TIBCO MFT Platform Server™ for Windows

User Guide

Software Release 7.1 .1
March 2012

TIBCO provides the two-second advantage™

TIBCO®
The Power of Now®

Important Information

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 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.

TIB, TIBCO, The Power of Now, TIBCO ActiveMatrix BusinessWorks, TIBCO Adapter, TIBCO Managed File Transfer, TIBCO Managed File Transfer Command Center, TIBCO Managed File Transfer Internet Server, TIBCO Managed File Transfer Platform Server, TIBCO Managed File Transfer Platform Server Agent, Edge Server, Information Bus, Predictive Business, RocketStream, RocketStream Accelerator, Silver, and Slingshot are either registered trademarks or trademarks of TIBCO Software Inc. or its subsidiaries in the United States and/or other countries.

EJB, Java EE, J2EE, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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.

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.

TIBCO® Managed File Transfer Platform Server for Windows with RocketStream® Accelerator is entitled TIBCO® Managed File Transfer Platform Server for Windows in certain other product documentation and in user interfaces of the product.

TIBCO welcomes your comments on this publication. Please address your comments to:

TIBCO Software Inc.
200 Garden City Plaza
Garden City, New York 11530 USA
Web site: http://www.tibco.com
Technical Support E-mail: support@tibco.com
Technical Support Call Centers:
North and South America: +1.650.846.5724 or +1.877.724.8227 (1.877.724.TACS)
EMEA (Europe, Middle East, Africa): +44 (0) 870.909.3893
Australia: +61.2.4379.9318 or 1.800.184.226
Asia: +61 2 4379 9318

When you send information to TIBCO, you grant TIBCO a non-exclusive right to use or distribute the information in any way TIBCO believes appropriate without incurring any obligation to you.

Table of Contents

Preface

Intended Audience

This publication is intended for those individuals responsible for installing, configuring, managing, and operating the Windows component of MFT Platform Server.

About This Manual

This manual provides instructions for using the Windows component of MFT Platform Server. Consult your network manager regarding network equipment and procedures at your installation site.

Chapter 1 *Installation* describes how to install, uninstall and repair MFT Platform Server.

Chapter 2 *The MFT Platform Server Administrator* describes the Administrative Client component of MFT Platform Server for Windows.

Chapter 3 *The Command Line Interface* describes how to produce batch programs using parameters created for all of the MFT Platform Server functions.

Chapter 4 *Extended Features* describes various utilities and components of MFT Platform Server for Windows.

Appendix A *The Event Log* describes how to use the Event Viewer, the event logging service that you can use to monitor events in your Windows system.

Appendix B *Cached Windows Passwords* describes all of the functionality, capabilities, and advantages of using the cached password feature.

Appendix C *File Name Tokens* describes the MFT Platform Server for Windows feature that enables the MFT Platform Server user to create or read file names based upon date, time, user, and file transfer information.

Appendix D *Configuring HIS for MFT Platform Server* assists users configuring MFT Platform Server with a Microsoft HIS server.

1

1. Pre-requisites

Minimum Operating Systems Version

MFT Platform Server for Windows is supported on the following minimum operating system levels or above:

- Windows XP SP3
- Windows Vista SP2
- Windows 7 SP1
- Windows 2008 SP2
- Windows 2008 R2 SP2
- Microsoft HIS 2000 or higher (for SNA transfers; not required for TCP/IP transfers)
  Note: TIBCO MFT Platform Server for Windows v7.1.1 will be the last release that supports SNA. Future releases of TIBCO MFT Platform Server for Windows will no longer support SNA.

MFT Platform Server™ for Windows is a 32-bit application which is fully supported on 64-bit Windows operating systems.

Customers should migrate to supported versions of Windows Client and Windows Server because in the event that you encounter an issue/outage in your environment on an unsupported product, Microsoft engineers may not be able to help resolve the issue until you've upgraded to a supported level.

Note: Support is provided by TIBCO only for the vendor's generally supported release versions. Once the operating system goes into extended support mode, or the vendor no longer supports a version, it will cease to be supported by TIBCO Technical Support.

