. Similarly, %PROC becomes PROC and %UDATA becomes UDATA if not interacting with a z/OS system.

## Accelerator Tab

If you are licensed to use the TIBCO Accelerator technology and you want to set transfer requests to be sent using the TIBCO Accelerator protocols of User Datagram Protocol (UDP), PDP, or TCP, you can enable it by selecting the **Accelerate** check box. You can see the properties panel is enabled to configure the TIBCO Accelerator host and port your transfer request to be sent to.

|            |                        |             |                  |
|------------|------------------------|-------------|------------------|
| Transfer   | Schedule               | Notify      | Advanced Options |
| Expiration | Post Processing Action | Accelerator | TCP/IP           |

☒ Accelerate

Properties

Host:

Port:

MaxSpeed (kbps):

Protocol

☐ TCP  
☐ UDP  
☒ PDP

Options

☐ Encryption

Compression

OK Cancel

The TIBCO Accelerator host can be 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 should be set to use.



- The standard TIBCO Accelerator port to use is 9099. It is not recommended to use another port unless instructed by your local administrator.
- It is not recommended to use MFT Platform Server Compression with the TIBCO Accelerator compression. One or the other should be used.

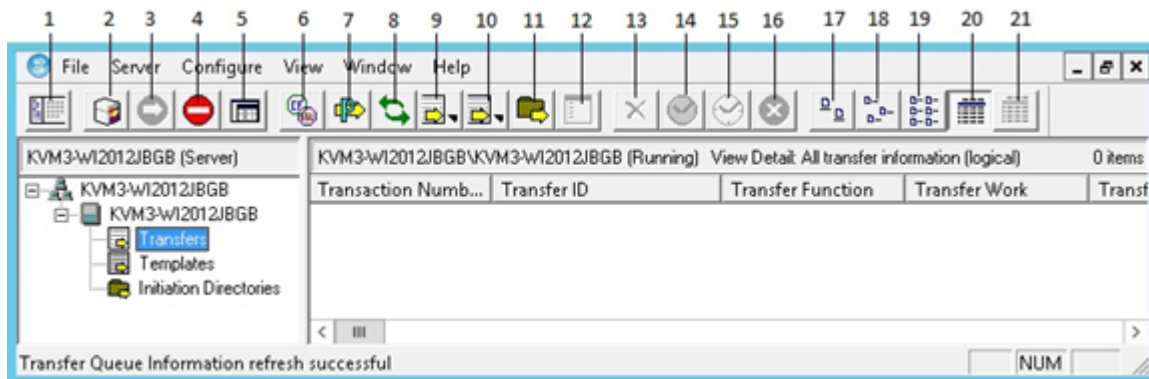


## TCP/IP Tab

The **TCP/IP** tab is displayed in TCP transfer. You can set the network information under this tab.

| Elements        | Description   |
|-----------------|---|
| Port Number     | This is the secondary network address for the TCP/IP transfer. 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, MFT Platform Server for Windows uses 46464. If other applications on your network use this port number, use a different port for your MFT Platform transfers.  |
| Secure Protocol | <p>Select the protocol used for this transfer. Valid options are:</p> <ul style="list-style-type: none"> <li>• Plain - TLS/SSL is not used</li> <li>• SSL - TLS is used to validate the client and server and to pass an encrypted encryption key in the TLS Session. Then the TLS Session is terminated and data is encrypted using the encryption key passed in the TLS Session. A message digest and sequence number is added to each packet and is verified by the Responder.</li> <li>• Tunnel - All data is sent through an encrypted TLS Tunnel. We suggest using Tunnel when sending data over a public network.</li> </ul> |

# 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 (transfer s, template s, 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.


## Buttons

MFT Platform Server Administrator provides a toolbar to perform tasks for convenience. You can use the menus to perform same tasks as the buttons in the toolbar. By default, the toolbar is displayed in the MFT


Platform Server Administrator window. If not, you can open it by clicking **View > Toolbars > Server Network** from the menu.

The toolbar contains the following buttons:


#### Create a New Network View

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


#### Add a New Server to Known Server List

Click  to add a server to the Network window.


#### Start Server

Click  to start a server.

#### Shutdown Server


Click  to stop a server.

#### View and Change Server Properties


Click  to open the MFT Platform Server Properties panel, which contains 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 appears as Read Only).


#### Configure SSL

Click  to open the MFT Platform Server SSL Settings panel. You can modify the current SSL settings on this panel.


#### Configure Post Processing

Click  to open the Post Processing panel. You can select the **Use Configured Post Processing** check box to enable the Configure Post Processing feature, and then specify the name of the file to be used for the post processing.

#### Refresh View Information

Click  to view the current server and transfer information.

#### Create a New Transfer

Click  to add a new transfer to the queue of the server you are working on. After selecting a transfer type, the Transfer Properties panel is displayed. You can specify all of the particulars of the file transfer that you want to add to the queue.


## Create a New Transfer Template

Click  to create a new transfer template.

## Create a New Initiation Directory

Click  to create a new Directory Named Initiation entry.


## View or Change Properties

Click  to view or change the parameters of a specific Transfer, Template, or Directory Named Initiation entry. The Properties panel is displayed. You can modify the properties on the Properties panel.

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

Any significant changes made to the MFT Platform Server queue view are logged to the event log.


## Delete Selected Objects

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

## Hold

Click  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.

## Release

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

## Abort


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

Select either of two options:


- Cancel transfer: Notifies the remote system that the transfer has been terminated.
- Terminate transfer immediately: Terminates the transfer and not notify the partner. In certain instances, this selection can stop a transfer that Cancel transfer cannot.

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.


## View Items as Large Icons

Click  to display a single large icon for each file transfer with its Transfer ID directly below it. The appearance of the icons depends on the file transfer type selected.


### View Items as Small Icons

Click  to display a single small icon for each file transfer with its Transfer ID directly next to it. The appearance of the icons depends on the file transfer type selected.


### View Items in a List

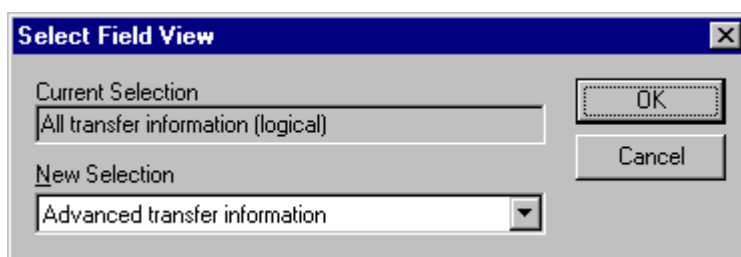
Click  to display a single small icon for each file transfer with its Transfer ID directly next to it. The way the icon appears differs depending upon the file transfer type selected.

### View Items in Detail

Click  to view detailed information about the transfers in the Queue view. The fields varies depending on your selection in the Select Field View panel. By default, all the fields in the queue are displayed.

### Select Field View

Click  to select which fields you want to view from a predefined group. The Select Field View panel is displayed.



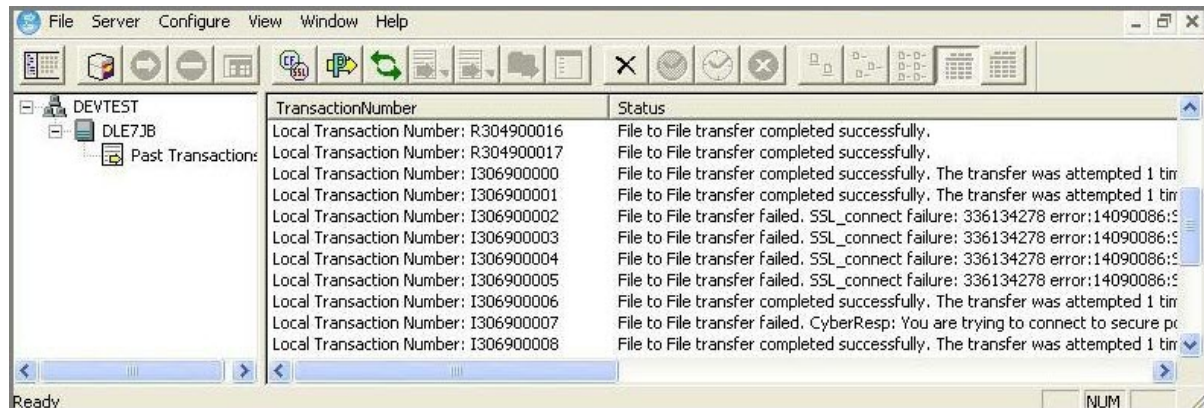
- Current Selection:  
Based on your selection, different information is displayed about current transfers.

## Past Transactions

You can use the Past Transactions feature of MFT Platform Server Administrator to view the status of transfers, which are completed.

Click **View > Past Transactions** to view the status of completed file transfers. You need to add a specific server to view status of the previously completed transfers on that server. The addition is exactly the same as adding a server in a network view in the Administrator.

You cannot double-click a transfer to view status information about the transfer. The status information about file transfers is pulled from the event logs of the respective server. Therefore, if you clear event logs, the past transactions are also deleted.

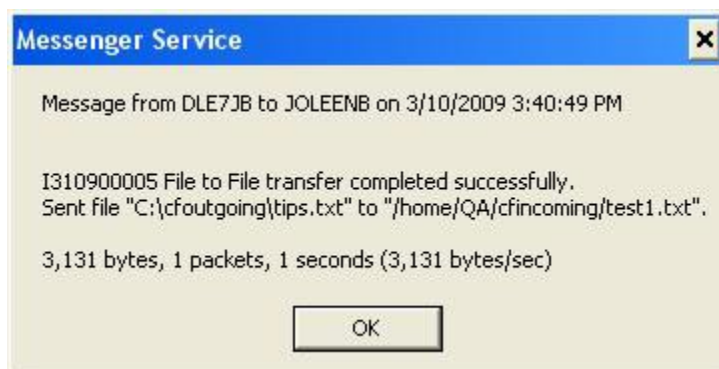


A backup event log on the server is created every time you open or refresh the Past Transactions dialog. The backup is in the `c:\temp\tmp.evt` file. You can click **File > Open Backup Eventlog** to read data from the backup event log in the Administrator. If you delete the `c:\temp\tmp.evt` file, no transactions are available when you open the backup event log. You can sort the transactions by clicking any column header.



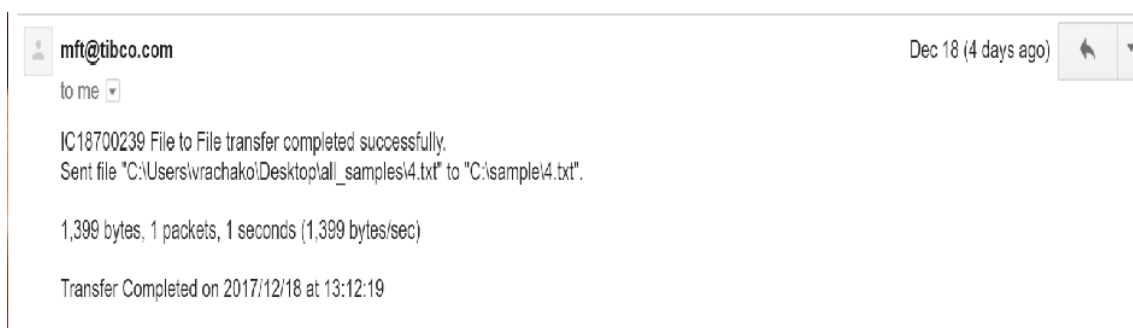
The **Open Backup Eventlog** menu item is available only after you select a specific server.

## Notification




### MFT Platform Server Email Notification

Upon completion of a file transfer, MFT Platform Server sends an email to the address that is specified in the **Notify** tab. The following is an example email.



# Server Properties


To open the Server Properties panel, you can choose either of the following two ways:

- Click  on the toolbar.
- Right-click the server name in the left panel, and then click **Properties**.

## General Tab

| Accelerator   |           | Service Control Manager   |       |
|---|-----------|---|-------|
| General   | Responder | Throttle  | Trace |
| <p>Master Domain <input type="text"/></p> <p>Dispatcher Cycle <input type="text" value="1 Minute"/></p> <p>Restart Type <input type="text" value="Warm"/></p> <p>SMTP Server <input type="text" value="smtp.tibco.com"/></p> <p>Sent From <input type="text" value="mft@tibco.com"/></p>  |           |   |       |
| <p>Responder</p> <p>Timeout <input type="text" value="120"/> (min)</p> <p><input checked="" type="checkbox"/> Required Node Definition</p>  |           | <p>Initiator</p> <p>Timeout <input type="text" value="120"/> (min)</p> <p><input type="checkbox"/> Required Node Definition</p> |       |
| <p>System Configuration</p> <p>EOF Options <input type="text" value="No Processing"/></p> <p>Security Policy <input type="text" value="None"/></p> <p>Log Directory Transfers <input type="text" value="Yes"/></p> <p><input type="checkbox"/> Run PPA at end of directory transfer <input checked="" type="checkbox"/> CRC</p> |           |   |       |
|   |           | <p>OK Cancel</p>  |       |



| Elements                           | Description  |
|------------------------------------|--|
| Master Domain                      | Specifies the name of the domain to be the default domain for verifying security rights when your server is acting as a responder. This means all a remote user has to define in the transfer information for the remote identification is the user id without a domain name preceding it.   |
| Dispatcher Cycle                   | Specifies the time that the scheduled dispatcher service waits before it next checks for transfers that need to be started or restarted. The selectable values in this field are 10 seconds, 30 seconds, 1–10 minutes, 15 minutes, 30 minutes, 45 minutes, 1 hour, 2 hours, 4 hours, 8 hours, 12 hours, 24 hours. The scheduled dispatcher service writes the date and time to the MFT Platform Server Monitor when it checks schedules for eligibility.   |
| Restart Type                       | <p>Specifies the type of restart.</p> <ul style="list-style-type: none"> <li>• Warm: All of the transfers that are in the persistent work queue are retained when MFT Platform Server is restarted.</li> <li>• Cold: All of the transfers that exist in the persistent work queue are not retained when MFT Platform Server restarts. The old PQF is overwritten by a new PQF.</li> </ul> <p> A Cold Start deletes your DNI definitions as well as any queued or active transfers</p> |
| SMTP Server                        | The name of the email server used to send out email notification. If you change the value in this field, then you should also stop and start the MFT Platform Server service in order for the new value to be picked up.   |
| Sent From                          | Identifies the name displayed in the email notification. This value cannot contain any spaces.   |
| Timeout: Responder                 | Specifies the amount of time (minutes) a connection stay open while waiting for a response from either the Initiator or the Responder. Once the time is reached, the connection is ended. The value can be from 1 to 1440. The default value is 120 (2 hours).   |
| Timeout: Initiator                 | Specifies the amount of time (minutes) a connection stays open while waiting for a response from either the Initiator or the Responder. Once the time is reached, the connection is ended. The value can be from 1 to 1440. The default is 120 (2 hours).  |
| System Configurations: EOF Options | Defines what permitted action (whether a Carriage Return Line Feed (CRLF), End of File (EOF), or both of them need to be added to records) takes place for transfers that have defined CRLF=YES. If a user has defined a CRLF=NO or has defined a permitted action along with CRLF=YES in the transfer, this global setting will be ignored.   |

| Elements   | Description   |
|--|---|
| System Configurations: Security Policy   | <p>Defines whether to enforce HIPAA or FIPS-140 regulations on initiated and responding transfers.</p> <ul style="list-style-type: none"> <li>• HIPAA: Requires MFT Platform Server to comply with HIPAA standards. At this time the standards require that all files are transferred using encryption key that is 128 bits or greater.</li> <li>• FIPS-140: Requires MFT 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 the files are transferred using SSL with an encryption type of Rijndael (AES) which uses a key length of 256 bits. For more information on configuring SSL, see the <a href="#">SSL Configuration</a> section.</li> </ul> |
| Log Directory Transfers  | <p>This parameter defines whether to log cfdir requests when doing directory transfers.</p> <p>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.</p>  |
| Run PPA at end of directory transfers (Directory Transfer or Distribution List Transfer) | <p>Defines when a directory transfer is complete and \or a Distribution List is used and Post Processing Action(s) are configured that 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.</p>   |
| CRC  | <p>Defines global CRC computing when enabled. This global setting can be overwritten by CRC parameter defined in <code>cfnode.cfg</code> file.</p> <p>If transfer is done via node, then node setting will take precedence. Valid values for CRC node option is Yes, No, or Default (meaning Global ).</p> <p>If transfer is done without using a node, then CRC checking can be selected in Transfer Properties Sheet.</p>   |

## Responder Tab

You can configure the responder under this tab.

Accelerator

Service Control Manager

General

Responder

Throttle

Trace

Static

TCP/IP

Responder, Port Numbers

IPv4: 46464

TLS IPv4: 56565

Tunnel IPv4: 56567

IPv6: 47474

TLS IPv6: 57575

Tunnel IPv6: 57577

Responder, Listen Adapter IP Addresses

IPv4:

IPv6:

Initiator, Connect Adapter IP Addresses

IPv4:

IPv6:

Default Class of Service

Nodes, ResponderProfile

Default

No

Access Control Config File

...

CFAlias Config File

...

OK

Cancel

| Elements                  |      | Description  |
|---------------------------|------|--|
| TCP/IP Transfer Responder | IPv4 | MFT Platform Server for Windows responds to transfers 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 reside at the same IP address on the same machine, but makes them unique so they can co-exist.  |
|                           |      | The default IP port number for MFT Platform Server is 46464, but you can change it to any number between 5000 and 65535, inclusive. However, some lower port numbers can be reserved for standard applications at your installation. |
|                           |      | Select the <b>Disable</b> check box to turn the regular TCP/IP port number off.  |

| Elements  | Description   |
|---|---|
| IPv4  | <p>The port number on which SSL is listening. The default for the SSL IP port number for MFT Platform Server is 56565, but you can change it to any number between 5000 and 65535, inclusive. However, some lower port numbers can be reserved for standard applications at your installation.</p>  |
|   | <p><b>Responder Listen Adapter IP Addresses</b></p> <p>If a machine has more than one IP address, you can bind the connection to a particular one. The default value for this parameter is ALL, which means binding to any IP Address. If this parameter is defined, the Responder accepts incoming requests from only this IP address. You can specify Adapter IP Addresses for IPV4 and IPV6 for incoming requests.</p> <p><b>Initiator Listen Adapter IP Addresses</b></p> <p>If a machine has more than one IP address, you can bind the connection to a particular one. All initiator connections go through this particular IP Address. The default value for this parameter is ALL, which means binding to any IP Address. You can specify Adapter IP Addresses for IPV4 and IPV6 for incoming requests.</p> |
| Default Class of Service:<br>Allows you to select a Class of Service from the drop down box.<br>Nodes | <p><b>Responder Profile</b></p> <p>Defines a local username and password used in place of the incoming username and password. By using responder profiles, a remote MFT Platform Server installation does not have to know an actual username and password on your local machine to initiate a transfer.</p>  |
| Access Control  | <p><b>Configuration</b></p> <p>Sends a file to the Windows platform and it automatically goes to a pre-defined directory based on user-defined criteria. The default file name for the Access Control configuration is <code>AccessControl.cfg</code>. Please refer to the section on Access Control for more information. This is used by the MFT Platform Server Responder only.</p>  |
| CFAlias   | <p><b>Configuration</b></p> <p>Sends a file to the Windows platform and it automatically goes to a pre-defined directory based on user-defined criteria. The default file name for the CFAlias configuration is <code>CfAlias.cfg</code>. Please refer to the section on CfAlias for more information. This is used by the MFT Platform Server Responder only.</p>  |

# Throttle

You can set server throttling to limit total active transfers, initiators and responders.

Accelerator

Service Control Manager

General

Responder

Throttle

Trace

Transfers Limits

☒ Total Active

100

☒ Initiators

50

☒ Responders

50

OK

Cancel

| Elements     | Description  |
|--------------|--|
| Total Active | Indicates how many active transfers are allowed at any given time. |
| Initiators   | Indicates how many Initiators only are allowed as any given time.  |
| Responders   | Indicates how many Responders only are allowed as any given time.  |

# Trace Tab

You can configure the tracing for the MFT Platform Server under this tab.

The **Trace** tab contains the following three tabs:

- Server** tab: Allows you to configure the tracing for the activities of the server including actions related to performing file transfers, and managing Transfer, DNI, and Template objects.

- **Communications** tab: Allows you to configure the tracing for the server, specific 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.
- **Log File** tab: Allows you to view past transactions through the MFT Command Center.