Minimum Hardware

- 1 GB of memory
- 100 MB of disk space
- An appropriate amount of additional local storage is recommended for file transfer data
- An x86 architecture based processor is required to run MFT Platform Server for Windows and RocketStream file transfer acceleration

Sizing Guidelines

In addition to the above minimum requirements:

- For up to 100 concurrent transfers, two or more processor cores at 2.5 GHz or faster
- For up to 200 concurrent transfers, four or more processor cores at 2.5 GHz or faster
- For more than 200 concurrent transfers, eight or more processor cores at 2.5 GHz or faster
- One additional processor core at 2.5 GHz or better for extensive use of encryption or compression.

- When performing over 100 concurrent transfers, TIBCO recommends using the transfer throttling feature.
- TIBCO recommends a maximum of 50 DNI templates per server

Please contact MFTSalesEnablementTeam@tibco.com for assistance with the above guidelines or any special requirements.

2. Installation

In this chapter, you will learn how to install MFT Platform Server for Windows. The main procedures will go through:

- Defining user rights for proper operation of the MFT Platform Server for Windows service
- Install the MFT Platform Server for Windows software

2.1 Account Rights Needed for MFT Platform Server for Windows

The MFT Platform Server for Windows Service account must belong to the Administrators group on the local machine. The following four rights will be assigned to that account automatically during installation:

- Act as part of the operating system
- Create a token object
- Log on as a service
- Replace a process token level

If a Local System account is used for the MFT Platform Server for Windows Service account, then these four rights are not assigned.

2.2 Install

2.2.1 Installing MFT Platform Server for Windows Software

1) Email TIBCO at support@tibco.com the machine name of the server you will be installing MFT Platform Server software on. TIBCO will respond by sending you a license key for you to use either during the install or to apply at a later date.

2) Exit any other programs that you might have running before installing MFT Platform Server for Windows

3) Go to the directory you have placed the MFT Platform Server software. Work your way down through the sub directories until you get to the disk1 folder. In this folder are the install files needed. Double click on **Setup.exe** to begin the install process. Below is the configuration screens you will be given during the install, fill in the information as requested:

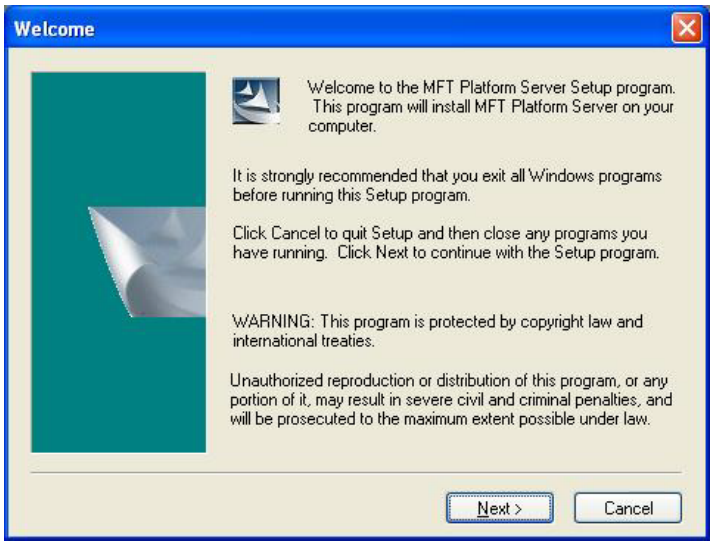


Figure 1

Read and click the Accept button on the License Agreement screen to continue.

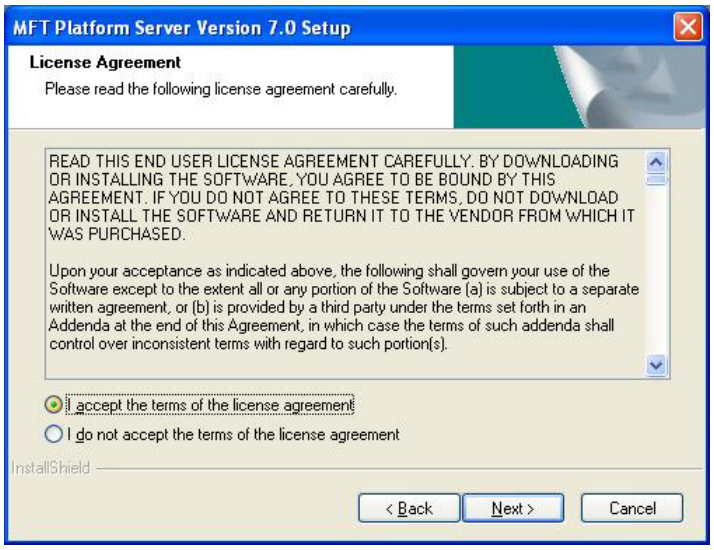


Figure 2

Figure 3

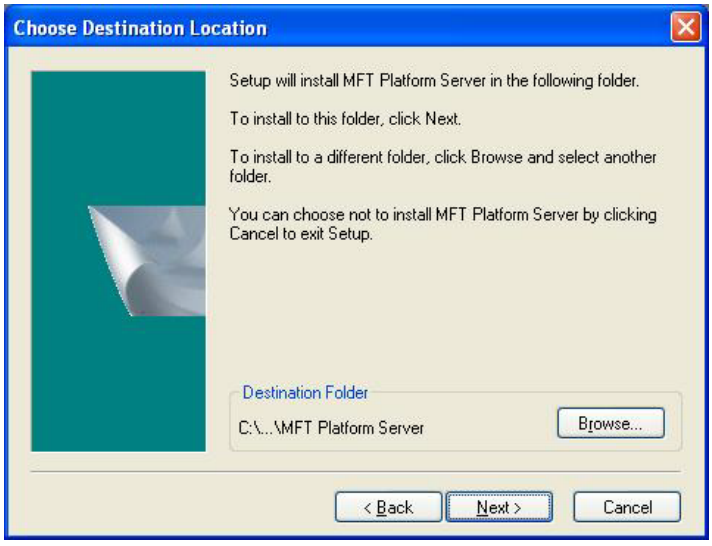


Figure 4

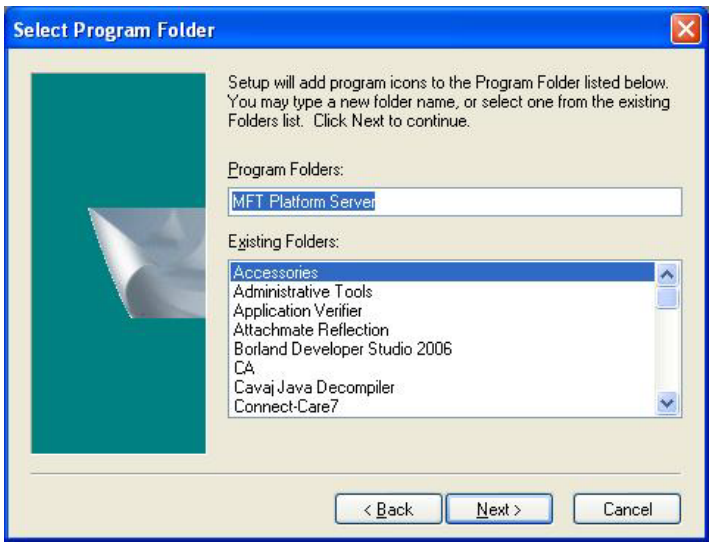


Figure 5

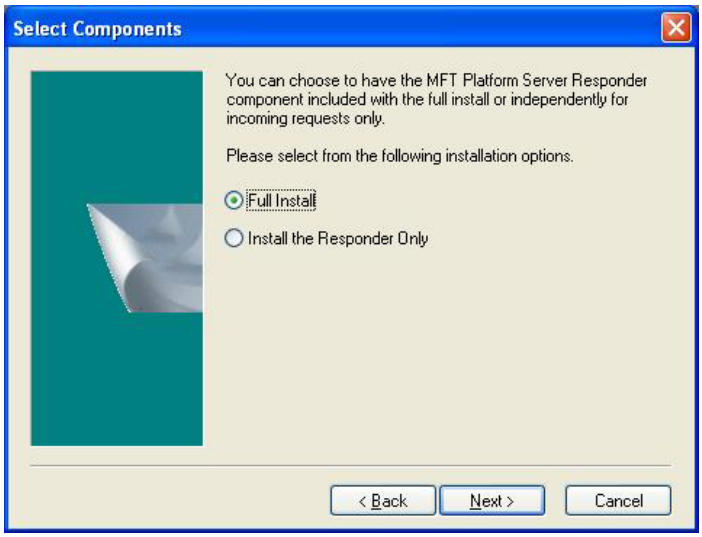


Figure 6

Note: If you are not initiating any transfers from this platform server, install it as a Responder Only.