### Server or Communications Tab

Accelerator | Service Control Manager

General | Responder | Throttle | **Trace**

Server | **Communications** | Log File

Files

1: C:\PSW\_800\MFT Platform Server\Trace\FtmsSvr1.trc

2: C:\PSW\_800\MFT Platform Server\Trace\FtmsSvr2.trc

Trace Level: No Tracing

Flip Length: 1048575 (bytes)

☒ Truncate Files when Opened

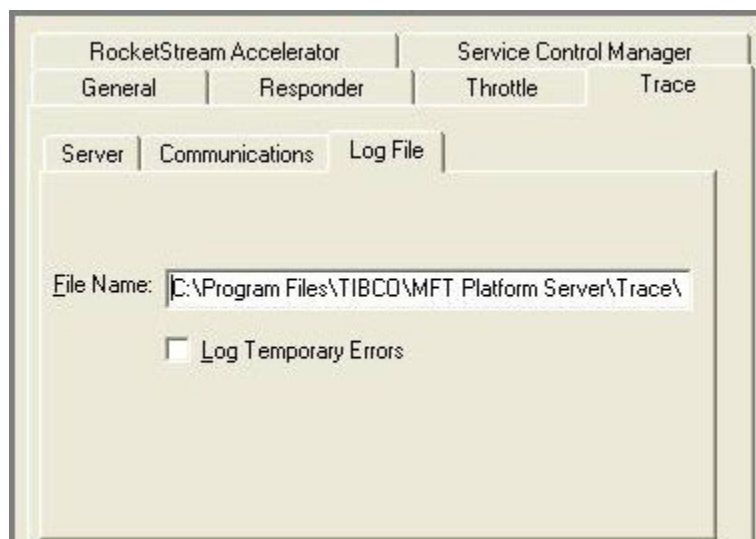
Trace Priority: Normal

OK Cancel

| Elements    | Description  |
|-------------|--|
| File 1      | Indicates which file to use for the first flip file.   |
| File 2      | Indicates which file to use for the second flip file.  |
| Trace Level | 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. Tracing should only be used to troubleshoot a problem and Diagnostic Level 3 should only be turned on at the request of TIBCO Technical Support. |
| Flip Length | The maximum amount of information (in bytes) that is written before the trace files flip. This value should not be less than 1024.   |

| Elements                   | Description   |
|----------------------------|---|
| 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, it opens the existing files and appends the information to the end.   |
| Trace Priority             | <p>Indicates the priority given to the thread that is responsible for receiving and formatting the trace information from the system. Increase this value if it appears that the system is generating trace information that exceeds the ability of system to write the information to the trace files. Tracing should only be turned on at the request of TIBCO Technical Support.</p> <p>While the fields described above apply separately to each trace file, this field applies to all of the trace files at the same time.</p> |

### Server or Communications Tab



| Elements                  | Description   |
|---------------------------|---|
| File Name                 | The path name for the Log file to which the information is written. This file is accessed when inquiring on transactions using the <b>cfinq</b> utility as well as by MFT Command Center. |
| Log All Transfer Attempts | Select this check box to set Log All Transfer Attempts to on. Setting this to off causes MFT Platform Server to log only the final transfer attempt in a restart situation.               |

## Accelerator

You can maintain the configuration of the TIBCO Accelerator service (RsTunnel.exe) under this tab. This panel allows you to stop and start the Accelerator service from this location. If you edit the TIBCO Accelerator Host or Port, you must restart the Accelerator service for the new settings to be taken.

General | Responder | Throttle | Trace

Accelerator | Service Control Manager

Accelerate

Accelerator Host: LOCALHOST

Accelerator Port: 9099

Local Server Status

Start | Stop | Status

Accelerator is running.

OK | Cancel

| Elements                      | Description  |
|-------------------------------|--|
| RocketStream Accelerator Host | The Hostname or IP of the Accelerator Host.  |
| RocketStream Accelerator Port | The port number the TIBCO Accelerator is listening on. The default port is 9099.             |
| Local Server Status           | Starts and stops the RSTunnel Service as well as displays the current status of the service. |



## Service Control Manager

You can maintain the configuration of the MFT Platform Server for Windows service in the Windows Service Control Manager. Since the MFT Platform Server for Windows operates as a Windows Server (on Windows), this tab allows maintenance of both types of service.

General | Responder | Throttle | Trace

Accelerator | Service Control Manager

Service

Display NameTIBCO MFT Platform Server

Image PathC:\PSW\_800\MFT Platform Server\System\ftmssvr.

Logon As

☒ System Account

☐ This AccountLocalSystem

Password\*\*\*\*\*

Confirm\*\*\*\*\*

Start Type

☒ Automatic

☐ Manual

☐ Disabled

OK

Cancel

| Elements     | Description  |
|--------------|--|
| Display Name | The service shown in the Windows Service Control utilities. If not given, tools will display MFT Platform Server as the service description. |
| Image Path   | The full name to the executable for the service. For MFT Platform Servers, this shows ... \ftmssvr.exe.                                      |
| Logon As     | The user ID (local system or specified user) and password that Windows uses to start the MFT Platform Server Service.                        |

| Elements   | Description  |
|------------|--|
| Start Type | <p>Indicates how the service is started.</p> <ul style="list-style-type: none"> <li>Automatic: Starts when the system reboots (recommended setting)</li> <li>Manual: Starts when the administrator tells it to start.</li> <li>Disabled: prevents the service from ever starting.</li> </ul> |

## View - Options

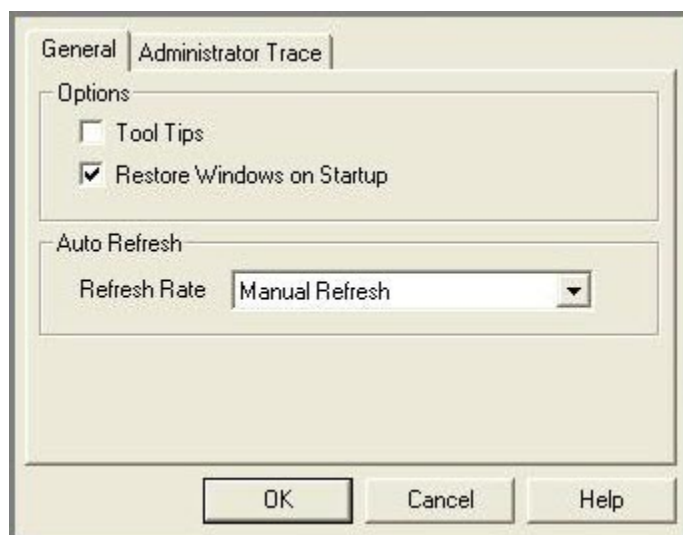
You can view the Options dialog using different ways.

To open the Options dialog, you can use either of the following ways:

- Click **View > Options**.
- Right-click any place in the gray window space, and then click **Options**.

## Options - General Tab

You can configure general properties in the **General** tab.



| Element                    | Description   |
|----------------------------|---|
| Tool Tips                  | Select the <b>Tool Tips</b> check box to view tool tips when you start MFT Platform Server for Windows.   |
| Restore Windows on Startup | Select the <b>Restore Windows on Startup</b> check box to restore Windows settings when you start MFT Platform Server for Windows. This check box is selected by default. |


| Element      | Description   |
|--------------|---|
| Refresh Rate | <p>The administrator has the ability to automatically refresh the information it displays.</p> <p>The <b>Refresh Rate</b> field indicates how often the refresh should occur. The available options are:</p> <ul style="list-style-type: none"> <li>• Manual Refresh: You must select the <b>Refresh</b> command to update the view.</li> <li>• 5 Seconds: The refresh occurs every 5 seconds.</li> <li>• 10 Seconds: The refresh occurs every 10 seconds.</li> <li>• 20 Seconds: The refresh occurs every 20 seconds.</li> <li>• 30 Seconds: The refresh occurs every 30 seconds.</li> <li>• 60 Seconds: The refresh occurs every 60 seconds.</li> <li>• 2 Minutes: The refresh occurs every 2 minutes.</li> <li>• 5 Minutes: The refresh occurs every 5 minutes.</li> <li>• 10 Minutes: The refresh occurs every 10 minutes.</li> <li>• 30 Minutes: The refresh occurs every 30 minutes.</li> <li>• 60 Minutes: The refresh occurs every 60 minutes.</li> </ul> |

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

## Options - Administrator Trace Tab

You can configure tracing properties in the **Administrator Trace** tab.

| Element      | Description   |
|--------------|---|
| Trace File 1 | Specify the file to be used for the first flip file.  |
| Trace File 2 | Specify the file to be used for the second flip file. |

| Element                    | Description   |
|----------------------------|---|
| Trace Level                | <p>A trace level indicates the amount of information that is reported to a trace file. The value is directly proportional to the amount of information written to the trace files.</p> <div>  <p>Tracing can only be used to troubleshoot a problem, and Diagnostic Level 3 can only be used at the request of TIBCO Technical Support.</p> </div> |
| Flip Length                | Specify the maximum amount of information (in bytes) to be written before the trace files flip. The value of the <b>Flip Length</b> field 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 the <b>Truncate Files when Opened</b> is selected, 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             | The <b>Trace Priority</b> field indicates the priority given to the thread responsible for receiving and formatting the trace information from the system. Increase the value if the system generates trace information that exceeds the system's ability to write the information to the trace files.  |



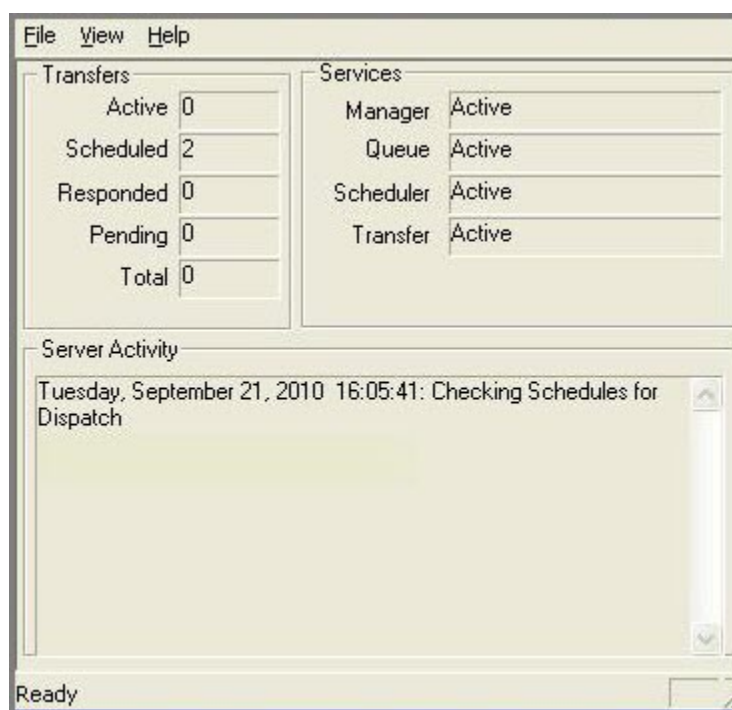
The **Administrator Trace** tab is used to configure the MFT Platform Server Administrator locally. Tracing can only be used at the request of TIBCO Technical Support.

# MFT Platform Server Monitor

You can use the MFT Platform Server Monitor to view all the activities that the MFT Platform Server performs on the server on which it is running.

## MFT Platform Server Monitor Overview

You cannot enter any information or change any values in the MFT Platform Server Monitor window.



| Element         | Description   |
|-----------------|---|
| Transfers       | The <b>Transfers</b> area displays the number of transfers that are present in a particular server's queue. |
| Services        | The <b>Services</b> area displays the status of each service available on a selected server.                |
| Server Activity | The <b>Server Activity</b> area displays all of the actions that the selected server performs.              |

## Functions

From the **View** menu of the MFT Platform Server Monitor, you can click any menu item to perform different functions.

| Function        | Description   |
|-----------------|---|
| View Status Bar | To show or hide the system status bar at the bottom of the window. Hiding the status bar provides more desktop area for viewing information in the MFT Platform Server Activity window. |

| Function            | Description   |
|---------------------|---|
| Always On Top       | To indicate that the window is always on the 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 | To direct 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 show the window, double-click the <b>Monitor</b> icon on the system tray. |
| Clear Display       | To clear information from the MFT Platform Server Activity window.  |

# Command Line Interface

The command line interface allows you to produce clear and readable batch programs using parameters created for all of the MFT Platform Server functions.

To write clear batch programs, long descriptive parameter names are needed. However, interactive command typing needs to be brief. Therefore, several methods for specifying parameters to the command line are supported.

Any given parameter can be specified using:

- environment variables
- short (1 or 2 characters) command line parameters
- long command line parameters



The environment variable settings stay active until you change it or remove it using the **SET** command with no value specified.

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

## Command Line Format

you can provide command lines in different formats.

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

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

The following example shows the format of a simple transfer from the command line that does use environmental variables.

```
SET NODE=nodename
SET CR_LF=no
SET REMOTE_USER_ID=userid
FTMSCMD /SEND /FILE [parameters] "c:\local\file\name.txt" "remote.file.name"
```

In the previous example, mandatory parameters are specified in the environmental variables. You do not need to specify parameters in the parameter section. However, you can still specify any of the additional parameters in the parameter section or in the environmental variables.

The environment variable setting stays available until you change or remove it using the set command with no value specified (for example: **SET CR\_LF=**).

## Specifying Command Line Parameters

To set a command line argument, use the following syntax: **FTMSCMD [parameters] "local\_file\_name" "remote\_file\_name"**

Options can include any number of the following forms:

- Options are indicated by a forward slash (/) or a hyphen (-) followed by the option. Some options need arguments, while some do not. A forward slash is provided for the DOS standard. A hyphen is provided for the UNIX standard.

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

- When an option requires an argument, the argument is separated from the option name by a colon (:) or an equal sign (=), as the following example illustrates.

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

Typing `FTMSCMD /?` provides a list of all arguments.

## File to File Transfers

To send or receive a file, you must specify several parameters on the command line.

Required parameters are associated with the following aspects:

- The transfer's direction.
- The action that should be performed at the destination (written to a file, sent to printer or executed as a job).
- The local file name.
- The remote file name.

### Transfer Direction Parameters

| Parameter | Description  |
|-----------|--|
| Send      | The file is sent from the local side to a remote system.   |
| Receive   | The file is received from a remote system.   |
| Submit    | <p>This parameter is used with the <b>FS:ServerName</b> parameter to submit a transfer to another MFT Platform Server. Specify transfer parameters as you normally do on a command line.</p> <p>If the <b>Submit</b> parameter is specified and a server name is not specified (<b>/fs:ServerName</b>), an error occurs.</p> |

### Action Parameters

| Parameter      | Description   |
|----------------|---|
| File           | To store the contents of the file transfer in a file. This is the default action.   |
| Print          | To send the file being transferred directly to the print queue or spool on the remote side.   |
| Job            | To send a local file to a remote system where the partner executes it as a batch job.   |
| Remote Command | To execute a command on a remote system. The output is stored in a local file that you specify. If the remote system is z/OS, the output is not returned. |





When you receive a file to be executed as a job on a Windows system, the job is executed in the \Windows \SYSTEM32 directory. You need to change the directory in which the batch job executes when writing your batch jobs.

### File Name Parameters

| Parameter        | Description  |
|------------------|--|
| LOCAL_FILE_NAME  | The name of the file on the local system that is to be involved in a transfer.   |
| REMOTE_FILE_NAME | The remote file name of the virtual file stored on the remote system that is the subject of the activity. The parameter value can be any combination of up to 255 characters. If the name contains embedded spaces or commas, specify the name in single quotes. If the remote system is z/OS, only the first 54 characters are significant. |

The following is an example of sending a file to a remote system:

```
FTMSCMD /S /F /NODE:Node1 /DT=BINARY /RL=1 /RI=USERID /RW=pswd
"F:\JOHN\QA\ONEX1.BIN" "JTPLM.QAL.BATCHB.ONEX1"
```

The following is an example of receiving a file from a remote system:

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

## File to Job Transfers

The file to job transfer describes how to transfer a file and have the output of the transfer executed as a job.

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

```
FTMSCMD [parameter] /SEND /JOB [/other parameters] file_name
```

The transfer can be in either direction (receiving a file from the remote side and having it executed on the local side, or sending a file to and having it executed on the remote side). The specific file name depends on the way in which the transfer occurs. For example, to receive a file from the remote side and have it executed on the local system, use *file\_name* to specify the name of the remote file. You do not need to specify a local file name since the output is not written to any local file.

To send a file to the remote side and have it executed on the remote system, use *file\_name* to specify the name of the local file.

The following is an example of sending a job to a remote system:

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

The following is an example of receiving a job from a remote system:

```
FTMSCMD /R /JOB /NODE:Node1 /DT=A /CR=YES /RI=USERID HLQ.TEST.JOB
```



The destination (DS or LU) must be set when doing a transfer.

## File to Print Transfers

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

This is done in the same way that the file's positional parameters are specified when you perform a file to file transfer.

```
FTMSCMD [parameters] /SEND /PRINT  
/REMOTE_PRINTER_NAME=prntername file_name
```

The example illustrates a file transfer whose output is directed to a printer. 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 on the remote side). The specific file name depends on the way in which the transfer occurs. For example, to receive a file from the remote side and print it to a local printer, use *file\_name* to specify the name of the remote file. You do not need to specify a local file name since the output is not written to any local file.

To send a file to the remote side and print it to a remote printer, use *file\_name* to specify the name of the local file.

```
SET REMOTE_PRINTER=prntername  
FTMSCMD [parameters] file_name
```

In the example, mandatory parameters are specified in environmental variables. You do not need to specify any parameters in the parameter section. However, you can specify any of the additional parameters in the parameter section or in the environmental variables.

### Specifying a Printer Name

You can specify a printer name in different ways.

To specify the name of a local area network (LAN) printer, use the UNC for that device. To specify a printer name 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  
/REMOTE_PRINTER_NAME=\\SERVER1\HP_LASERJET_QUEUE file_name
```

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

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

### Printer Name Parameters

The name of a printer has several related parameters.

#### REMOTE\_PRINTER\_NAME

|                  |                    |
|------------------|--------------------|
| Default Value    | Not applicable     |
| Allowable Values | 1 - 255 characters |
| Minimum Value    | 1 character        |
| Maximum Value    | 255 characters     |

You can use the **REMOTE\_PRINTER\_NAME** parameter to specify the name of the printer to which the job is sent.

**SYSOUT\_CLASS**

|                          |              |
|--------------------------|--------------|
| Default Value            | None         |
| Allowable Values         | 0 - 9, A - Z |
| Minimum Value            | 0, A, or a   |
| Maximum Value            | 9, Z, or z   |
| Alternate Specifications | CL           |

You can use the **SYSOUT\_CLASS** parameter to specify the class to which the JES output is routed. On a z/OS system, the printer queues are organized around a printer class, and not a specific printer. The class has a one-character name which is either alphabetic or numeric. You can specify the value according to z/OS.

**SYSOUT\_COPIES**

|                          |         |
|--------------------------|---------|
| Default Value            | None    |
| Allowable Values         | 1 - 999 |
| Minimum Value            | 1       |
| Maximum Value            | 999     |
| Alternate Specifications | SP      |

You can use the **SYSOUT\_COPIES** parameter to specify the number of copies to print of a particular report on the remote computer.

**SYSOUT\_DESTINATION**

|                          |                  |
|--------------------------|------------------|
| Default Value            | None             |
| Allowable Values         | 1 - 8 characters |
| Minimum Value            | Not applicable   |
| Maximum Value            | Not applicable   |
| Alternate Specifications | SD               |

You can use the **SYSOUT\_DESTINATION** parameter to specify the destination of the job submitted to the internal reader.

**SYSOUT\_FCB**

|                  |                  |
|------------------|------------------|
| Default Value    | None             |
| Allowable Values | 1 - 4 characters |

|                          |                |
|--------------------------|----------------|
| Minimum Value            | Not applicable |
| Maximum Value            | Not applicable |
| Alternate Specifications | SB             |

The **SYSOUT\_FCB** parameter is applied when the remote computer uses a z/OS system. You can use this parameter to specify the name of a form control buffer as defined to JES.

### **SYSOUT\_FORM**

|                          |                  |
|--------------------------|------------------|
| Default Value            | None             |
| Allowable Values         | 1 - 8 characters |
| Minimum Value            | Not applicable   |
| Maximum Value            | Not applicable   |
| Alternate Specifications | SF               |

You can use the **SYSOUT\_FORM** parameter to specify the form name upon which the report is printed on the remote computer.

### **SYSOUT\_USERNAME**

|                          |                  |
|--------------------------|------------------|
| Default Value            | None             |
| Allowable Values         | 1 - 8 characters |
| Minimum Value            | Not applicable   |
| Maximum Value            | Not applicable   |
| Alternate Specifications | SI               |

You can use the **SYSOUT\_USERNAME** parameter to specify the user name assigned to a job submitted to the internal reader.

### **SYSOUT\_WRITER**

|                          |                  |
|--------------------------|------------------|
| Default Value            | None             |
| Allowable Values         | 1 - 8 characters |
| Minimum Value            | Not applicable   |
| Maximum Value            | Not applicable   |
| Alternate Specifications | SW               |

You can use the **SYSOUT\_WRITER** parameter to specify the external writer name that is used to process a printer file on the z/OS. This is the name of a service program on the z/OS, which is given control when it is time to process this file from the printer queue. The service program written by the customer decides how it

processes this print file. Do not specify a value for this parameter unless directed to by the system analyst on the z/OS.

## Remote Command Transfers

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

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

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

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

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

The following example illustrates an execution of the **dir** command on a remote machine and whose output is stored on the local machine in the *local\_file\_name* file.

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

In the following 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 name is unnecessary.

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

## Parameters

MFT Platform Server for Windows uses a number of different parameters. Some variables are specified as part of the parameters on the program call.

All of the parameters, except for **local filename** and **remote file name**, can be specified both as environmental variables and as parameters on a command line. Each parameter can be specified in three different ways, all of which are valid both on a command line and as an environment variable. For example, **data type** can be specified as **DATA\_TYPE**, *DataType*, and **DT**.

When entering a parameter on a command line, you must type a forward slash (/) before the parameter name. For example, **/DATA\_TYPE=E**.

## Optional Parameters

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

**ALLOCATION\_TYPE={ TRACKS | CYLINDERS | MEGABYTES | KILOBYTES }**

|                          |   |
|--------------------------|---|
| Default Value            | TRACKS                                  |
| Allowable Values         | TRACKS, CYLINDERS, MEGABYTES, KILOBYTES |
| Minimum Value            | Not applicable                          |
| Maximum Value            | Not applicable                          |
| Alternate Specifications | AllocationType, AT                      |

You can use the **ALLOCATION\_TYPE** parameter to instruct z/OS to create new files. This parameter is ignored when it is sent to a platform other than z/OS. The following table lists valid parameter values and their descriptions.

| Parameter Value |           | Description                                    |
|-----------------|-----------|--|
| T               | Tracks    | Used when data size is expressed in tracks.    |
| C               | Cylinders | Used when data size is expressed in cylinders. |
| M               | Megabytes | Used when data size is expressed in megabytes. |
| K               | Kilobytes | Used when data size is expressed in kilobytes. |

#### **ALLOCATION\_PRIMARY**

|                          |                       |
|--------------------------|-----------------------|
| Default Value            | Not applicable        |
| Allowable Values         | Numeric values        |
| Minimum Value            | Not applicable        |
| Maximum Value            | Not applicable        |
| Alternate Specifications | AllocationPrimary, AP |

The primary allocation field defines the z/OS primary allocation quantity when creating a new dataset.

#### **ALLOCATION\_SECONDARY**

|                          |                         |
|--------------------------|-------------------------|
| Default Value            | Not applicable          |
| Allowable Values         | Numeric values          |
| Minimum Value            | Not applicable          |
| Maximum Value            | Not applicable          |
| Alternate Specifications | AllocationSecondary, AS |

The secondary allocation field defines the z/OS primary allocation quantity when creating a new dataset.

#### **BLOCK\_SIZE**

|                  |                |
|------------------|----------------|
| Default Value    | Not applicable |
| Allowable Values | Numeric values |

|                          |                |
|--------------------------|----------------|
| Minimum Value            | Not applicable |
| Maximum Value            | Not applicable |
| Alternate Specifications | BlockSize, BS  |

You can use the **BLOCK\_SIZE** parameter to specify the size of a block. For FB, the block size must be a multiple of record length, and for VB, the record length can be any size up to the block size minus four. The maximum number is 32760.

#### **CHECK\_POINT\_RESTART={ YES | NO | *nn* }**

|                          |                            |
|--------------------------|----------------------------|
| Default Value            | YES (5 minutes by default) |
| Allowable Values         | YES, NO, <i>nn</i>         |
| Minimum Value            | 1 minute                   |
| Maximum Value            | 90 minutes                 |
| Alternate Specifications | CheckpointRestart, CP      |



The **CHECK\_POINT\_RESTART** parameter requires you to submit your transfer.

When checkpoint restart is enabled using the **CHECK\_POINT\_RESTART** parameter, data packets are sent periodically within file transfers. These data packets 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. A checkpoint is specified in units of time.

The following table lists allowable parameter values and their descriptions.

| Parameter Value | Description   |
|-----------------|---|
| YES             | Turn on checkpoint restart using the default interval of 5 minutes. |
| NO              | Turn off checkpoint restart.  |
| <i>nn</i>       | Turn on checkpoint restart using the interval of <i>nn</i> minutes. |

#### **COMMAND=**

|                  |                        |
|------------------|------------------------|
| Default Value    | Not applicable         |
| Allowable Values | Command to be executed |
| Minimum Value    | Not applicable         |
| Maximum Value    | Not applicable         |

Alternate Specifications

RC, RemoteCommand, E,  
EXEC, RE, REXXEXEC, SJ,  
SUBJCL, CJ, CALLJCL,  
CPG, CALLPGM

You can use the **COMMAND=** parameter with the File to Remote Command feature.

The alternate specifications for the **COMMAND=** parameter depend on the remote system that the command is executed on. The following table lists the relationship between the commands and platforms.

| 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            |

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

|                          |                           |
|--------------------------|---------------------------|
| Default Value            | NO                        |
| Allowable Values         | YES, RLE, LZ, Z1 - Z9, NO |
| Minimum Value            | Not applicable            |
| Maximum Value            | Not applicable            |
| Alternate Specifications | Compression, CM           |

You can use the **COMPRESSION** parameter to compress data at the sender side of a 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 than RLE and compresses more different types of data but uses a lot of CPU Cycles. We suggest using ZLIB2 compression since it is faster and compresses data better.



RLE is more data-dependent than LZ. The compression ratio of RLE may vary widely based on the type of data being compressed. Select RLE if network bandwidth is not a critical bottleneck for your network and you need to save CPU cycles.

Z1 through Z9 refer to the levels of zlib compression. Level 1 offers very fast compression. Levels 7 to 9 yield better compression at a much slower speed. Level 2 ZLIB2 typically offers the best compromise of compression and speed.

#### **CR\_LF={ YES | NO }**

|                          |                |
|--------------------------|----------------|
| Default Value            | NO             |
| Allowable Values         | YES, NO        |
| Minimum Value            | Not applicable |
| Maximum Value            | Not applicable |
| Alternate Specifications | CrLf, CR       |

You can use the **CR\_LF** parameter to define whether carriage return/line feed translation is performed during a transfer. This parameter has no effect when it is sent with the **DATA\_TYPE** parameter set to **B** (binary).

#### **CRCCHECK={N | Y }**

|                          |                |
|--------------------------|----------------|
| Default Value            | NO             |
| Allowable Values         | YES, NO        |
| Minimum Value            | Not applicable |
| Maximum Value            | Not applicable |
| Alternate Specifications | CRC            |

#### **DATA\_TYPE={ A | B | E }**

|                          |                |
|--------------------------|----------------|
| Default Value            | E              |
| Allowable Values         | A, B, E        |
| Minimum Value            | Not applicable |
| Maximum Value            | Not applicable |
| Alternate Specifications | DataType, DT   |

You can use the **DATA\_TYPE** parameter to specify what format data is stored in on the remote system. A value of **B** indicates that there is no conversion done.

**DATA\_CLASS**

|                          |                  |
|--------------------------|------------------|
| Default Value            | Not applicable   |
| Allowable Values         | 1 - 8 characters |
| Minimum Value            | 1 character      |
| Maximum Value            | 8 characters     |
| Alternate Specifications | DataClass, DC    |

You can use the **DATA\_CLASS** parameter to specify the z/OS data class as defined to the Data Facility /System Managed Storage. In addition, this parameter is used to indirectly select file attributes such as Record Format and Logical Record Length. The parameter value is a string of 1 to 8 characters, which contain either numeric, alphabetic, or national characters (in the United States these are \$, #, or @). The first character must be alphabetic or national.

**DESTINATION**

|                          |                                       |
|--------------------------|---------------------------------------|
| Default Value            | Not applicable                        |
| Allowable Values         | LU name, IP name, or IP address       |
| Minimum Value            | Not applicable                        |
| Maximum Value            | Not applicable                        |
| Alternate Specifications | Destination, DS, LuName, LU, Host, HO |

You can use the **DESTINATION** parameter to specify the address of the remote system.

**ENCRYPTION = {DES | 3DES | BF | BFL | AES | NONE}**

|                          |                               |
|--------------------------|-------------------------------|
| Default Value            | OFF                           |
| Allowable Values         | DES, 3DES, BF, BFL, AES, NONE |
| Minimum Value            | Not applicable                |
| Maximum Value            | Not applicable                |
| Alternate Specifications | en                            |



You can select only one type of encryption per transfer. We suggest selecting AES when performing encryption.

**EXPIRATION\_DATE**

|                          |   |
|--------------------------|---|
| Default Value            | None  |
| Allowable Values         | MM/DD/YYYY,<br>HH:MM:SS, SUN, MON,<br>TUES, WED, THURS, FRI,<br>SAT |
| Minimum Value            | Not applicable  |
| Maximum Value            | Not applicable  |
| Alternate Specifications | ExpirationDate, ED  |

You can use the **EXPIRATION\_DATE** parameter to specify the exact date and time when a transfer no longer is attempted. However, if the transfer is scheduled, it takes precedence over expiration. If Expiration and Retention are used, then whichever value occurs first takes precedence.

**FILE\_AVAIL={ I | D }**

|                          |                      |
|--------------------------|----------------------|
| Default Value            | I                    |
| Allowable Values         | I, D                 |
| Minimum Value            | Not applicable       |
| Maximum Value            | Not applicable       |
| Alternate Specifications | FileAvailability, FA |

The following table lists valid parameter values and their descriptions.

| Parameter Value |           | Description   |
|-----------------|-----------|---|
| I               | Immediate | The file is available to be transferred immediately. The default value is I.  |
| D               | Deferred  | The remote file availability may be deferred if the remote system uses this option. In the responder function, MFT Platform Server treats Deferred as tape and Immediate as disk. |

**FILE\_TRANSFER\_SERVER**

|               |      |
|---------------|------|
| Default Value | NONE |
|---------------|------|

|                          |                        |
|--------------------------|------------------------|
| Allowable Values         | 1 - 31 characters      |
| Minimum Value            | 1 character            |
| Maximum Value            | 31 characters          |
| Alternate Specifications | FileTransferServer, FS |

You can use the **FILE\_TRANSFER\_SERVER** parameter with the **Submit** parameter to submit a transfer to another MFT Platform Server. The MFT Platform Server uses the **ServerName** parameter to obtain an RPC Binding Handle to the MFT Platform Server that is processing the file transfer. Then the MFT Platform Server submits the transfer to the server's queue.

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

If an MFT Platform Server is selected and the **Submit** parameter is not specified, MFT Platform Server accepts the request for a transfer. However, it only performs a two-stage client to host the transfer.

You can select a server that resides in a different domain than the domain from where the file transfer is initiated. The selection is done by specifying the domain name and server name in the file transfer server parameter as follows:

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

### LOCAL\_CTFILE

|                          |                      |
|--------------------------|----------------------|
| Default Value            | Not applicable       |
| Allowable Values         | 1 - 16 characters    |
| Minimum Value            | 1 character          |
| Maximum Value            | 16 characters        |
| Alternate Specifications | InitiatorCTFile, LCT |

You can use the **LOCAL\_CTFILE** parameter to convert data between ASCII and EBCDIC. The parameter value contains the name of a file, which is used to translate on the local side. This parameter is unnecessary if you are communicating from PC to PC.

### LOCAL\_DOMAIN

|                          |                   |
|--------------------------|-------------------|
| Default Value            | Not applicable    |
| Allowable Values         | 1 - 15 characters |
| Minimum Value            | 1 character       |
| Maximum Value            | 15 characters     |
| Alternate Specifications | LocalDomain, LD   |

You can use the **LOCAL\_DOMAIN** parameter to provide information about the user who initiates a transfer. This parameter is only used when you specify the Submit option.

**LOCAL\_PASSWORD**

|                          |                   |
|--------------------------|-------------------|
| Default Value            | X:                |
| Allowable Values         | 1 - 20 characters |
| Minimum Value            | 1 character       |
| Maximum Value            | 20 characters     |
| Alternate Specifications | LocalPassword, LW |

You can use the **LOCAL\_PASSWORD** parameter to provide the local logon password. The password can be a string of up to 20 characters and is case sensitive. This parameter is only used when you specify the Submit option.

**LOCAL\_USER\_ID**

|                          |                   |
|--------------------------|-------------------|
| Default Value            | None              |
| Allowable Values         | 1 - 20 characters |
| Minimum Value            | 1 character       |
| Maximum Value            | 20 characters     |
| Alternate Specifications | LocalUserId, LI   |

You can use the **LOCAL\_USER\_ID** parameter to provide information about the user who initiates a transfer. The parameter value is not case-sensitive. This parameter is only used when you specify the Submit option.

**LIST**

|                          |                   |
|--------------------------|-------------------|
| Default Value            | None              |
| Allowable Values         | 1 - 32 characters |
| Minimum Value            | 1 character       |
| Maximum Value            | 32 characters     |
| Alternate Specifications | list              |

You can use the **LIST** parameter to assign the distribution list to use for the transfer request.

**MGMT\_CLASS**

|                  |                  |
|------------------|------------------|
| Default Value    | None             |
| Allowable Values | 1 - 8 characters |
| Minimum Value    | 1 character      |

|                          |               |
|--------------------------|---------------|
| Maximum Value            | 8 characters  |
| Alternate Specifications | MgmtClass, MC |

You can use the **MGMT\_CLASS** parameter to define the z/OS Management Class as defined to the Data Facility /System Managed Storage.

The parameter value is a string of 1 to 8 characters, which contain either numeric, alphabetic, or national characters (in the United States these are \$, #, or @). The first character must be alphabetic or national.

#### **NOTIFY\_LOCAL\_USER=local\_user\_email**

|                          |                      |
|--------------------------|----------------------|
| Default Value            | None                 |
| Allowable Values         | 1 - 255 characters   |
| Minimum Value            | 1 character          |
| Maximum Value            | 255 characters       |
| Alternate Specifications | NotifyLocalUser, NLU |

The parameter value is the 1-255 character name of the local users to notify when a file transfer is complete, either successfully or unsuccessfully. For this name, it support multiple emails separated by a comma.

#### **NOTIFY\_LOCAL\_USER\_TYPE=M[AIL] [:S[UCCESS] | F[AILURE]]**

|                          |                           |
|--------------------------|---------------------------|
| Default Value            | None                      |
| Allowable Values         | MAIL                      |
| Minimum Value            | Not Applicable            |
| Maximum Value            | Not Applicable            |
| Alternate Specifications | NotifyLocalUserType, NLUT |

You can use the **NOTIFY\_LOCAL\_USER\_TYPE** parameter to define the type of the user ID to notify after a file transfer terminates. The parameter is used with the **NOTIFY\_LOCAL\_USER=** parameter. The following table lists valid parameter values and their descriptions.

| Parameter Value | Description  |
|-----------------|--|
| MAIL            | To provide e-mail notification for both successful and failed transfers. |
| MAIL:SUCCESS    | To provide e-mail notification only for successful transfers.            |
| MAIL:FAILURE    | To provide e-mail notification only for failed transfers.                |

**NOTIFY\_REMOTE\_USER=remote\_user\_email**

|                          |                       |
|--------------------------|-----------------------|
| Default Value            | None                  |
| Allowable Values         | 1 - 255 characters    |
| Minimum Value            | 1 character           |
| Maximum Value            | 255 characters        |
| Alternate Specifications | NotifyRemoteUser, NRU |

The parameter value is the 1-255 character name of the remote users to notify when a file transfer is complete, either successfully or unsuccessfully. For this name, it support multiple emails separated by a comma.

**NOTIFY\_REMOTE\_USER\_TYPE=M[AIL] [:S[UCCESS] | F[AILURE]]**

|                          |                            |
|--------------------------|----------------------------|
| Default Value            | None                       |
| Allowable Values         | MAIL                       |
| Minimum Value            | Not Applicable             |
| Maximum Value            | Not Applicable             |
| Alternate Specifications | NotifyRemoteUserType, NRUT |

You can use the **NOTIFY\_REMOTE\_USER\_TYPE** parameter to define the type of the user ID to notify after a file transfer terminates. The parameter is used with the **NOTIFY\_REMOTE\_USER=** parameter. The following table lists valid parameter values and their descriptions.

| Parameter Value | Description  |
|-----------------|--|
| MAIL            | To provide e-mail notification for both successful and failed transfers. |
| MAIL:SUCCESS    | To provide e-mail notification only for successful transfers.            |
| MAIL:FAILURE    | To provide e-mail notification only for failed transfers.                |

**PERMITTED\_ACTIONS={S | H | A | R | C | Z | E | T }**

|                  |                        |
|------------------|------------------------|
| Default Value    | None                   |
| Allowable Values | S, H, A, R, C, Z, E, T |
| Minimum Value    | Not applicable         |

|                          |                      |
|--------------------------|----------------------|
| Maximum Value            | Not applicable       |
| Alternate Specifications | PermittedActions, PA |

The following table lists valid parameter values and their descriptions.

| Parameter Value |                        | Description   |
|-----------------|------------------------|---|
| S               | System                 | The file is a system file and can be displayed only by the operating system.  |
| H               | Hidden                 | A file is invisible to you.   |
| A               | Archive                | This action is to mark a file that has changed since it was last backed up.   |
| R               | Read Only              | A file being accessed can only be read. No changes can be made to the file.   |
| C               | NTFS Compressed        | This action is to compress a file that is going to an NTFS drive.   |
| Z               | Control Z              | When enabled, the feature appends a CR/LF (0x0d, 0x0a) to the end of a file, followed by the DOS End of File character - Control Z (0x1a). If a trailing Control Z or CR/LF already exists, no addition is made. This feature is only available when Carriage Return/Line Feed processing is enabled. |
| E               | Control Z added to EOF | When enabled, the feature appends a Control Z (0x1a) to the end of a file.  |



| Parameter Value | Description  |
|-----------------|--|
| T               | CR/LF added to EOF   |
|                 | When enabled, the feature appends a CR/LF (0x0d, 0x0a) to the end of a file. |

## PORT

|                  |           |
|------------------|-----------|
| Default Value    | 46464     |
| Allowable Values | 1 - 65535 |

You can use the **PORT** parameter to provide the port number for a TCP/IP transfer. The default port number for MFT Platform Server is 46464, but you can change it to any number between 1 and 65535, inclusive. However, some small port numbers may be reserved for standard applications at your installation. For a TLS or Tunnel transfer, the **SECURE** parameter must also be used.

## PRIORITY= { 3 | *n* }

|                          |              |
|--------------------------|--------------|
| Default Value            | 3            |
| Allowable Values         | 1 - 6        |
| Minimum Value            | 1            |
| Maximum Value            | 6            |
| Alternate Specifications | Priority, PR |

You can use the **PRIORITY** parameter to specify the priority that is applied when the thread for a file transfer is created. This priority 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, indicating the priority of a file transfer. A smaller value indicates a higher priority. TIBCO recommends you set the default value to 3.

## PROCESS\_NAME

|                          |                  |
|--------------------------|------------------|
| Default Value            | CyberFus         |
| Allowable Values         | 1 - 8 characters |
| Minimum Value            | 1 character      |
| Maximum Value            | 8 characters     |
| Alternate Specifications | ProcessName, PN  |

The parameter value is an eight-character string that describes the transaction being processed.

## REMOTE\_CTFILE

|               |                |
|---------------|----------------|
| Default Value | Not applicable |
|---------------|----------------|

|                          |                      |
|--------------------------|----------------------|
| Allowable Values         | 1 - 16 characters    |
| Minimum Value            | 1 character          |
| Maximum Value            | 16 characters        |
| Alternate Specifications | ResponderCTFile, RCT |

You can use the **REMOTE\_CTFILE** parameter to convert data between ASCII and EBCDIC. The parameter value contains the name of a file, which is used to translate on the remote side. This parameter is unnecessary if you are communicating from PC to PC.

#### **RECORD\_FORMAT={ F | FB | V | VB | U }**

|                          |                  |
|--------------------------|------------------|
| Default Value            | None             |
| Allowable Values         | F, FB, V, VB, U  |
| Minimum Value            | Not applicable   |
| Maximum Value            | Not applicable   |
| Alternate Specifications | RecordFormat, RF |

You can use the **RECORD\_FORMAT** parameter to define the significance of the character logical record length. You can specify a fixed, variable, or undefined format. This parameter is specific to z/OS. The following table lists allowable values and their descriptions.

| Parameter Value |                | Description   |
|-----------------|----------------|---|
| F               | Fixed          | Each string contains exactly the number of characters defined by the <b>string length</b> parameter.          |
| FB              | Fixed Block    | All blocks and all logical records are fixed in size. One or more logical records reside in each block.       |
| V               | Variable       | The length of each string is less than or equal to the value of the <b>string length</b> parameter.           |
| VB              | Variable Block | Blocks as well as logical record length can be of any size. One or more logical records reside in each block. |

| Parameter Value | Description  |
|-----------------|--|
| U               | Undefined<br><br>Blocks are of an undefined size. There are no logical records. The logical record length appears as zero. This record format is usually only used in load libraries. Block size must be used if you specify the undefined format. |

#### **RECORD\_LENGTH={ nnnnn | 0 }**

|                          |                            |
|--------------------------|----------------------------|
| Default Value            | None                       |
| Allowable Values         | 1 - 32760                  |
| Minimum Value            | 1 (F or FB)<br>4 (V or VB) |
| Maximum Value            | 32760                      |
| Alternate Specifications | RecordLength, RL           |

You can use the **RECORD\_LENGTH** parameter to provide the maximum logical record length, which is sometimes called the string length used to encode the data records of a file. The maximum logical record length in z/OS is 32760. TIBCO recommends that you omit this parameter if you are sending or receiving a file into a file that already exists since MFT Platform Server determines the appropriate length. This parameter is ignored when it is sent to MFT Platform Server for Windows because it is a z/OS-specific parameter.



If the **RECORD\_FORMAT** parameter is set to F or FB, the allowable values of the **RECORD\_LENGTH** parameter are 1 to 32760. If the **RECORD\_FORMAT** parameter is set to V or VB, the allowable values of the **RECORD\_LENGTH** parameter are 4 to 32760.

#### **REMOTE\_DOMAIN**

|                          |   |
|--------------------------|---|
| Default Value            | The domain of the remote system where MFT Platform Server is executing. |
| Allowable Values         | 1 - 15 characters   |
| Minimum Value            | 1 character   |
| Maximum Value            | 15 characters   |
| Alternate Specifications | RemoteDomain, RD  |

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

**REMOTE\_PASSWORD**

|                          |                    |
|--------------------------|--------------------|
| Default Value            | None               |
| Allowable Values         | 1 - 20 characters  |
| Minimum Value            | 1 character        |
| Maximum Value            | 20 characters      |
| Alternate Specifications | RemotePassword, RW |

You can use the **REMOTE\_PASSWORD** parameter to provide the remote logon password. The password can be a string of up to 20 characters and is case sensitive. Specify this parameter only if required by the remote computer.

**REMOTE\_PRINTER\_NAME**

|                          |                       |
|--------------------------|-----------------------|
| Default Value            | Not applicable        |
| Allowable Values         | 1 - 255 characters    |
| Minimum Value            | 1 character           |
| Maximum Value            | 255 characters        |
| Alternate Specifications | RemotePrinterName, RP |

You can use the **REMOTE\_PRINTER\_NAME** parameter to specify the name of the remote printer to which the job is sent when using File to Job.

**REMOTE\_USER\_ID**

|                          |                   |
|--------------------------|-------------------|
| Default Value            | None              |
| Allowable Values         | 1 - 20 characters |
| Minimum Value            | 1 character       |
| Maximum Value            | 20 characters     |
| Alternate Specifications | RemoteUserId, RI  |

You can use the **remote user ID** parameter to specify 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.

**REMOVE\_TRAILING\_SPACES**

|                  |      |
|------------------|------|
| Default Value    | N    |
| Allowable Values | Y, N |

|                          |                           |
|--------------------------|---------------------------|
| Minimum Value            | N/A                       |
| Maximum Value            | N/A                       |
| Alternate Specifications | RemoveTrailingSpaces, RTS |

You can use the **REMOVE\_TRAILING\_SPACES** parameter to remove all spaces or binary zeros at the end of a record when transferred from the z/OS platform.

### RETENTION\_PERIOD

|                          |                     |
|--------------------------|---------------------|
| Default Value            | 0                   |
| Allowable Values         | 0 - 32767           |
| Minimum Value            | 0                   |
| Maximum Value            | 32767               |
| Alternate Specifications | RetentionPeriod, RT |

You can use the **RETENTION\_PERIOD** parameter to specify the number of days that pass from the transfer start to the point it is no longer attempted. If Expiration and Retention are used, then whichever value occurs first takes precedence.

### RSAccelerator

|                          |      |
|--------------------------|------|
| Default Value            | N    |
| Allowable Values         | Y, N |
| Minimum Value            | N/A  |
| Maximum Value            | N/A  |
| Alternate Specifications | RSA  |

Setting the **RSAccelerator** parameter to Y forces a transfer to be conducted through a Windows MFT Platform TIBCO Accelerator server that uses the TIBCO Accelerator technology. Using the technology greatly improves data transfer speeds over IP networks with high latency.



You must be licensed for RSA to use this technology.

### RSCompression

|                  |                            |
|------------------|----------------------------|
| Default Value    | N                          |
| Allowable Values | N, Y   Best, Default, Fast |
| Minimum Value    | N/A                        |
| Maximum Value    | N/A                        |

## Alternate Specifications

RSC, RSCOMPRESS

When conducting file transfers through an RSAccelerator (RSA), you can configure the RSA server to compress the data being transferred. The RSA uses a proprietary compression compatible with zlib. By setting the **RSCompression** parameter to `Default`, your file receives the greatest compression and may take slightly longer time to transfer. If you set the parameter to `Fast`, your file is less compressed but sent out faster.

**RSEncryption**

|                          |                |
|--------------------------|----------------|
| Default Value            | N              |
| Allowable Values         | Y, N           |
| Minimum Value            | N/A            |
| Maximum Value            | N/A            |
| Alternate Specifications | RSE, RSENCRYPT |

When conducting file transfers through an RSA, you can instruct the RSA server to encrypt data with a 256-bit Blowfish encryption key by setting the **RSEncryption** parameter to `Y`.

**RSHost**

|                          |         |
|--------------------------|---------|
| Default Value            | None    |
| Allowable Values         | Host, N |
| Minimum Value            | N/A     |
| Maximum Value            | N/A     |
| Alternate Specifications | RSH     |

You can use the **RSHost** parameter to specify the IP address or host name of the Windows MFT Platform TIBCO Accelerator server. By defining a host on a command line or in a transfer template, you override the **RSHost** parameter value configured in the `config.txt` file if it is defined. If you set the parameter to `N` and the **RSAccelerator** parameter to `Y`, the value configured for **RSHost** in the `config.txt` file is used.

**RSMaxSpeed**

|                          |               |
|--------------------------|---------------|
| Default Value            | 1000000       |
| Allowable Values         | 256 - 1000000 |
| Minimum Value            | N/A           |
| Maximum Value            | N/A           |
| Alternate Specifications | RSMAX         |

When using the TIBCO Accelerator, this parameter sets the Max Speed in Kilobytes per second.

**RSPort**

|                          |         |
|--------------------------|---------|
| Default Value            | None    |
| Allowable Values         | Port, N |
| Minimum Value            | N/A     |
| Maximum Value            | N/A     |
| Alternate Specifications | RSPORT  |

You can use the **RSPort** parameter to specify the port number the Windows MFT Platform TIBCO Accelerator server is listening on for transfers using the TIBCO Accelerator technology. By specifying a port number on a command line or in a transfer template, you override the **RSPort** parameter value configured in the `config.txt` file. The default value is 9099. If you set the parameter to N and the **RSAccelerator** parameter to Y, the value configured for **RSPort** in the `config.txt` file is used.

**RSProtocol**

|                          |               |
|--------------------------|---------------|
| Default Value            | None          |
| Allowable Values         | TCP, UDP, PDP |
| Minimum Value            | N/A           |
| Maximum Value            | N/A           |
| Alternate Specifications | RSP           |

During file transfers through an RSA, you can instruct the RSA server to use its own enhanced version of UDP, TIBCO Accelerator's parallel implementation of TCP, called PDP, or straight TCP.

**SCHEDULE\_AT**

|                          |                      |
|--------------------------|----------------------|
| Default Value            | None                 |
| Allowable Values         | MM/DD/YYYY, HH:MM:SS |
| Minimum Value            | Not applicable       |
| Maximum Value            | Not applicable       |
| Alternate Specifications | ScheduleAt, SAT      |

You can use the **SCHEDULE\_AT** parameter to specify the date and time when a transfer is executed.



The **SCHEDULE\_AT** parameter requires you to submit your transfer.

**SCHEDULE\_REPEAT = { N | I | T:x | U }**

|               |      |
|---------------|------|
| Default Value | None |
|---------------|------|

|                          |                      |
|--------------------------|----------------------|
| Allowable Values         | N, I, T:x, U         |
| Minimum Value            | Not applicable       |
| Maximum Value            | Not applicable       |
| Alternate Specifications | Schedule repeat, SRE |

You can use the **SCHEDULE\_REPEAT** parameter to specify the rate at which the schedule is repeated. The following table lists valid parameter values and their descriptions.

| Parameter Value |          | Description  |
|-----------------|----------|--|
| N               | NO       | Do not repeat a transfer.  |
| I               | INFINITE | Repeat a transfer forever.   |
| T:x             | TIMES    | Repeat a transfer for <i>x</i> times.  |
| U               | UNTIL    | Repeat a transfer until the specified date and time.<br>Format: MM/DD/YYYY, HH:MM:SS |

**SCHEDULE\_INTERVAL = {D7|WK|2WK|MON|2MON|QTR|2QTR|YR|2YR| E:n:u }**

|                          |   |
|--------------------------|---|
| Default Value            | None  |
| Allowable Values         | D7, WK, 2WK, MON, 2MON, QTR, 2QTR, YR, 2YR, E:n:u |
| Minimum Value            | Not applicable                                    |
| Maximum Value            | Not applicable                                    |
| Alternate Specifications | ScheduleInterval, SRI                             |

You can use the **SCHEDULE\_INTERVAL** parameter to specify the interval at which the transfer is repeated. This parameter can be used only when you are scheduling a transfer. The following table lists valid parameter values and their descriptions.

| Parameter Value |           | Description              |
|-----------------|-----------|--------------------------|
| D7              | Daily 7   | Sunday through Saturday. |
| WK              | Weekly    | Every week.              |
| 2WK             | Bi-Weekly | Every other week.        |



| Parameter Value |               | Description  |
|-----------------|---------------|--|
| MON             | Monthly       | Every month.   |
| 2MON            | Bi-Monthly    | Every other month.   |
| QTR             | Quarterly     | Every 3 months.  |
| 2QTR            | Semi-Annually | Every 6 months.  |
| YR              | Yearly        | Every year.  |
| 2YR             | Bi-Yearly     | Every other year.  |
| E:n:u           | Every         | Every <i>n</i> second(s), minute(s), hour(s), day(s), week(s), month(s), or year(s). |

#### SECURITY\_ATTRIB\_FILENAME

|                          |                            |
|--------------------------|----------------------------|
| Default Value            | CyberFus                   |
| Allowable Values         | 1 - 8 characters           |
| Minimum Value            | 1 character                |
| Maximum Value            | 8 characters               |
| Alternate Specifications | SecurityAttribFileName, SA |

You can use the **SECURITY\_ATTRIB\_FILENAME** parameter to specify the file name that the receiving partner uses as a template for its ACL. The ACL of this file is copied to that of the destination file. For this feature to function properly on Windows, the specified file must be readable by the partner which receives the file to file transfer, and the file being created must reside on an NTFS drive.

#### StopOnFailure

|                          |      |
|--------------------------|------|
| Default Value            | N    |
| Allowable Values         | Y, N |
| Minimum Value            | N/A  |
| Maximum Value            | N/A  |
| Alternate Specifications | sof  |

You can use the **StopOnFailure** parameter for directory transfers and transfers using a distribution list. This parameter indicates if the current file transfer fails, the rest of files are not transferred.

**STORE\_CLASS**

|                          |                  |
|--------------------------|------------------|
| Default Value            | None             |
| Allowable Values         | 1 - 8 characters |
| Minimum Value            | 1 character      |
| Maximum Value            | 8 characters     |
| Alternate Specifications | StoreClass, SC   |

You can use the **STORE\_CLASS** parameter to define the z/OS Storage Class as defined to the Data Facility / System Managed Storage that is used to indicate the host file's media type and the installation's backup, restore, and archive policies. The parameter value must contain either numeric, alphabetic, or national characters (in the United States these are \$, #, or @). The first character must be alphabetic or national.

**Test**

|                  |      |
|------------------|------|
| Default Value    | N    |
| Allowable Values | Y, N |
| Minimum Value    | N/A  |
| Maximum Value    | N/A  |

When performing a directory transfer, you can use the **Test** parameter to display the Local and Remote File Names rather than do the actual transfers as a means of verifying that file names are correct. This parameter is used when running directory transfer requests and transfers using a distribution list.

**TRACE\_LEVEL**

|                          |                |
|--------------------------|----------------|
| Default Value            | 1              |
| Allowable Values         | 1 - 9          |
| Minimum Value            | 1              |
| Maximum Value            | 9              |
| Alternate Specifications | TraceLevel, TL |

You can use the **TRACE\_LEVEL** parameter to define the level of messages that are produced during a transfer. Higher values produce more output but slow system performance. The traces are written to the trace directory defined by the Server properties Trace tab.

**TRUNCATE={Y| N | W}**

|                  |                |
|------------------|----------------|
| Default Value    | None           |
| Allowable Values | Yes, No , Wrap |

|                          |                |
|--------------------------|----------------|
| Minimum Value            | Not Applicable |
| Maximum Value            | Not Applicable |
| Alternate Specifications | TRN            |

#### **TRY\_COUNT= { nn | 1 }**

|                          |                            |
|--------------------------|----------------------------|
| Default Value            | 1                          |
| Allowable Values         | 1 - 10 or unlimited (or 0) |
| Minimum Value            | 1                          |
| Maximum Value            | Unlimited                  |
| Alternate Specifications | TryCount, TC               |

You can use the **TRY\_COUNT** parameter to specify the maximum number of times that a file transfer can be attempted before it is purged from the MFT Platform Server work queue. *nn* is a decimal number ranging from 0 to 10.

If you set the **TRY\_COUNT** parameter to 0, a file transfer is attempted infinitely. The default parameter value is recommended.

#### **UNIT = SYSDA**

|                          |                  |
|--------------------------|------------------|
| Default Value            | SYSDA            |
| Allowable Values         | 1 - 8 characters |
| Minimum Value            | 1 character      |
| Maximum Value            | 8 characters     |
| Alternate Specifications | Unit, UN         |

You can use the **UNIT** parameter to specify a unit where the Host data set is to be allocated. The unit name contains 1 to 8 characters.

#### **USER\_DATA= User Data/Description**

|                          |                   |
|--------------------------|-------------------|
| Default Value            | None              |
| Allowable Values         | 0 - 25 characters |
| Minimum Value            | 0 or None         |
| Maximum Value            | 25 characters     |
| Alternate Specifications | UserData, UD      |

You can use the **USER\_DATA** parameter to describe a transfer on the local and remote system. The description is logged into history files. If you need to embed spaces in this parameter, you can either specify

this parameter in the Environment Variable (SET command) or enclose the value in double quotation marks. The description can contain any alphabetic, numeric, or national characters of up to 25 characters.

#### **UTF8BOM={A| R| B}**

|                          |                   |
|--------------------------|-------------------|
| Default Value            | None              |
| Allowable Values         | Add, Remove, Both |
| Minimum Value            | Not Applicable    |
| Maximum Value            | Not Applicable    |
| Alternate Specifications | BOM               |

#### **VOL\_SER**

|                          |                        |
|--------------------------|------------------------|
| Default Value            | None                   |
| Allowable Values         | 1 - 6 characters       |
| Minimum Value            | None                   |
| Maximum Value            | 6 characters           |
| Alternate Specifications | VolumeSerialNumber, VS |

You can use the **VOL\_SER** parameter to specify the default volume serial to use for new datasets created by the MFT Platform Server Responder. If the VOL\_SER parameter is not defined, Platform Server for z/OS uses the **VOL\_SER** that is specified in GLOBAL parameters on the z/OS system. The **VOL\_SER** parameter is ignored when sent to MFT Platform Server for Windows.

#### **WRITE\_MODE= { C | R | A | CR | CA | CN }**

|                          |                     |
|--------------------------|---------------------|
| Default Value            | CR                  |
| Allowable Values         | C, R, A, CR, CA, CN |
| Minimum Value            | Not applicable      |
| Maximum Value            | Not applicable      |
| Alternate Specifications | WriteMode, WM       |

You can use the **WRITE\_MODE** parameter to define the action on the remote file. The following table lists the allowable parameter values and their descriptions.

| Parameter Value |        | Description  |
|-----------------|--------|--|
| C               | Create | To create the remote file. If it already exists, abort the transfer. |

| Parameter Value |                    | Description  |
|-----------------|--------------------|--|
| R               | Replace            | To replace the remote file only. If it does not exist, then abort the transfer.  |
| A               | Append             | To append to the remote file.  |
| CR              | Create Replace     | To create the remote file or replace it if it already exists.  |
| CA              | Create Append      | To create the remote file or append to it if it already exists.  |
| CN              | Create Replace New | To create the remote file or replace it with new attributes. When you specify this parameter for transfers to Windows, CN indicates that the system creates directory paths as needed. |

## Use of Errorlevel with FTMSCMD

FTMSCMD passes back return codes to assist programmers in writing batch jobs.

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