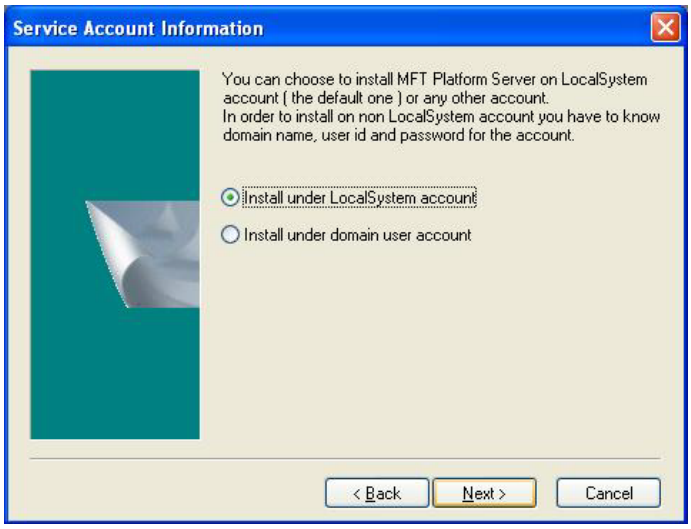


Figure 7

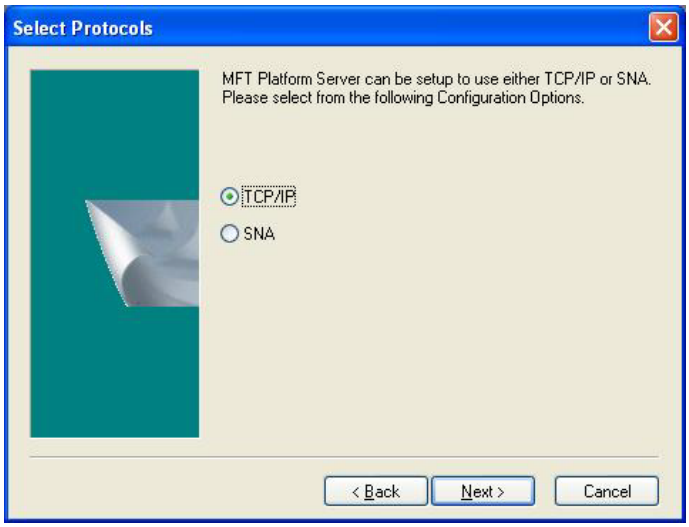


Figure 8

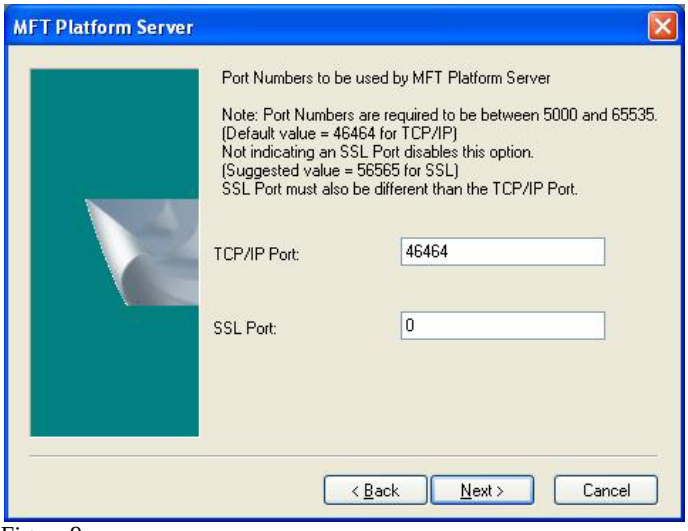


Figure 9

Both Non-SSL and SSL ports with MFT Platform Server. These can be configured later if needed.

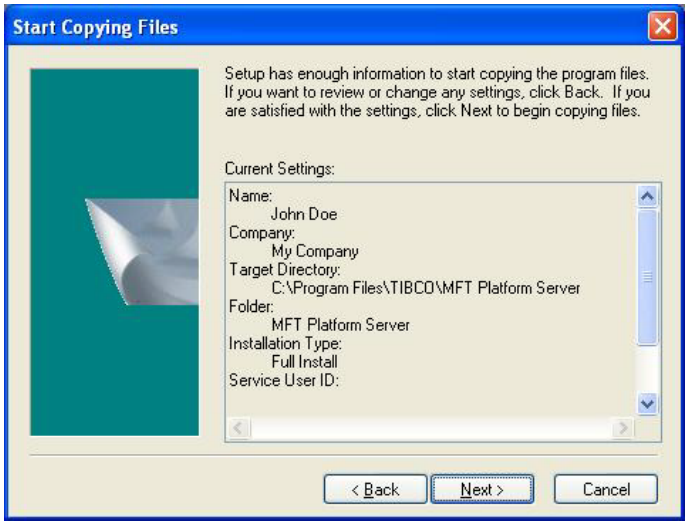


Figure 10

Figure 11

You can receive a license key from TIBCO by emailing support@tibco.com. If you do not have a key at this time you may click **No** and apply a key at a later date. If you have the key click **Yes** and proceed to the next screen.

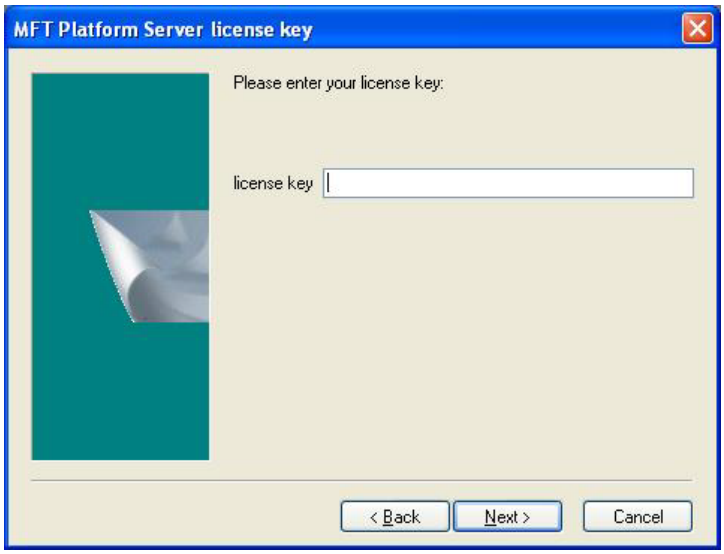


Figure 12

Figure 13

Once the license key is applied the install will continue. You will see two local groups get created on the server, cfadmin and cfbrowse. These groups are primarily used in conjunction with the Platform Server Command Line Interface. Please see the Command Line Interface section for more information:

Figure 14

Figure 15

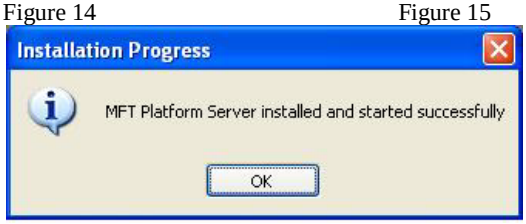


Figure 16

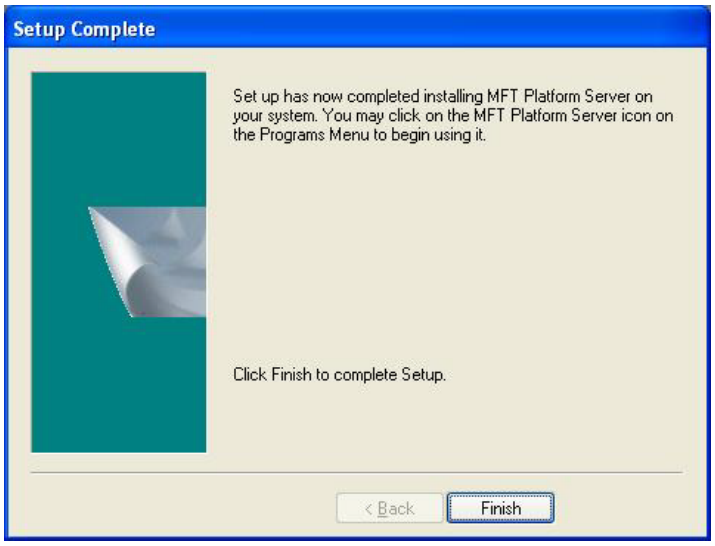


Figure 17

Once you have installed the software and clicked **Finish** you can go to:

Start > Programs > MFT Platform Server > MFT Platform Server Administrator

If you could not install a License Key at the time of install go to section *1.4 License Key*.