```
@echo off
FTMSCMD /nologo /lu:danl1i2 /ri:ftmsusr1 /rw:ftmsspwd /rl:80 /rf:f
"c:\data\production information file1.dat" prftms.xabl.data.prodinfl
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
```

### 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.



`/NOLOGO` is used to instruct 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 skip to the area labeled `:ERROR` and perform the tasks in that area if the error level passed back from the ASNA program is 1.

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



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

For more information about how to write batch programs using `errorlevel`, see Microsoft's MS DOS documentation.

## 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
- fusing Utility
- fusutil Utility
- Nodes, Profiles, and Distribution Lists
- TIBCO Accelerator
- SSL

### Access Control

From 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.

### 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   |
|---------------|---|
| <b>USERID</b> | Defines the local user ID. Either this or <b>NODE/IPADDR</b> must be specified. Both <b>USERID</b> and <b>NODE/IPADDR</b> can be specified. A value of <b>DEFAULT</b> indicates that this is the default value for a system.  |
| <b>NODE</b>   | Defines the node definition. Either the <b>NODE/IPADDR</b> or <b>USERID</b> must be specified. Both <b>USERID</b> and <b>NODE/IPADDR</b> can be specified. A value of <b>DEFAULT</b> indicates that this is the default value for a system. This parameter is mutually exclusive with the <b>IPADDR</b> parameter. When defining nodes in this file, ensure that you use the proper case as these files are case sensitive. |

|                    |  |
|--------------------|--|
| <b>IPADDR</b>      | Defines the IP address in dotted decimal notation. Either the <b>NODE/IPADDR</b> or <b>USERID</b> must be specified. Both <b>USERID</b> and <b>NODE/IPADDR</b> can be specified. This parameter is mutually exclusive with the <b>NODE</b> parameter.  |
| <b>DESCRIPTION</b> | Defines a 32 -byte description or comment.   |
| <b>SEND_DIR</b>    | 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.  |
| <b>RECEIVE_DIR</b> | 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.  |
| <b>COMMAND_DIR</b> | 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.   |
| <b>SUBMIT_DIR</b>  | 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 <b>SUBMIT_OPTION</b> is set to ROOT or FORCE. This parameter is valid only for MFT Platform Server on z/OS. .   |
| <b>SEND_OPTION</b> | Defines the options for sending files. The valid values are as follows: <b>ROOT</b> - If a directory is specified, the directory is appended to the directory defined by the SEND_DIR parameter. <b>FORCE</b> - 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. <b>ALLOW</b> - 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. <b>REJECT</b> - If a directory is specified on a Send, the file transfer terminate s with errors. Otherwise, data is processed from the SEND_DIR directory. <b>NEVER</b> - The <b>NODE</b> , <b>USERID</b> , or <b>IPADDR</b> cannot send a file. <b>USE</b> - 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 | <p>Defines the options for receiving files. The valid values are as follows: <b>ROOT</b> - If a directory is specified, the directory is appended to the directory defined by the RECEIVE_DIR parameter. <b>FORCE</b> - 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. <b>ALLOW</b> - 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. <b>Note</b> : 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. <b>REJECT</b> - If a directory is specified on a Send, the file transfer terminates with errors. Otherwise, data is processed from the RECEIVE_DIR directory. <b>NEVER</b> - The <b>NODE</b> or <b>USERID</b> cannot receive a file. <b>USE</b> - 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.</p> |
| COMMAND_OPTION | <p>Defines the options for executing commands. The valid values are as follows: <b>ROOT</b> - If a directory is specified, the directory is appended to the directory defined by the COMMAND_DIR parameter. <b>NEVER</b> - The <b>NODE</b> , <b>USERID</b> , or <b>IPADDR</b> cannot execute commands. <b>USE</b> - 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.</p>  |
| SUBMIT_OPTION  | <p>Defines the options for submitting jobs. The valid values are as follows: <b>ALLOW</b> - The user can submit jobs. <b>NEVER</b> - The <b>NODE</b> or <b>USERID</b> cannot receive a file.</p>   |

### Directory Name Used in Request

If the directory name is defined in the **RECEIVE\_DIRECTORY** parameter and the **FORCE** parameter XE " Access Control Parameters:FORCE" 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 XE " Access Control Parameters:ROOT" 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

## Continuation and Comments

Parameters XE " Access Control Parameters:Multiple Lines" 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.

## Default Entries

You can specify default entries for the **USERID** and **NODE** parameters by using the value **DEFAULT** XE "DEFAULT" XE " Access Control Parameters:DEFAULT Value" . 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
```

## 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.

## 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.

## 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

## 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. |

| Parameter | Description   |
|-----------|---|
| 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     | <p>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.</p> <p><b>NODE/IPADDR</b> and/or <b>USERID</b> must be defined. When <b>ALLOW</b> is not defined, <b>FILE</b> and <b>ALIAS</b> must be defined. When <b>ALLOW</b> is defined, the <b>FILE</b> and <b>ALIAS</b> parameters are not supported. If a sender's parameters do not match any entry in the alias config file, the transfer is rejected.</p> |

## 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:

| Parameter | Description   |
|-----------|---|
| %JDATE    | Julian Date (YYDDD)                                   |
| %JDATEC   | Julian Date (CCYYDDD)                                 |
| %GDATE    | Gregorian Date (YYMMDD)                               |
| %GDATEC   | Gregorian Date (CCYYMMDD)                             |
| %TIMET    | Time (HHMMSSST)                                       |
| %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

## 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.

## 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=monthlysals

```

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

## 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.

## 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=8.0,LocalFileName=C:\sample\4.txt,RemoteFileName=C:\Users\vrachako\Desktop\all_samples\4.txt,Priority=N/A,LocalTranNumber=RC19700000,RemoteTranNumber=IC19700007,TransferStartTime=012048,TransferStartDate=20171219,TransferCompletionTime=012049,TransferCompletionDate=20171219,TransferEndTime=012049,TransferEndDate=20171219,TransferDirection=Receive,TransferWork=File,TransferCommand=N/A,TransferProcessName=Name,TransferScheduleDate=N/A,TransferScheduleTime=N/A,TransferExpirationDate=N/A,TransferExpirationTime=N/A,CompressionType=RLZ,CompressedBytes=1399,ConvertCRLF=no,EBCDICTranslate=no,TLS=Tunnel,EncryptionType=TLS,RecordFormat=N/A,FileCreateOptions=CreateReplace,FileAttributes=N/A,UNIXFilePermissions=N/A,AllocationType=N/A,AllocationDirectory=N/A,AllocationPrimary=0,AllocationSecondary=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=Administrator,LogonDomain=ROHINIVM6,RemoteUserid=vrachako,RemoteNodeName=MUL_HOST1,RemoteNodeType=N/A,RemotePortNumber=N/A,TryCount=0,TryMaxCount=0,ByteCount=1399,RecordCount=1,MemberCount=N/A,CheckPointCount=0,CheckPointRestart=no,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,PPA1Action=N/A,PPA1Source=N/A,PPA1Status=N/A,PPA1Data=N/A,PPA1ReturnCode=N/A,PPA2Action=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,EmailSuccessAddr=N/A,EmailFailureAddr=N/A,Accelerate=N/A,ACCProtocol=N/A,ACCEncryption=N/A,ACCCompression=N/A,ACCMaxSpeed=N/A,ACCHost=N/A,ACCPort=N/A,SecurityPolicy=None,RemoveTrailingSpaces=N,ScanSubDir=N/A,ClassOfService=Default,MaintainBDW=N/A,MaintainRDW=N/A,RetentionPeriodExpirationDate=N/A,NodeWinners=5,CRC=6071c49b,TLSProtocol=TLSv1.2,TLSCipher=AES256-GCM-SHA384,
```

## 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.

## CFINQ Parameters

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

| Parameter   | Alternate Specification | Description   |
|-------------|-------------------------|---|
| DAYS        | None                    | <p>Number of days to search</p> <p>If <b>SDATE</b> and <b>EDATE</b> are both defined, this field is ignored. <b>DAYS</b> must not exceed 1826 (5 years). If <b>SDATE</b> is not defined, the start date equals the current date minus the number of days. If <b>SDATE</b> is not defined and <b>EDATE</b> is defined, the CFINQ program starts searching at the date calculated from (<b>EDATE</b> - # of <b>DAYS</b>) and ends at the <b>EDATE</b> date.</p> |
| DESCRIPTION | DESCR                   | <p>MFT Platform Server user data</p> <p>This parameter defines the MFT Platform Server user data. When the <b>DESCRIPTION</b> 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.</p>  |
| ENDDATE     | EDATE                   | <p>End date</p> <p>The <b>ENDDATE</b> defines the end date in the format of <i>yyyymmdd</i>.</p> <p>EDATE=TOD or EDATE=TODAY means today.</p> <p>EDATE=YES or EDATE=YESTERDAY means yesterday.</p> <p>If <b>EDATE</b> is not defined, the default value is TODAY.</p>   |
| ENDTIME     | ETIME                   | <p>End time</p> <p>This parameter defines the end time in the 24 hour format of <i>hhmmss</i>. The default value is 240000. If <b>ETIME</b> is not defined, the CFINQ program searches for the MFT Platform Server transaction only within 000000 - <b>ETIME</b> period.</p>  |
| EXCEPTIONS  | EXC                     | <p>Status of the transaction</p> <p>This parameter defines the type of transfers to select.</p> <p>U = Unsuccessful</p> <p>S = Successful</p> <p>Default = Successful and Unsuccessful</p>  |
| IRFLAG      | IRF                     | <p>Initiator or responder records</p> <p>This parameter defines the type of records to select.</p> <p>B = Both</p> <p>I = Initiator</p> <p>R = Responder</p> <p>Default = Both</p>  |

| Parameter   | Alternate Specification | Description   |
|-------------|-------------------------|---|
| LOCALFILE   | LF                      | Local file name<br><br>This parameter defines the MFT Platform Server local file name.  |
| LOCALUSER   | LUSER                   | Local user I D<br><br>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 | LTRN                    | Local transaction number<br><br>The <b>LOCTRANSNUM</b> parameter defines the unique local transaction number of the MFT Platform Server transfer. When the <b>LOCTRANSNUM</b> 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 <b>LOCTRANSNUM</b> is specified.                 |
| LOGDIR      | LOGD                    | MFT Platform Server log files directory<br><br>This parameter defines the MFT Platform Server log files directory.  |
| MAXXFER     | MAX                     | Number of transactions to list<br><br>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     | PRO                     | MFT Platform Server process name<br><br>This parameter defines the MFT Platform Server process name.  |
| REMHOST     | RHOST                   | Remote system name<br><br>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 | RTRN                    | Remote transaction number<br><br>The <b>REMTRANSNUM</b> parameter defines the unique remote transaction number of the MFT Platform Server transfer. When the <b>REMTRANSNUM</b> 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. |



| Parameter | Alternate Specification | Description   |
|-----------|-------------------------|---|
| STARTDATE | SDATE                   | <p>Start date</p> <p>The <b>STARTDATE</b> defines the start date of the search in the format of <i>yyyymmdd</i>.</p> <p>SDATE=TOD or SDATE=TODAY means today.</p> <p>SDATE=YES or SDATE=YESTERDAY means yesterday.</p> <p>If <b>SDATE</b> is not defined, the default value is TODAY.</p> |
| STARTTIME | STIME                   | <p>Start time</p> <p>This parameter defines the start time in the 24 hour format of <i>hhmmss</i>. The default value is 000000. If <b>ETIME</b> is not defined, the CFINQ program searches for the MFT Platform Server transaction only within <b>STIME</b> - 24 hour period.</p>         |
| TEMPERROR | TMPERR                  | <p>Return temporary error records</p> <p>This parameter defines whether temporary error records are selected.</p> <p>Y = Yes</p> <p>N = No</p> <p>Default = Yes</p>   |

## Example of Using CFINQ Utility

This example shows how to use CFINQ XE "CFINQ Example" 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 <b>Enter</b> 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 LOCALFILENAME\DIRECTORY
*****
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 I 629500019.

```

*****
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 yyyyymmdd
      ENDDATE=     or EDATE=   Defines the End date in format yyyyymmdd
      DAYS=        or         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
*****

```

- Navigation commands are case sensitive.
- Date is returned in an FIFO (First In, First Out) order. If the number of records exceeds the value of maxxfer (default: 500), the oldest transfers are selected first. Newer 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



## 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.

## 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: <ul style="list-style-type: none"> <li>• SEND</li> <li>• RECEIVE</li> <li>• BOTH</li> </ul>  |
| SOURCE        | Defines the source of the file transfer request. The following values can be defined by the SOURCE parameter: <ul style="list-style-type: none"> <li>• INITIATOR</li> <li>• RESPONDER</li> <li>• BOTH</li> </ul>   |
| STATUS        | Defines whether a transfer request is successful or unsuccessful. The following values can be defined by the STATUS parameter: <ul style="list-style-type: none"> <li>• SUCCESS</li> <li>• FAILURE</li> <li>• BOTH</li> </ul>  |
| 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 <b>NODE</b> 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 <b>NODE</b> parameter. When defining nodes in this file, make certain that you use the proper case as these files are case sensitive. |

## Sample of CfgPostProc.cfg File

The following example is a sample Configured Post Processing file, called CfgPostProc.cfg by default.

```
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
```

## Arguments for 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                      |

## 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/EBCDIC 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
79818283848586878889919293949596
979899A2A3A4A5A6A7A8A9C04FD0A107
9F000000000000000000000000000000
00000000000000000000000000000000
41AA4AB100B26AB5BDB49A8A5FCAAFBC
908FEAFABEA0B6B39DDA9B8BB7B8B9AB
6465626663679E687471727378757677
AC69EDEEEBEFECBF80FDFEFBFCADAE59
4445424643479C485451525358555657
8C49CDCEBCFCCE170DDDEDBDC8D8EDF
002E2E2E2E2E2E2E2E0A2E2E0D2E2E
2E2E2E2E2E0A2E2E2E2E2E2E2E2E2E
2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E
2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E
20A0E2E4E0E1E3E5E7F1A22E3C282B7C
26E9EAE8E8E8E8E8E8E8E8E8E8E8E8E
2D2FC2C4C0C1C3C5C7D1A62C255F3E3F
F8C9CACBC8CDCECFCC603A2340273D22
D8616263646566676869ABBBF0FDFEB1
B06A6B6C6D6E6F707172AABAE6B8C680
B57E737475767778797AA1BFD0DDDEAE
5EA3A5B7A9A7B6BCBDBE5B5DAFA8B4D7
7B414243444546474849ADF4F6F2F3F5
7D4A4B4C4D4E4F505152B9FBFCF9FAFF
5CF7535455565758595AB2D4D6D2D3D5
30313233343536373839B3DBDCD9DA2E

```

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

## 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 |

## Definition of 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:

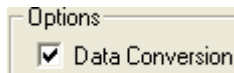
|    | x0 | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 | xA | xB | xC | xD | xE | xF |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0x | 00 | 2E | 2E | 2E | 2E | 2E | 2E | 2E | 2E | 2E | 0A | 2E | 2E | 0D | 2E | 2E |
| 1x | 2E | 2E | 2E | 2E | 2E | 0A | 2E | 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 |

## Additional Information for Data Conversion

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 the 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.

## 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.

### 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).



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 or the first day of every month).
- DNI provides store and forward capabilities where the DNI scan directory is the destination of a platform server transfer. DNI scan s the directory and forward s 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 complete d 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](#)).



- For optimum performance, if you have more than 50 DNI Scan directories, you should install 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 transfer red and which file transfer failed.

## Transfer Template

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.

## Creating a transfer template

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

### Procedure

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.

### Advanced TCP Template Definition

The TCP template property pages are similar to the transfer property pages. The following template property tabs enable you to define the advanced TCP template:

- Template Tab:

| Fields  | Description   |
|---------|---|
| Name    | The name used to identify this template. Make sure 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 bytes. |

- Transfer Tab

See [Transfer Tab](#) section for more information on the parameters of this tab.

- Notify Tab

See [Notify Tab](#) section for more information on the parameters of this tab.

- Advanced Options Tab

See [Advanced Options Tab](#) section for more information on the parameters of this tab.

- Expiration Tab

See [Expiration Tab](#) for more information on the parameters of this tab.



The expiration date option is not usable for templates.

- Post Processing Action Tab

See [Post Processing Action Tab](#) section for more information on the parameters of this tab.

- TCP/IP Tab

See [TCP/IP Tab](#) section for more information on the parameters of this tab.

- Accelerator Tab

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

### Advanced Batch Template Definition

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.



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

A screenshot of a Windows-style dialog box titled 'Batch Template'. It has three tabs: 'Template', 'Batch Job', and 'Notify'. The 'Template' tab is selected. Inside the dialog, there is a 'General' section containing two text input fields: 'Name' with the value 'BatchTemplateName' and 'Comment' with the value 'Your comments'. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

| Field   | Description  |
|---------|--|
| 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



The image shows a Windows-style dialog box for configuring a batch job. It has three tabs at the top: 'Template', 'Batch Job', and 'Notify'. The 'Batch Job' tab is selected. Inside the dialog, there are two main sections. The first section, titled 'Batch Job', contains a single text field labeled 'Local Job' which contains the text '\$(LocalFilePath)\\$(LocalFileName)'. The second section, titled 'Local Identification', contains two text fields: 'User ID' and 'Password'. At the bottom right of the dialog are two buttons: 'OK' and 'Cancel'.

### 1. 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).

### 2. Local Identification

The user credentials provided in the fields are used to run the batch templates.

### 3. Notify Tab



a. **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.

## 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 [File Name Token List](#).

## 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.



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.

## 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.



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.



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
- 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.



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.

## 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**

Put s 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**

Schedule d 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.



To apply the information in th es e field s 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.

## 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.



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: [perl.org](http://perl.org).

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

## fusping Utility

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

### 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
```

## Examples of Using fusing Utility

The following examples show how to use the fusing 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:

```
fusing 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:

```
fusing 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)
```

## 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.

| Function | Shortcut | Windows Equivalent Command |
|----------|----------|----------------------------|
| RENAME   | R        | move                       |

| Function | Shortcut | Windows Equivalent Command |
|----------|----------|----------------------------|
| DELETE   | D        | erase                      |
| EXIST    | E        | N/A                        |

m command.

## 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  | Short cut | Description                                 |
|-----------|-----------|---|
| RENAME    | R         | Rename s a file.                            |
| DELETE    | D         | Delete s a file.                            |
| EXIST     | E         | Verifies whether a file exists.             |
| RENAMEDIR | RDIR      | Rename s a directory.                       |
| DELETEDIR | DDIR      | Delete s a non-empty directory recursively. |
| REMOVEDIR | RMDIR     | Removes an empty directory only.            |
| EXISTSDIR | EDIR      | Verifies whether a directory exists.        |

## Examples of Using fusutil Utility

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 EXISTS DIR <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>
```



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

## Special Processing

When processing the EXIST function, the platform server also checks whether the file is available for use (that is, if the file is being used by another application). This is done on all platforms except UNIX, because no standard call is available to accomplish this on UNIX.

## 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).

## 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 system s. 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 XE "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



Distribution Lists support send requests only.

The MFT Platform Server configuration file `cflist.cfg` is located in the MFT Platform Server installation directory.



## 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
ResponderProfil = N
Description     = Sample TCP node
```

```
CommandSupport    = PING
Winners           = 2
```

## 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.



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)                      | <p>The node parameter is used to specify the name of the node to be added or updated to the <code>cfnode.cfg</code> file. The node name can be up to 256 characters long and cannot contain any spaces.</p> <p>Examples:</p> <pre>node:dataserver n:dataserver</pre>   |
| systemType (s)                | <p>The systemType parameter is used to specify the type of system represented by this node definition. The valid system types are as follows:</p> <ul style="list-style-type: none"> <li>• HPUX</li> <li>• SUN/SOLARIS</li> <li>• AIX</li> <li>• LINUX</li> <li>• Windows</li> <li>• IBMi</li> <li>• z/OS</li> <li>• Command_Center</li> <li>• Other</li> </ul> <p>Examples:</p> <pre>systemType:Windows s:Windows</pre> |
| hostname (h)                  | <p>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. Now this parameter allows multiple IP Name/IP Address values to be specified for the responder to use in node lookup.</p> <p>Examples:</p> <pre>hostname:11.22.33.44,11.22.33.55 h:computer.domain.com</pre>   |



| Required Parameter (Shortcut) | Description   |
|-------------------------------|---|
| port (p)                      | <p>The port parameter is used to specify the port number on which the remote node is listening.</p> <p>Examples:</p> <pre>port:46464</pre> <pre>p:46464</pre> |

See the following table for the optional node parameters.



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)                    | <p>The action parameter is used to specify the action to be taken. The valid values are : Delete, List, and Add.</p> <p>Example:</p> <pre>action:delete n:NYNode</pre>  |
| netmask (net)                 | <p>This is the netMask for the remote I Pv4 address.</p> <p>Examples:</p> <pre>netMask:255.255.255.0 net:255.255.255.128</pre>  |
| netMask6 (net6)               | <p>This is the netMask for the remote I Pv6 address . This is a number between 8 and 128 and a multiple of 8.</p>   |
| ssl                           | <p>The ssl parameter is used to specify whether SSL/TLS is used for TCP/IP communications.</p> <p>Example:</p> <pre>ssl:Y</pre> <ul style="list-style-type: none"> <li>• N: TLS is not used</li> <li>• Y: TLS is used</li> <li>• T: TLS Tunnel is used</li> </ul> |

| Optional Parameter (Shortcut) | Description   |
|-------------------------------|---|
| compress (c)                  | <p>The compress parameter is used to specify the default compression type for all transfers with this node. The valid values are as follows:</p> <ul style="list-style-type: none"> <li>• LZ</li> <li>• RLE</li> <li>• ZLIB1 - ZLIB9</li> <li>• NO - No default compression</li> <li>• EVER - Never use compression</li> <li>• N</li> </ul> <p> NEVER is the only option that cannot be overridden by options on the command line.</p> <p>Examples:</p> <pre>compress:LZ c:NEVER</pre>   |
| encrypt (e)                   | <p>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:</p> <ul style="list-style-type: none"> <li>• DES</li> <li>• 3DES - Triple DES</li> <li>• BF - Blow fish Encryption</li> <li>• BFL - Blow fish Long</li> <li>• AES - Rijndael</li> <li>• NO - No encryption</li> <li>• NEVER - Never use encryption</li> </ul> <p> NEVER is the only option that cannot be overridden by options on the command line.</p> <p>Examples:</p> <pre>encrypt:DES e:NEVER</pre> |
| security (sl)                 | <p>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 <a href="#">General Tab</a>, System Configurations: Security Policy for more information.</p> <p>Example:</p> <pre>security:HIPAA</pre>   |
| lct                           | <p>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.</p> <p>Example:</p> <pre>lct:convert.txt</pre>   |

| Optional Parameter (Shortcut) | Description  |
|-------------------------------|--|
| rct                           | <p>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.</p> <p>Example:</p> <pre>rct:convert.txt</pre>   |
| responder (r)                 | <p>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.</p> <p>Example:</p> <pre>responder:Yes r:Y</pre> |
| description (d)               | <p>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 cfnod command does not require the description parameter defined if the prompt:NO parameter is supplied.</p> <p>Examples:</p> <pre>description:"This is a sample description"</pre> <pre>d:"This is a sample description"</pre>  |

| Optional Parameter (Shortcut) | Description  |
|-------------------------------|--|
| commandsupport (ccc)          | <p>It defines the actions that Command Center can perform on this node. The valid values are as follows:</p> <ul style="list-style-type: none"> <li>• ALL - NODE, PROFILE, AUDIT, ALTER, PING, and TRANSFER are supported on this node.</li> <li>• NONE - No Command Center function is supported on this node. This is the default setting if the parameter is not defined.</li> <li>• AUDIT - This node supports requests that inquire on the MFT Platform Server audit file.</li> <li>• NODE - Node List and Update functions are supported on this node.</li> <li>• PING - MFT Platform Server fusing requests are supported on this node.</li> <li>• PROFILE - Profile list and update functions are supported on this node.</li> <li>• TRANSFER - This node supports the Command Center Transfer function that initiates file transfers.</li> </ul> <p>Examples:</p> <pre>ccc:PING commandsupport:TRANSFER</pre> |
| winners (win)                 | <p>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.</p> <p>This parameter only works for transfers submitted to the platform server. It does not work for command line transfers.</p> <p>Example:</p> <pre>Winners:2</pre>   |
| prompt                        | <p>The prompt parameter is used to put cfnode into an interactive mode. If <b>prompt: YES</b> 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 <b>prompt: NO</b>.</p>  |
| -?                            | <p>The -? parameter is used to display the online help for cfnode.</p>   |

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.