2.2.2 Silent Install

The installation file, setup.exe, in our example was generated from InstallShield and accepts additional command line parameters. The following parameters are used in the creation and execution of a silent install script:

/r	Record Mode
/s	Silent Mode
/f1	Response Filename

Create a response file by executing the command:

setup.exe /r /f1”c:\PS71Install.iss”

The InstallShield will record all the input entered and store it in the file path given. Subsequent installations may be installed silently by referencing the install script.

setup.exe /s /f1”c:\PS71Install.iss”

Note: The MFT Platform Server service may need to be started after installation.

2.3 License Key

2.3.1 Machine Name

MFT Platform Server for Windows needs a license key in order to function. The name of the machine where MFT Platform Server for Windows is installed is necessary in order to generate a key. Please follow the steps below to find the machine name.

1. From the command prompt, issue a “net name” command. The first name displayed under the dashed lines is you machine name.

2. Email this machine name to support@tibco.com with a request for a license key.

2.3.2 Applying the License Key

Once you receive your key from TIBCO you have to choose 1 of 3 ways to apply the key.

1. You can click on: Start>Programs>MFT Platform Server > MFT Platform Server Apply Key. This will bring up the command prompt window requesting you to enter your key. Copy and paste the key in and hit <Enter>.

2. You can run the MFT Platform Server **cfapplykey** program provided. This program is located in the directory where MFT Platform Server for Windows was installed; if you used the default directory it would be located here: c:\Program Files\TIBCO\MFT Platform Server\System. Double click on cfapplykey.exe program and copy and paste your new key in and press <Enter>.

3. You may also apply the license key using the following command which can be used to apply the license key remotely as well:

cfapplykey –K kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk

K – operand used to indicate the license key is to follow.

Copy and paste your 56 character key in place of the lowercase k’s and then hit <Enter>.

Assuming everything is done correctly you should see the following message:

```
The license key has been successfully applied.
```

Note: After applying the license key you must stop and start the MFT Platform Server service.

2.4 Security Modifications needed for MFT Platform Server

2.4.1 Service Account Rights

During the installation process you have the choice to install the MFT Platform Server service to be run under the Local System Account or under a domain user's account. When installed using the Local System account the service will run without any changes needed to the account. If you choose to install the product using a domain user's account the following rights should be automatically assigned to that user's account during the install:

- Logon As A Service
- Create a Token Object
- Act as Part of the Operating System
- Replace a Process Level Token

If the account the service is running under should change for any reason your local administrator will need to assign these rights to the new account being used. Changes of this nature require the MFT Platform Server service to be restarted.

2.4.2 User Account Rights

MFT Platform Server for Windows fully exploits all of the security features that are provided in Windows. When MFT Platform Server for Windows is acting as responder, it uses the User ID and password that are passed from the remote system to perform the transfer.

The system logs the user into the Windows Server as a batch process. To do this, the User ID/s that will be used with MFT Platform Server for Windows must be granted the right to **Log on as a batch job**. Any User ID that will be used to respond to requests from a remote system or to validate queue transactions must be granted this right. Please consult your local administrator in order to define this right on the systems where MFT Platform Servers will be responding to requests.

Once the operating system accepts the User ID and password, the thread which performs that transfer changes itself to run using the authority of the given user. This impersonation is in effect from the time of file access or creation until the completion of the transfer.

2.4.3 Remote Domain

There are several ways that MFT Platform Server specifies the remote domain name so that a user can be authenticated.

2.4.3.1 Configurable Remote Domain

The remote domain name can be specified explicitly as part of the transfer. This can be done on the Transfer Properties panel by specifying "domain/user" in the User Id field. Invoking this feature enables the user to specify the exact user on the network under whose authority the transfer should execute. This is the first check that will be made for authentication of the user.

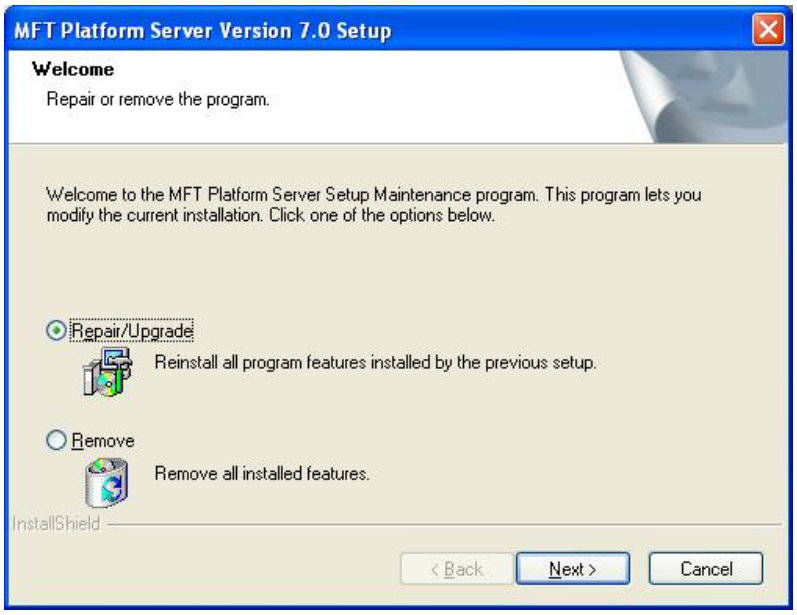
2.5 Upgrading MFT Platform Server

Upgrading from a prior version of MFT Platform Server for Windows requires you to stop the present MFT Platform Server service from running.

1. As a precaution we advise you to backup your files prior to upgrading. The following files should be backed up.
   - AccessControl.cfg
   - Cfalias.cfg
   - CfgPostProc.cfg
   - cfnode.cfg
   - cfprofile.cfg
   - cfrprofile.cfg
   - sslauth.cfg
   - Comtblg.dat
   - ftmssvr.pqf
2. Run setup.exe.
3. A Welcome window will open asking you if you want to Repair/Upgrade or Remove the product. Select the Repair/Upgrade option and click the Next button.
4. You will receive a warning that a copy of the product already exists and asked the question if you want to continue with the upgrade. Click the Yes button.
5. You will then be asked if you want to save your current MFT Platform Server settings. Answer Yes or No. If you answer No you will be asked if you want to apply a license key at this time. Answer Yes or No. If you say No you will need to apply a license key later using the cfapplykey command.
6. The upgrade will proceed and complete based on your answer to the above step. Replace the configuration files, with the files you backed up from the prior release.
7. Start the MFT Platform Server for Windows service.

2.6 Removing or Repairing MFT Platform Server

To remove or repair MFT Platform Server, you must run the setup.exe program. This may accessed from the program group, by selecting Uninstall MFT Platform Server.



2.6.1 Uninstall

Before removing MFT Platform Server, you must perform the following steps.

1. Close all MFT Platform Server programs before attempting to uninstall the MFT Platform Server product.
2. Since cached passwords are saved in a restricted area of the registry, the uninstall program cannot delete them. Therefore, if you are using cached passwords you must use X:DELETEALL to remove the passwords prior to running the uninstall program. Should you run the program without removing the passwords, the \\HKEY_LOCAL_MACHINE\SOFTWARE\TIBCO registry key entry will not be removed.

1. Click the **uninstall MFT Platform Server icon** (which lies in the same program group as the MFT Platform Server icon).
2. Click the Remove radio button and select next.

   You will be asked if you would like to remove the MFT Platform Server application and all of its components.
3. Click **Yes** to start the uninstall process.

   During the uninstall process you will be warned that several .DLL files will be deleted. The message will also alert you to the fact that these shared .DLLs are not being used by any other application. The application is aware of this because there is a directory that is maintained in the Registry that keeps track of all the .DLL files that are shared among different applications. You should positively confirm that it is OK to remove the .DLL files specified. Removal of these files will not cause any problems on your system.