```

## 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               |

## Profile Definitions

Profile definitions are stored in files named `cfprofile.cfg` (User Profiles) and `cfprofile.cfg` (Responder Profiles) located in the MFT Platform Server directory.

You can add and update local user profiles 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`.

You can specify the responder profile parameters in the **cfprofile** command line.

## 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. Use the **cfprofile** command to create, delete and update a user profile. When a node is supplied in a transfer, a user profile is chosen for the node based on the current log on 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:

| Required Parameter (Shortcut) | Description   |
|-------------------------------|---|
| node ( n )                    | Node with which the Local User Profile is associated.               |
| lUser (l)                     | Local user name who can use this profile.                           |
| rUser (r)                     | Remote user name to use to log on to the node.                      |
| rPass (rp)                    | Remote password to use to log on to the node (in encrypted format). |

## 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
```


## Responder 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 **cfprofile** command.

See the following table for the required responder profile parameters.



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 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 <code>cfnode.cfg</code> to successfully add or update a responder profile.<br><br>Example: <code>node:dataserverA n:dataserverB</code>                               |
| 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: <code>password:apple p:computer</code>   |
| 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.<br><br>Example: <code>lUser:john l:john</code>   |
| rPass (rp)                    | The rPass parameter is used to specify the remote password that is sent by the remote MFT Platform Server initiating the transfer.<br><br> If this responder profile is to be in conjunction with an already verified user, rPass must be set to *VER.<br><br>Example: <code>rPass:apple rp:*VER</code> |
| 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.<br><br>Example: <code>rUser:kenny r:kenny</code>  |

See the following table for the required responder profile parameters.



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. Example:<br><br>action:delete<br><br>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 cf rprofile.   |

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.
```

### 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

```

## Distribution Lists

Distribution lists define multiple nodes and a default directory to which you can perform sendtransfers. 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.



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

```

## 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 name s 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. |

| Parameter | Description   |
|-----------|---|
| 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. |

## 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).

## 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.



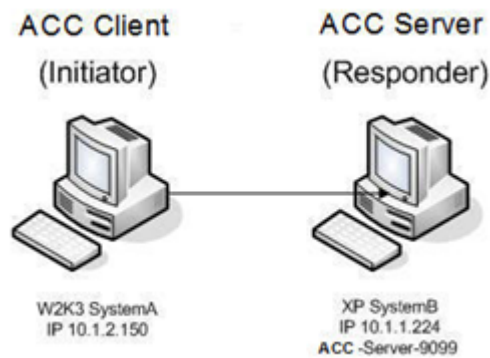
Ports 9000, 9002, and 9100 - 9199 must be opened in the firewall to allow TIBCO Accelerator Client to access TIBCO Accelerator Server. If requests are initiated from an external computer, these ports must be opened on the firewall for incoming traffic. If requests are initiated from an internal computer, these ports must be opened on the firewall for outgoing traffic.

## Usage of 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).

## 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.



## Procedure

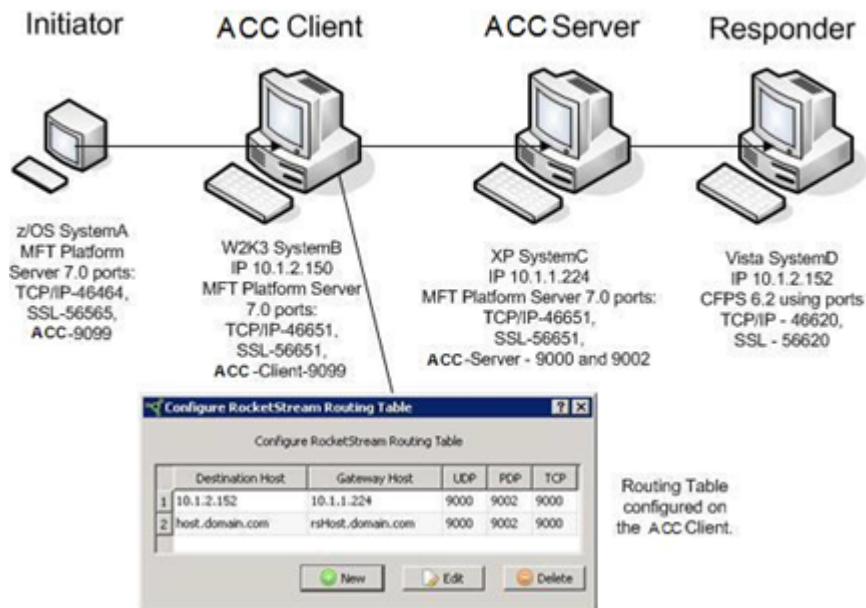
1. Verify SystemB has the TIBCO Accelerator service running and is listening on the default port 9099.
  1. To verify this, open the MFT Platform Server Administrator window on SystemB, and display the server properties.
  2. Click the **Accelerator** tab.
2. On SystemA, set an advanced TCP transfer by filling in the necessary transfer detail information in the various Transfer Property tabs as seen in [Transfer Properties](#), except for when you get to the Accelerator tab as seen below:

The screenshot shows the 'Accelerator' configuration window. At the top, the 'Accelerate' checkbox is checked. Below it, the 'Properties' section contains three text fields: 'Host' with the value '10.1.2.150', 'Port' with the value '9099', and 'MaxSpeed (kbps)' with the value '1000000'. The 'Protocol' section has three radio buttons: 'TCP', 'UDP', and 'PDP', with 'PDP' selected. The 'Options' section has two checkboxes, 'Encryption' and 'Compression', both of which are unchecked. The 'Compression' dropdown menu is set to 'No'.

By default, the TIBCO Accelerator parameters are grayed out. To have this transfer be sent via TIBCO Accelerator, you must select the **Accelerate** checkbox. 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 section. In this example, the default values are used for all other fields on this screen.

3. After your transfer details are completed, click **OK** at the bottom of your Transfer Properties window. Your file is now sent using TIBCO Accelerator.

## 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.

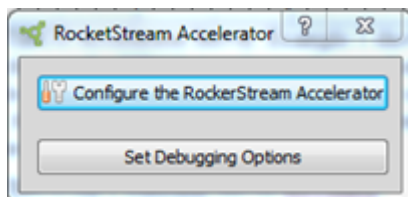
When conducting TIBCO Accelerator transfers of this kind, you must configure a TIBCO Accelerator Routing Table on the TIBCO Accelerator Client that the Platform Server Initiator connects to. The TIBCO Accelerator Client needs the connectivity information for the destination TIBCO Accelerator Server that connects to the Platform Server Responder.

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.

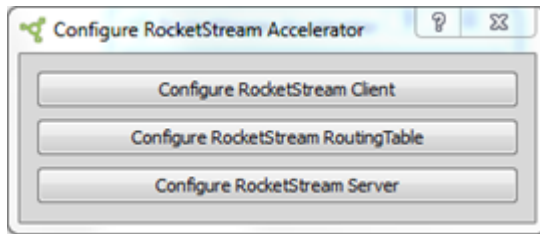
## Procedure

1. To configure the routing table, open Windows Explorer and navigate to the following folder:  
`<PlatformServer_Install>\TIBCO\MFT Platform Server\RSTunnel\`
2. Double-click the file `RSTunnelConfig.exe`. The following window is displayed:

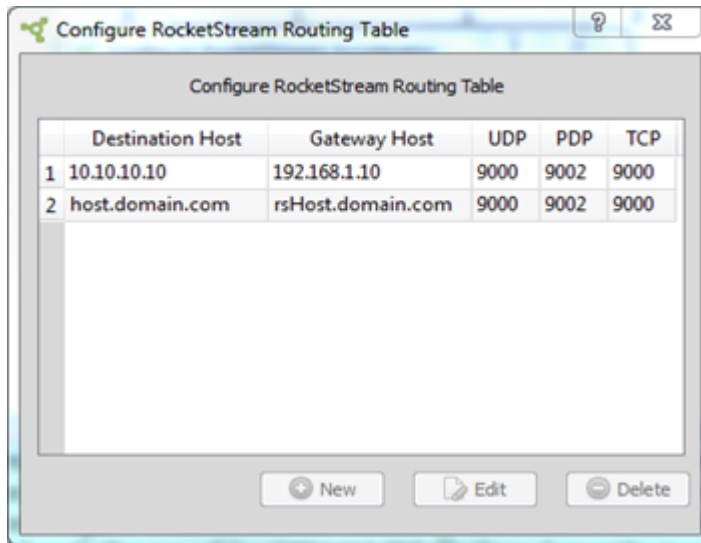


3. Click **Configure the RocketStream Accelerator** (Set Debugging Options can only be used when instructed by TIBCO Technical Support.) The following window is displayed:

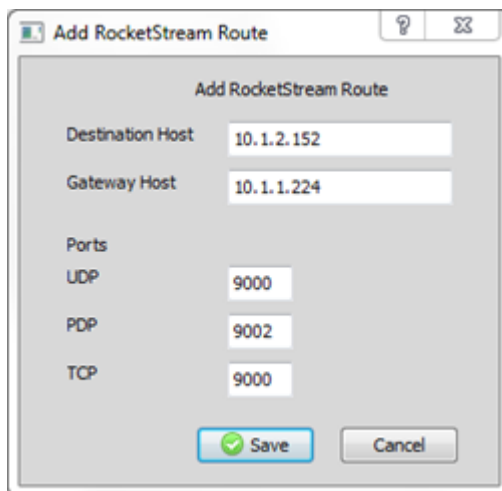




4. Click **Configure TIBCO Accelerator Routing Table**. You can see an example setup.



5. Select the first row. Click **Edit**. The following window is displayed:

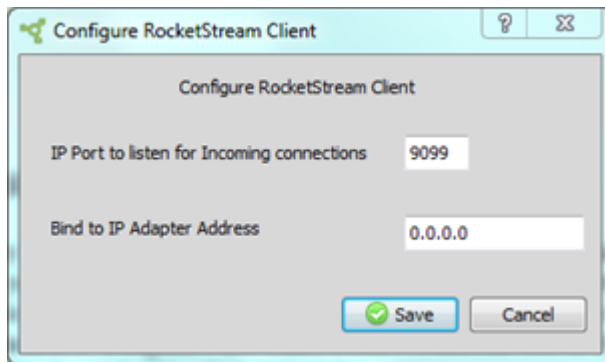


- Define the **Destination Host** (the IP of the server that is the final destination for your file being transferred: the Responder) and the **Gateway Host** (the remote TIBCO Accelerator Server that is initially receiving your file transfer before passing it off to the Destination Host).
- If required, 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 Save.



You are then presented with a warning that the RocketStream Tunnel service must be restarted for the changes to take effect.

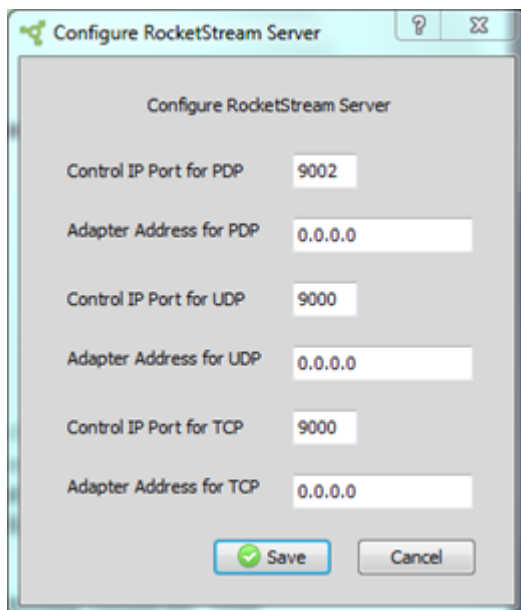
- d) Stop and start TIBCO Accelerator from your Server Properties window or you can open your Services window and restart MFT Platform Server.
  - e) Close the Configure RocketStream Routing Table window.  
The routing table is configured on the TIBCO Accelerator Client defining what ports is used when sending files with the various protocols TIBCO Accelerator offers.
6. If multiple network cards are available, at this time, define the port and IP address your Client binds to. By default, the client listens on port 9099. If you must change the default port number, click **Configure RocketStream Client** (see figure in Step3). The following window is displayed:



7. If your server has multiple network cards, define the IP adapter address you want the TIBCO Accelerator Client to bind to; otherwise, leave the **Bind to IP Adapter Address** field.
8. When you are done, click **Save** and close the window.  
If your TIBCO Accelerator Client would ever be switching roles and acting as a TIBCO Accelerator Server in the future, configure the TIBCO Accelerator Server ports and IP address to bind to at this time by clicking the **Configure RocketStream Server** button (see figure in Step3). The following window is displayed:



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.

## 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.

### SSL or TLS Transfers

All SSL or TLS 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.

Two TCP ports are available for TLS, one for IPv4 requests and one for IPv6. The TLS port is optional. Entering 0 as the TLS port number disables TLS.

Two TCP ports are available for TLS Tunnel, one for IPv4 requests and one for IPv6. The TLS port is optional. Entering 0 as the TLS port number disables TLS. Valid range is the same as for other ports supported, between 1025 and 65535.

Only one regular TCP port can be configured during installation. 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 or TLS 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.

## 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. T



he bit strength must meet the requirements of CA.

## Certificate Creation

The following menu depicts the 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.



The name s 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.

## Certificate View

To view a certificate, select 2 from the SSL Utilities menu.

|   |
|---|
| SSL Utilities Menu<br>1. Generate a Certificate Request<br>2. View a Certificate<br>3. Exit<br>Please enter your choice: 2<br>View Certificate Menu<br>Please enter the Certificate Filename:<br>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
```

```

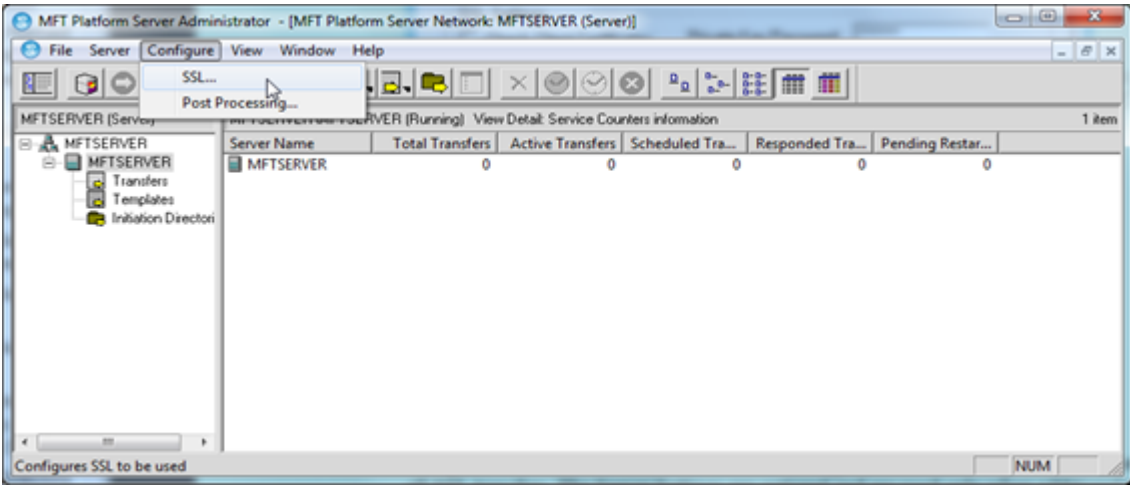
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 7 (0x7)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=US, O=TIBCO, OU=TIBCO Local CertAuth
    Validity
      Not Before: Aug 13 00:00:00 2005 GMT
      Not After : Aug 13 23:59:59 2006 GMT
    Subject: C=US, ST=NY, L=Garden 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-----
DXMxCzAJBgNVBAGTAm55QMowCAYDVQQHEwFnMQowCAYDVQQKEwFwMQowCAYDVQQLEwFwMQowCAYDVQQDEwFqMRA
wDgYJKoZIhvcANQkBFgFqMIGfMA0GCSq
GSIB3DQEBQUAA4GNADCBiQKBgQCuaviVFGeDs0R03ppz89Dm2o3SymEwCd3Q3aS8I8T8+lWgd6JMJKIrsFo5QYo
JBGeP5eCro/moIl11sWBugupvWs9h1i73Nr
l2nk5t9QIDAQABo4GQMIGNMESGCWCGSAGG+
9yIE9TLzMSMCAoUkFDRikwHQYDVR0OBBYEFcZEDuSs4i2f4+xfMmdTsGrU6zbzMB8GA1UdIwQYMBaAFEJ3oseu
PaVHXDD/T1G4z
+2s0Zw6MA0GCSqGSIB3DQEBBQUAA4GBAJ99vWbxXSzPXcXMqhYW5VKuBILRZsbFAwoZZsHSyTBNpIXJkXl5sGH7
iGFEPiH60piFuN/Fd+
ruxoMojel/HCFsJlPQK6VIR5LkDA2
-----END CERTIFICATE-----

SSL Utilities Menu
1. Generate a Certificate Request
2. View a Certificate
3. Exit
Please enter your choice:

```

## SSL Configuration

To configure TIBCO MFT Platform Server for SSL, open the SSL Settings dialog by selecting **Configure > SSL...** from the menu bar.

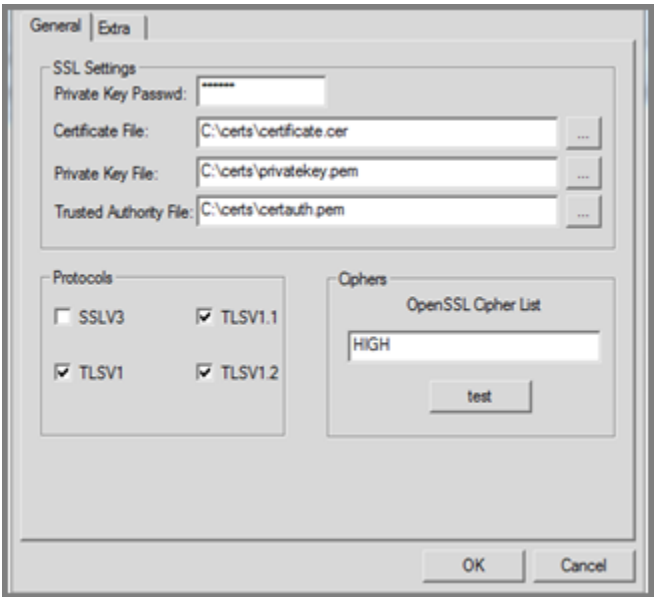


SSL Settings


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.

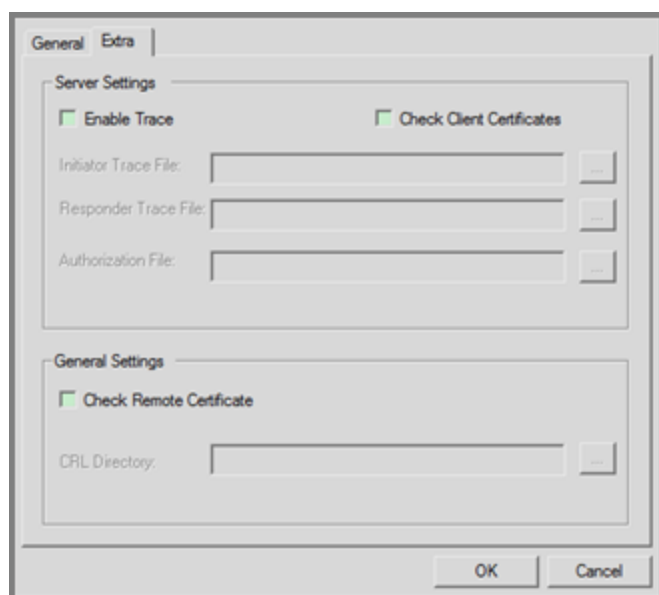
General tab



| Field                | Description  |
|----------------------|--|
| 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. |

| Field                                     | Description  |
|---|--|
| 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              | <p>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.</p> <p> To perform SSL transfer successfully, you must use the same cipher suite for the server and client.</p> |

### Extra tab





| Field                     | Description   |
|---------------------------|---|
| 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 <a href="http://www.ietf.org/rfc/rfc3280.txt">http://www.ietf.org/rfc/rfc3280.txt</a> .  |
| CRL Directory             | Defines the path where the CRL checking looks for the hashed file names.  |

## Using SSL/TLS Transfer

### Procedure

1. Open your Advanced TCP transfer window.
2. Set your transfer. See the following example:

The screenshot shows the 'Advanced Options' tab of a file transfer configuration window. The 'Destination' is set to 'host.domain.com'. Under 'Remote Identification', the 'User ID' is 'RemoteUser' and the 'Password' is masked with '\*\*\*\*'. Under 'Local Identification', the 'User ID' is 'MFTSERVER\tw' and the 'Password' is masked with '\*\*\*\*'. The 'Options' section includes checkboxes for 'Data Conversion', 'Convert CR/LF', 'Check Point/Restart', 'Compression', and 'Encryption', all of which are currently unchecked. The 'File to File' tab is selected, and the transfer direction is set to '1. Send'. The 'Local' file path is 'c:\outgoing\file.bdt' and the 'Remote' file path is 'user1.pds.files\file1'. The 'Create Option' is set to 'Create Replace'. The 'File Attributes' section includes checkboxes for 'System', 'Hidden', 'Archive', 'Read Only', and 'NTFS Compressed', all of which are unchecked. The 'Unix Permissions' field is empty. The 'OK' and 'Cancel' buttons are at the bottom right.

3. After completing the Transfer configuration, go to the **TCP/IP** tab.
4. Configure the SSL/TLS Tunnel port used by the remote server of MFT Platform Server and select the corresponding secure protocol from the drop-down list.

|            |                        |             |                  |
|------------|------------------------|-------------|------------------|
| Transfer   | Schedule               | Notify      | Advanced Options |
| Expiration | Post Processing Action | Accelerator | TCP/IP           |

Port Number

Secure Protocol

Class Of Service

CRC Checking

- After the transfer configurations are completed, click **OK** and the transfer request runs.

## 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:\tibco\MFT Platform Server | Sslauth   |
| UNIX     | \$CFROOT/config              | SSLAAuth  |



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.

| Parameter | Description   |
|-----------|---|
| /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 <i>ccyyymmdd</i> . 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 <i>hhmm</i> . 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 <i>ccyyymmdd</i> . 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.  |

| Parameter | Description   |
|-----------|---|
| /ETIME    | Defines the End time for the certificate in the format of <i>hhmm</i> . 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**

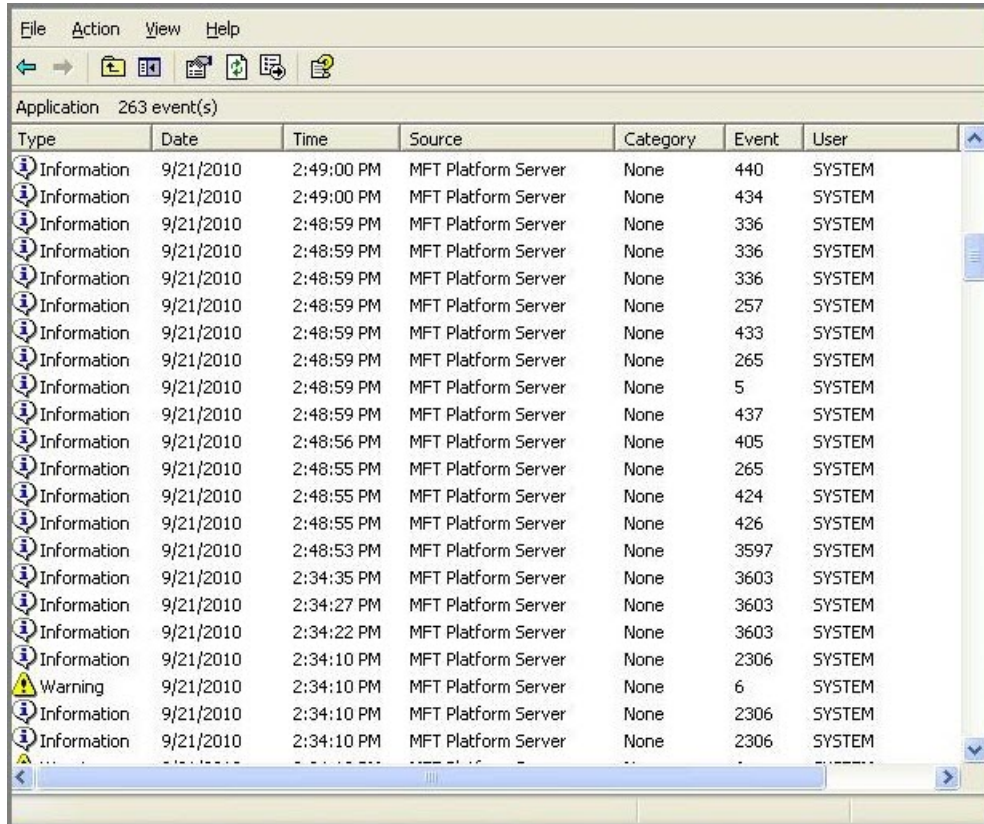
MFT Platform Server accepts all certificates that match the information defined by the /CN, /L, /ST, /C, /OU, and /O parameters. The certificate is valid from 1 December 2005 until 30 November 2006. If the certificate is received before 1 December 2005 or after 30 November 2006, the request is rejected. All other certificates not matching these criteria are rejected.

## Event Logs

Event logs are used to record and trace events in your system.

You can use Event Viewer to monitor events in your system. Event Viewer maintains logs about system, security, and application events. You can view and manage the event logs using Event Viewer. The event logging service starts automatically when you start Windows. To terminate the service, use the Services tool in the Control Panel.

Event Viewer is located in the Administrative Tools panel in Program Manager. To view a log, double-click the **Event Viewer** icon. The following figure shows a sample application log.



| Type        | Date      | Time       | Source              | Category | Event | User   |
|-------------|-----------|------------|---------------------|----------|-------|--------|
| Information | 9/21/2010 | 2:49:00 PM | MFT Platform Server | None     | 440   | SYSTEM |
| Information | 9/21/2010 | 2:49:00 PM | MFT Platform Server | None     | 434   | SYSTEM |
| Information | 9/21/2010 | 2:48:59 PM | MFT Platform Server | None     | 336   | SYSTEM |
| Information | 9/21/2010 | 2:48:59 PM | MFT Platform Server | None     | 336   | SYSTEM |
| Information | 9/21/2010 | 2:48:59 PM | MFT Platform Server | None     | 336   | SYSTEM |
| Information | 9/21/2010 | 2:48:59 PM | MFT Platform Server | None     | 257   | SYSTEM |
| Information | 9/21/2010 | 2:48:59 PM | MFT Platform Server | None     | 433   | SYSTEM |
| Information | 9/21/2010 | 2:48:59 PM | MFT Platform Server | None     | 265   | SYSTEM |
| Information | 9/21/2010 | 2:48:59 PM | MFT Platform Server | None     | 5     | SYSTEM |
| Information | 9/21/2010 | 2:48:59 PM | MFT Platform Server | None     | 437   | SYSTEM |
| Information | 9/21/2010 | 2:48:56 PM | MFT Platform Server | None     | 405   | SYSTEM |
| Information | 9/21/2010 | 2:48:55 PM | MFT Platform Server | None     | 265   | SYSTEM |
| Information | 9/21/2010 | 2:48:55 PM | MFT Platform Server | None     | 424   | SYSTEM |
| Information | 9/21/2010 | 2:48:55 PM | MFT Platform Server | None     | 426   | SYSTEM |
| Information | 9/21/2010 | 2:48:53 PM | MFT Platform Server | None     | 3597  | SYSTEM |
| Information | 9/21/2010 | 2:34:35 PM | MFT Platform Server | None     | 3603  | SYSTEM |
| Information | 9/21/2010 | 2:34:27 PM | MFT Platform Server | None     | 3603  | SYSTEM |
| Information | 9/21/2010 | 2:34:22 PM | MFT Platform Server | None     | 3603  | SYSTEM |
| Information | 9/21/2010 | 2:34:10 PM | MFT Platform Server | None     | 2306  | SYSTEM |
| Warning     | 9/21/2010 | 2:34:10 PM | MFT Platform Server | None     | 6     | SYSTEM |
| Information | 9/21/2010 | 2:34:10 PM | MFT Platform Server | None     | 2306  | SYSTEM |
| Information | 9/21/2010 | 2:34:10 PM | MFT Platform Server | None     | 2306  | SYSTEM |

### Viewing the Event Log

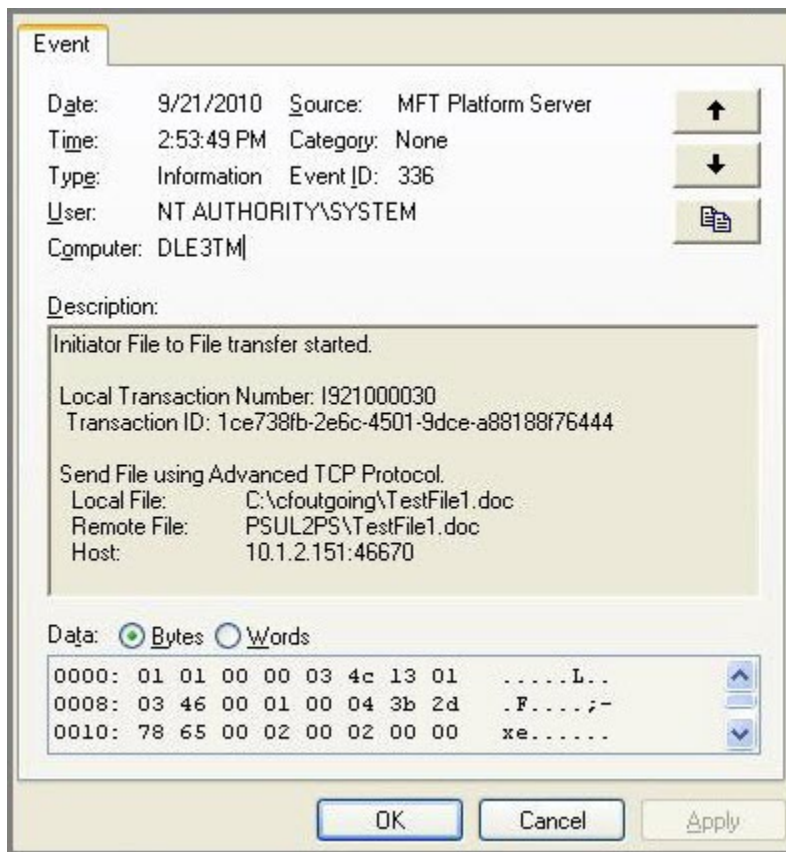
You can view three types of event logs: application, system, and security. To select the log to be displayed, click a log type on the **Log** menu.

Events displayed in Event Viewer are listed in sequence by date and time of occurrence. You view the events from newest to oldest (by default) or from oldest to newest.

MFT Platform Server writes events to the event log in both successful and unsuccessful cases. The Server also writes an information event when the transfer begins. An information event is also logged when the MFT Platform Server for Windows starts. In the event that the MFT Platform Server service has been stopped, there are no messages for any transfers that were active in the event log on the machine where the services were stopped.

To view a more detailed description of an event, double-click it. The Event Properties dialog opens.





## Event IDs and Transaction IDs

When MFT Platform Server writes an event 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. The transaction ID assigned is unique for all machines.

If a transaction is displayed in an 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 is not 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, there are three additional types of information provided on the Event Properties dialog: 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. The information includes the remote file name, local file name, transfer direction, and so on. Following the message information is information about 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, MFT Platform Server retries the transfer if the Try Count is set to a value greater than one. If MFT Platform Server retries the transfer more than once, the retry information states the number of times that the transfer is attempted before it completes successfully or unsuccessfully.

## Severity 1 Errors

Since MFT Platform Server can retry scheduled transfers that have failed, it does not retry a severity 1 error. Severe errors repeatedly fail.

The following errors are classified as Severity 1:

- Could not open the source file.

- The name is incorrectly formatted
- The volume name is incorrectly formatted
- The path is non-existent
- Could not 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
  - The IP Port is defined incorrectly

## Clearing an Event Log

When you receive a message that indicates that the event log is full, you must clear the log. You can clear an event log using either of the following ways:

- [Emptying the Current Log](#)
- [Replacing the Old Event with a New Event](#)

### Emptying the Current Log

Empty the current log to clear an event log.

#### Procedure

1. Switch to the log whose events you would like to clear.
2. From the **Log** menu, click **Clear All Events**.  
You are given options to save the currently logged events.
  - If you select to archive the events, you must select a file name and select the directory in which you want to store the log.
  - If you select not to save the events, Event Viewer empties the current log.

### Replacing the Old Event with a New Event

Replace the old event with a new event to clear an event log.

#### Procedure

1. From the **Log** menu, click **Log Settings**.  
The Event Log Settings panel opens.
2. Select **Overwrite Events as Needed**.  
When you select this option, each new event replaces the oldest event, even if the log is full.



## Cached Passwords

If you are a remote user, you can use cached Windows passwords to specify a password for a particular remote Windows User ID.

Since the passwords are stored in the Windows registry, you can perform MFT Platform Server transfers on Windows without specifying the password. You can easily manage the cached password from the remote end as needed.

To enable the cached password feature, you use a special set of tokens in the remote password field on the initiating MFT Platform Server partner. There are four types of tokens:

- X: password
- X:
- X:DELETE
- 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.

### **X:password**

Use the *X:password* token to set a password on the remote Windows system. As part of a file transfer, put x: in front of your password in the remote password field. The password is your Windows password.

When 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, a transfer performs.

### **Given X: without a password**

Use the X: token to instruct MFT Platform Server for Windows to look up the password in the registry based on your user ID. If the password is found, it is decrypted and used to log in to the Windows system. The transfer then performs. This token works from any of the remote MFT Platform Server systems.

### **X:DELETE**

Use the X:DELETE token to instruct MFT Platform Server to retrieve the cached password and decrypt it. The password is saved from a prior transaction for your user ID. You can use the password to log in to Windows to conduct a transaction and then delete the cached password from the registry. For any future transactions, you can either specify a password at logon time or use the *X:password* token to set a cached password on the Windows system.

### **X:DELETEALL**

Use the X:DELETEALL token to instruct MFT Platform Server to retrieve the cached password and decrypt it. The password is saved from a prior transaction for your user ID. You can use the password to log in to Windows to conduct a transaction and then delete all the cached passwords from the registry.

Use the *X:password* token to set or change a cached password on the Windows system. If your Windows password changes, you must delete the old password and create a new one. Simply use the *X:newpassword* token again to overwrite the old cached password.



The cached password feature is supported only on Windows. If you send over *X:password* on the z/OS side, z/OS interprets the full string as the password.

### **Restrictions**

The cached password feature has the following restrictions:

- The service must be running with System Authority.
- Since the X: token is contained within the password field, MFT Platform Server, which normally supports 20-character remote passwords, is limited to 18 characters.
- Passwords that could otherwise contain X:, X:*text*..., X:DELETE, or X:DELETEALL are accepted as triggers to the feature and not as legitimate Windows passwords.
- Since the passwords are saved in a restricted area of the registry, the uninstall program cannot delete them. You must use the X:DELETEALL token to remove the passwords before using the uninstall program. Otherwise, the \\HKEY\_LOCAL\_MACHINE\\SOFTWARE\\TIBCO registry key is not removed.

### Cached Password Example

A user with user ID being MARY wants to create a batch transfer to a remote Windows system. However, the user does not want everyone to know the password.

The user uses the X:*password* token 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
```

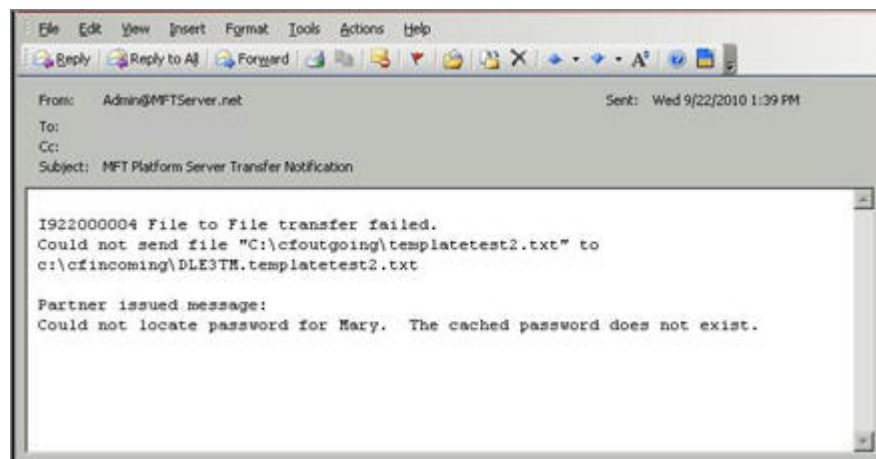
For all future transfers, the user can use the X: token instead of specifying a password. The following batch program is used for future transfers.

```
SET HOST=Fusion
SET PORT=46464
SET REMOTE_USER_ID=MARY
SET REMOTE_PASSWORD=X:
SET PROCESS_NAME=FTMS
```



The password field and the tokens are case-sensitive. If the password is lowercase, then the user needs to type X:pswdmary.

If the password is not yet cached, the following information is displayed.



# File Name Tokens

MFT Platform Server supports the File Name Tokens feature.

A string of tokens contains a character mixture of literal and substitution parameters. Given a string of tokens, MFT Platform Server generates a formatted file name. You can use the file name 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 MFT Platform Server for Windows.

## 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  |
| YYYY   | Year          | 000 - 999 (last 3 digits of year)  |
| YY     | Year          | 00 - 99 (last 2 digits of year)  |
| Y      | 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  |

| Token             | Definition   | Generated Value (Examples)  |
|-------------------|--|---|
| 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             | 24 Hour  | 01 - 12   |
| SH12              | 24 Hour  | 1 - C   |
| Sh12              | 24 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   |
| Swwww             | 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   |
| AllocationPrimary | The primary allocation size used in a file transfer. | Local file: c:\source\testfile1.txt<br>Remote file: CFUSR.F\$(AllocationPrimary).TEST<br>Token resolves to: CFUSR.F800.TEST |

| Token               | Definition   | Generated Value (Examples)   |
|---------------------|--|--|
| AllocationSecondary | The secondary allocation size used in a file transfer.                 | Local file: c:\source\testfile1.txt<br>Remote file: CFUSR.F\$(AllocationSecondary).TEST<br>Token resolves to: CFUSR.F500.TEST                  |
| AllocationType      | The allocation type used in a file transfer.                           | Resolves to: Tracks, Blocks, Cylinders, Megabytes, Kilobytes   |
| BlockSize           | The block size used in a file transfer.                                | Local file: c:\source\testfile1.txt<br>Remote file: CFUSR.F\$(BlockSize).TEST<br>Token resolves to: CFUSR.F6,160.TEST                          |
| CheckPointInterval  | The check point used in a file transfer.                               | Local file: c:\source\testfile1.txt<br>Remote file: d:\target\test\$(CheckPointInterval).txt<br>Token resolves to: d:\target\test5 minutes.txt |
| Compression         | The compression used in a file transfer.                               | LZ, RLE, or NO   |
| ComputerName        | The initiator computer name.   | Local file: c:\source\testfile1.txt<br>Remote file: d:\target\\$(ComputerName).txt<br>Token resolves to: d:\target\SYSTEM3032.txt              |
| CrLf                | Whether a carriage return line feed (CRLF) is used in a file transfer. | TRUE, FALSE  |
| DataClass           | The data class used in a file transfer to z/OS.                        | Local file: c:\source\directory\testfile1.txt<br>Remote file: PRJOE.\$(DataClass).FILE1<br>Token resolves to: PRJOE.DTCLS3.FILE1               |
| DataType            | The data type used in a file transfer.                                 | BINARY, EBCDIC   |
| Date1               | The days date formatted as YYYYMMDD.                                   | Local file: c:\source\test.txt<br>Remote file: d:\target\\$(Date1)\test.txt<br>Token resolves to: d:\target\20110809\test.txt                  |

| Token            | Definition   | Generated Value (Examples)   |
|------------------|--|--|
| Date2            | The days date formatted as MMDDYYYY.   | Local file: c:\source\test.txt<br>Remote file: d:\target\[Date2]\test.txt<br>Token resolves to: d:\target\08092011\test.txt  |
| Date3            | The days date formatted as DDMMYYYY.   | Local file: c:\source\test.txt<br>Remote file: d:\target\[Date3]\test.txt<br>Token resolves to: d:\target\09082011\test.txt  |
| Destination      | The IP address or host name of the final destination for a file being transferred. | Local file: c:\source\testfile1.txt<br>Remote file: d:\target\file1.[Destination].txt<br>Token resolves to: d:\target\file1.192.168.10.1.txt   |
| FileAvailability | The file availability used in a file transfer.                                     | IMMEDIATE, DEFERRED  |
| LocalDomain      | The local domain.  | A remote file name contains the local domain name.   |
| LocalFile        | The complete local file path.  | Local file: c:\source\testfile1.txt<br>Remote file: \$(LocalFile)<br>Token resolves to: c:\source\testfile1.txt  |
| LocalFileBase    | The local file name only.  | Local file: c:\source\directory\testfile1.txt<br>Remote file: \$(LocalFileBase)<br>Token resolves 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<br>Remote file: \$(LocalFileExt)<br>Token resolves to: txt (File transferred to the MFT Platform Server Windows Directory unless a path is configured.)        |

| Token            | Definition   | Generated Value (Examples)   |
|------------------|--|--|
| LocalFileName    | The local file name including the extension is used.                       | Local file: c:\source\directory\testfile1.txt<br>Remote file: \$(LocalFileName)<br>Token resolves 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<br>Remote file: \$(LocalFilePath)<br>Token resolves to: c:\source\directory  |
| LocalPathWODrive | The local file path without the drive letter or file is used.              | Local file: c:\source\directory\testfile1.txt<br>Remote file: \$(LocalPathWODrive)<br>Token resolves to: source\directory (File transferred to the MFT Platform Server Windows Directory unless a drive letter is configured.) |
| LocalUserId      | The local user ID used in a file transfer.                                 | Local User Id: TESTLAB\cfuser1<br>Local file: c:\source\directory\testfile1.log<br>Remote file: d:\target\file1\$(LocalUserId).txt<br>Token resolves to: d:\target\file1cfuser1.txt  |
| MgmtClass        | The management class used when a file is transferred to a z/OS system.     | Local file: c:\source\directory\testfile1.txt<br>Remote file: PRJOE.\$(MgmtClass).FILE1<br>Token resolves to: PRJOE.MGCLS12.FILE1  |
| NoLocalFileBase  | The base name of a local file is not used in the file name on a send.      | Local file: c:\source\directory\ a.b.c.txt<br>Remote file: c:\target\\$ (NoLocalFileBase)<br>Token resolves to: b.c.txt  |
| NoLocalFileExt   | The extension name of a local file is not used in the file name on a send. | Local file: c:\source\directory\ a.b.c.txt<br>Remote file: c:\target\\$ (NoLocalFileExt)<br>Token resolves to: a.b.c   |

| Token            | Definition   | Generated Value (Examples)   |
|------------------|--|--|
| NoRemoteFileBase | The base name of a remote file is not used in the file name on a receive.          | Local file: c:\target\\$<br>(NoRemoteFileBase)<br>Remote file: c:\source\directory<br>\a.b.c.txt<br>Token resolves to: b.c.txt   |
| NoRemoteFileExt  | The extension name of a remote file is not used in the file name on a receive.     | Local file: c:\target\\$<br>(NoRemoteFileExt)<br>Remote file: c:\source\directory<br>\a.b.c<br>Token resolves to: b.c.txt  |
| NotifyUser       | The remote user name configured to be notified in a file transfer.                 | Local file: c:\source\directory<br>\testfile1.txt<br>Remote file: d:\target\file1\$<br>(NotifyUser).txt<br>Token resolves to: d:\target<br>\file1JohnD.txt   |
| NotifyUserType   | The type of notification used for the remote user notification in a file transfer. | Local file: c:\source\directory<br>\testfile1.txt<br>Remote file: d:\target\file1\$<br>(NotifyUserType).txt<br>Token resolves to: d:\target<br>\file1Windows.txt (Windows, None, TSO, ROSCOE, Email) |
| PortNumber       | The port number used in the file transfer.   | Local file: c:\source\directory<br>\testfile1.txt<br>Remote file: d:\target\file1\$<br>(PortNumber).txt<br>Token resolves to: d:\target<br>\file146,464.txt  |
| PrinterName      | The printer name used in a file to print.  | <text>   |
| Priority         | The priority set in a file transfer.   | Local file: c:\source\directory<br>\testfile1.txt<br>Remote file: d:\target\file1\$<br>(Priority).txt<br>Token resolves to: d:\target<br>\file1Normal.txt  |



| Token   | Definition  | Generated Value (Examples)  |
|---|---|---|
| ProcessName                                   | The process name configured in a file transfer.       | Local file: c:\source\directory\testfile1.txt<br>Remote file: d:\target\file1\$(ProcessName).txt<br>Token resolves to: d:\target\file1CyberFus.txt  |
| RecordFormat                                  | The record format used in a file transfer.            | FIXED, BLOCKED, FIXED BLOCKED, VARIABLE, VARIABLE BLOCKED, UNDEFINED  |
| RecordLength                                  | The record length used in a file transfer.            | Local file: c:\source\testfile1.txt<br>Remote file: CFUSR.F\$(RecordLength).TEST<br>Token resolves to: CFUSR.F80.TEST   |
| RemoteDomain                                  | The remote domain used in a file transfer.            | A remote file name contains the remote domain name.   |
| RemoteFile (token used during a receive)      | The complete remote file path.                        | Local file: \$(RemoteFile)<br>Remote file: c:\source\testfile1.txt<br>Token resolves to: c:\source\testfile1.txt  |
| RemoteFileBase (token used during a receive)  | The remote file name only.                            | Local file: \$(RemoteFileBase)<br>Remote file: c:\source\directory\testfile1.txt<br>Token resolves to: testfile1 (File transferred to the MFT Platform Server Windows Directory unless a path is configured.)     |
| RemoteFileExt (token used during a receive)   | Only the extension of a remote file is used.          | Local file: \$(RemoteFileExt)<br>Remote file: c:\source\directory\testfile1.txt<br>Token resolves to: txt (File transferred to the MFT Platform Server Windows Directory unless a path is configured.)            |
| Remote filename (token used during a receive) | The remote file name including the extension is used. | Local file: \$(RemoteFileName)<br>Remote file: c:\source\directory\testfile1.txt<br>Token resolves to: testfile1.txt (File transferred to the MFT Platform Server Windows Directory unless a path is configured.) |

| Token   | Definition  | Generated Value (Examples)  |
|---|---|---|
| RemoteFilePath (token used during a receive)    | The remote file path without the file name is used.                       | Local file: \$(RemoteFilePath)<br>Remote file: c:\source\directory\testfile1.txt<br>Token resolves to: c:\source\directory  |
| RemotePathWODrive (Token used during a receive) | The remote file path without the drive letter or file name used.          | Local file: \$(RemotePathWODrive)<br>Remote file: c:\source\directory\testfile1.txt<br>Token resolves to: source\directory (File transferred to the MFT Platform Server Windows Directory unless a drive letter is configured.) |
| RemoteTransactionNumber                         | The remote transaction number used in a file transfer.                    | Local file: d:\fn\\$(RemoteTransactionNumber).txt<br>Remote file: c:\source\directory\testfile1.txt<br>Token resolves to: d:\fn\  |
| RemoteUserId                                    | The remote user ID used in a file transfer.                               | Remote user ID: TEST\cfuser1<br>Local file: c:\fn\file1.\$(RemoteUserId).txt<br>Remote file: c:\source\directory\testfile.txt<br>Token resolves to: c:\fn\file1.cfuser1.txt   |
| StorageClass                                    | The storage class used during a file transfer to a z/OS system.           | Local file: c:\source\directory\testfile1.txt<br>Remote file: PRJOE.\$(StorageClass).FILE1<br>Token resolves to: PRJOE.STANDARD.FILE1   |
| SysoutClass                                     | The SYSOUT class used during a file to print to a z/OS system.            | Local file: c:\source\directory\testfile1.txt<br>Remote file: PRJOE.\$(SysoutClass).FILE1<br>Token resolves to: PRJOE.A.FILE1   |
| SysoutCopies                                    | The amount of SYSOUT copies used during a file to print to a z/OS system. | Local file: c:\source\directory\testfile1.txt<br>Remote file: PRJOE.TS\$(SysoutCopies).FILE1<br>Token resolves to: PRJOE.TS2.FILE1  |

| Token             | Definition   | Generated Value (Examples)   |
|-------------------|--|--|
| SysoutDestination | The SYSOUT destination used during a file to print to a z/OS system. | Local file: c:\source\directory\testfile1.txt<br>Remote file: HST.\$(SysoutDestination).FILE1<br>Token resolves to: HST.NYPRINTER.FILE1              |
| SysoutFcb         | The SYSOUT FCB used during a file to print to a z/OS system          | Local file: c:\source\directory\testfile1.txt<br>Remote file: PRJOE.\$(SysoutFcb).FILE1<br>Token resolves to: PRJOE.STD2.FILE1                       |
| SysoutForms       | The SYSOUT forms used during a file to print to a z/OS system.       | Local file: c:\source\directory\testfile1.txt<br>Remote file: PRJOE.\$(SysoutForms).FILE1<br>Token resolves to: PRJOE.INVC.FILE1                     |
| SysoutUserId      | The SYSOUT user name used during a file to print to a z/OS system.   | Local file: c:\source\directory\testfile1.txt<br>Remote file: PRJOE.\$(SysoutUserId).FILE1<br>Token resolves to: PRJOE.MVSUSER1.FILE1                |
| SysoutWriter      | The SYSOUT writer used during a file to print to a z/OS system.      | Local file: c:\source\directory\testfile1.txt<br>Remote file: PRJOE.\$(SysoutWriter).FILE1<br>Token resolves to: PRJOE.WRITER1.FILE1                 |
| TransactionNumber | The local transaction number used in a file transfer.                | Local file: c:\source\directory\testfile1.txt<br>Remote file: d:\target\fs\$(TransactionNumber).txt<br>Token resolves to: d:\target\fsI331600053.txt |
| TransferFunction  | The transfer function used in a file transfer.                       | SEND, RECEIVE  |

| Token        | Definition  | Generated Value (Examples)  |
|--------------|---|---|
| TransferId   | The transfer ID assigned to a file transfer.  | Local file: c:\source\directory\testfile1.txt<br>Remote file: d:\target\file1.\$(TransferId).txt<br>Token resolves to: d:\target\file1.d1544fd2-5fb7-4ce6-a717-ac8907697e4f.txt |
| TransferWork | The type of a transfer being done. For example, file to file, file to job, and so on. | F-FILE, J-JOB, P-PRINT  |
| TryCount     | The try count used in transfer.   | Local file: c:\source\directory\testfile1.txt<br>Remote file: d:\target\file1\$(TryCount).txt<br>Token resolves to: d:\target\file13Times.txt                                   |
| Unit         | The unit used for a transfer to and from a z/OS system.                               | Local file: c:\source\directory\testfile1.txt<br>Remote file: PRJOE.\$(Unit).FILE1<br>Token resolves to: PRJOE.SYSDA.FILE1  |
| UserData     | The user data name used in a file transfer.   | Local file: c:\source\directory\testfile1.txt<br>Remote file: d:\target\file1\$(UserData).txt<br>Token resolves to: d:\target\file1MyUserData.txt                               |
| VolSer       | The volume used for a transfer to and from a z/OS system.                             | Local file: c:\source\directory\testfile1.txt<br>Remote file: PRJOE.\$(VolSer).FILE1<br>Token resolves to: PRJOE.CFP101.FILE1   |
| WriteMode    | The write mode used in a file transfer.   | C, R, A, CR, CA, CN   |

## Examples of Using File Name Tokens

During the transfer of a file, type the file's name using File Name Tokens instead of a regular file name.

The following examples use the system date/time: Wednesday, April 25, 1996 5:03:45.061 PM.

In the following example, you enter a string of File Name Tokens instead of entering a standard file name. Then the 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 the following example, you use the File Name Tokens to generate a resolved file name that has dashes between the date and time fields:

- File name: C:\directory\\$(SDD)-\$(SMON)-\$(YYYY)\\$(SHH24)-\$(SMI)-\$(SSS).dat
- Resolved name: C:\directory\25-APR-1996\17-03-45.dat

In the following example, the 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 name: \\Server\Volume\April\projectX\Wednesday\products.xls

In the following 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 days in a 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 name: C:\DOS\SHORTNM4.4P6

In the example, you use the File Name Tokens to create a file name in which the month is substituted for the server name, the day is substituted for the volume name, the time is separated by spaces, and the file name with the 3-character day of week abbreviation serves as the 3 character file name extension.

In the following example, you use DNI and File Name Tokens. You place sample.txt in the DNI directory and use File Name Tokens to designate the transferred file's directory and file name.

- File name: C:\\$(RemoteUserID)\\$(LocalFileName)\\$(LocalPathWODrive)
- Resolved name: C:\pat\sample.txt\\$(RemotePathWODrive)

Using \$(LocalPathWODrive) or \$(RemotePathWODrive) takes the path specified in the file name and transfers the file to the same directory, but different drives.

The various available time tokens are resolved at the beginning of a file transfer from the Initiating Platform Server. 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 due to retries.

## Rules for Use

When you create a file name that uses File Name Tokens, you must abide by the following rules:

- Substitution parameters are enclosed in \$(...). A dollar sign (\$) followed by an open parenthesis is followed by a token and a close parenthesis.
- Each \$(...) contains only one token.
- Any text in a remote file name that is not a substitution parameter is kept as is into the generated name.
- Codes can appear anywhere within a remote file name, such as the file name, directory name, share name, or server name.
- There can be any number of substitution parameters embedded within a file name.
- If the length of a resolved remote file name is greater than the maximum file name length allowed by MFT Platform Server for Windows (255 characters), the remote file name 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.
- The capitalization of substitution parameters effects the capitalization of the output. See [File Name Tokens List](#) for details.

- If a formatted name containing an invalid substitution code is given, the transfer fails with an error stating that a substitution code is bad.
- The feature is designed to work with DOS 8.3 and Win32 Long File Names. It is up to the user to ensure that the generated name is valid for the target system. Be careful when using a forward slash (/), back slash (\), or colon (:) to delimit dates and times as these are contain special meaning to the operating system.
- For remote systems which support long file names, embedded spaces are valid for a generated file name. However, MFT Platform Server for z/OS currently does not support embedded spaces in remote file names.

## PPA Tokens

The PPA substitutable fields use the percent sign (%) as the escape character instead of the dollar sign (\$) that file tokens use. The following table lists the substitutable parameters that are supported for PPA. The C:\a\b\c\d\config.txt file is used as an example.

| Substitutable Parameter | Description   | Example  |
|-------------------------|---|--|
| %DIR                    | The directory without any file name or drive name.      | a\b\c\d<br>sharename\a\b\c\                            |
| %DRIVE                  | The drive name.   | C<br>\\server\   |
| %NODRIVE                | The file name without any drive name.                   | a\b\c\d\config.txt<br>\sharename\a\b\config.txt        |
| %SDIR                   | The lowest level directory.                             | d  |
| %HDIR                   | The highest level directory.                            | a  |
| %NOSDIR                 | The directory name without the lowest directory.        | a\b\c  |
| %NOHDIR                 | The directory name without the highest level directory. | b\c\d  |
| %FILE                   | The file name without the directory.                    | config.txt   |
| %LFILE                  | The file name with a directory.                         | C:\a\b\c\d\config.txt<br>\\server\sharename\a\test.txt |

| Substitutable Parameter | Description  | Example       |
|-------------------------|--|---------------|
| %LLQ                    | The low level qualifier of a file (data after the last period(.)). | txt           |
| %HLQ                    | The high level qualifier of a file.                                | config        |
| %TRN                    | The transaction number.  | I824500001    |
| %PROC                   | The process name.  | ABC123        |
| %UDATA                  | The user data.   | USRDATAABC123 |
| %JDATE                  | The Julian date (YYDDD).   | 05236         |
| %JDATEC                 | The Julian date with the century (CCYYDDD).                        | 2005236       |
| %TIME                   | The time (hhmmss).   | 165030        |
| %GDATE                  | The Gregorian date (yymmdd).                                       | 050824        |
| %GDATEC                 | The Gregorian date with the 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, then the value **DRIVE** is used as a substitution. Similarly, %**PROC** becomes **PROC** and %**UDATA** becomes **UDATA** if there is no interaction with a z/OS system.

## Directory Tokens

There are two special tokens that are used for directory transfers.

### \$(SDIR)

The case-sensitive \$(SDIR) token can be used with a Receive as part of the LocalFileName path, and with a Send as part of the RemoteFileName path. For example, you can set file names for a Receive as follows:

- LocalFileName: C:\johndoe\data\\$(SDIR)\\$(RemoteFileName)
- RemoteFileName: C:\MFT Platform Server\data\\*

The text before the \$(SDIR) token is assumed to be a base directory.

If ScanSubDir is checked on and there are files in both the remote directory (C:\MFT Platform Server\data) and in remote subdirectories, the subdirectories are created in the local directory (C:\johndoe\data).

If this token is missing but ScanSubDir is checked on, then all the files from the remote directory and all subdirectories are located in the local base directory. The file 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 in the base directory path (for example, directory data from LocalFileName), the subdirectory is created with the same access as its base directory (johndoe). And all subdirectories created below it are created with the same access rights.

For a Send, \$(SDIR) is used as part of the RemoteFileName path, in the form of C:\MFT Platform Server\data\\$(SDIR)\\$(LocalFileName).

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

### **\$(MEMBER)**

The \$(MEMBER) token is used only for a Receive from a z/OS system. It is used for a similar purpose as the \$(SDIR) token, but we use a different token because dataset names work differently from directory names. Therefore, you can use this token to have file names on the local side that are the same as member names on the z/OS side.

If there is no \$(Member) in the file name from the z/OS side, the \$(MEMBER) token is not used. For example, if the path is C:\MFT Platform Server\\$(MEMBER)\whatever, it becomes C:\MFT Platform Server\whatever.



# TIBCO Documentation and Support Services

---

## How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit <https://docs.tibco.com>.

The [TIBCO Product Documentation](https://docs.tibco.com) website is updated frequently and is more current than any other documentation included with the product.

## Product-Specific Documentation

Documentation for TIBCO® Managed File Transfer Platform Server for Windows is available on the [TIBCO® Managed File Transfer Platform Server for Windows](https://docs.tibco.com) Product Documentation page.

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

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit <http://www.tibco.com/services/support>.
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at <https://support.tibco.com>.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to <https://support.tibco.com>. If you do not have a user name, you can request one by clicking Register on the website.

## How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](https://community.tibco.com). For a free registration, go to <https://community.tibco.com>.

## Legal and Third-Party Notices

---

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

ANY SOFTWARE ITEM IDENTIFIED AS THIRD PARTY LIBRARY IS AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND IS NOT PART OF A TIBCO PRODUCT. AS SUCH, THESE SOFTWARE ITEMS ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, and Slingshot are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2003-2022. TIBCO Software Inc. All Rights Reserved.