   When the files have been deleted or saved the Uninstall process will continue and the dialog box will display that the Uninstall has completed successfully.
3. You will then receive a pop-up window stating that MFT Platform Server has been removed from your system. It is recommended that you restart your machine to remove files that were in use during uninstall.

4. Click **OK**.

2.6.2 Silent Uninstall

As in the case for installation, a script may be created for unattended uninstalls. The uninstall file in our example, setup.exe, was generated from InstallShield. The procedure below requires an installed copy of MFT Platform Server. The Platform Server service must be stopped before proceeding.

Create the response file:

setup.exe –r –f1"c:\PS71Uninstall.iss"

The script file may now be referenced for subsequent silent uninstalls.

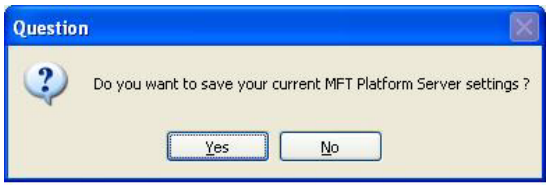
etup.exe –s –f1"c:\PS71Uninstall.iss"

Note: The MFT Platform Server service must be stopped before un-installing, otherwise the uninstall will fail to completely remove the product.

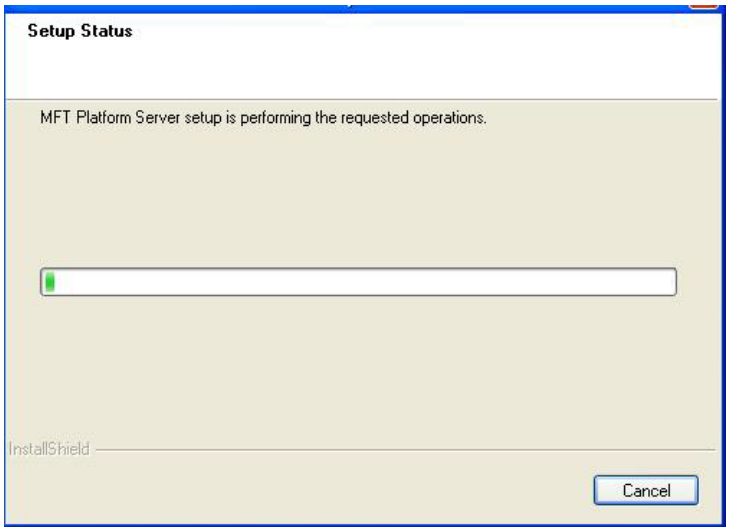
2.6.3 Repairing MFT Platform Server

To repair MFT Platform Server, select the Repair button after running the setup.exe program.

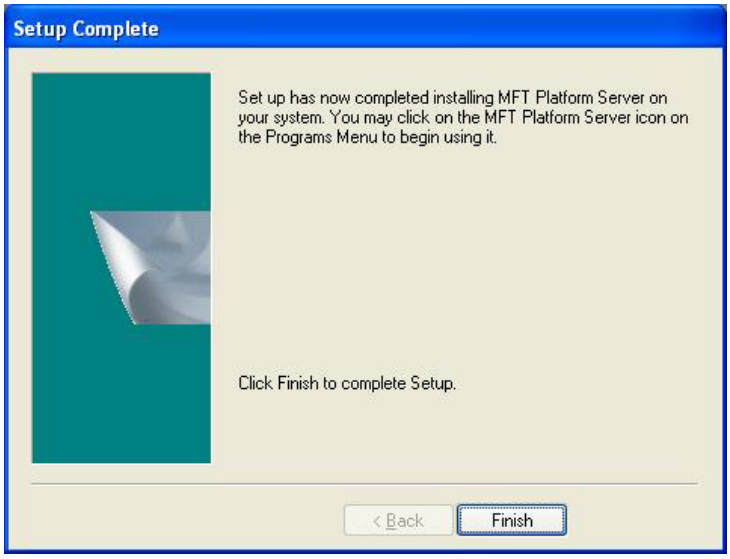
You will receive the following prompt:



A status bar will display the repair process progress.



The setup Complete will appear when the repair has been completed.



2

3. The MFT Platform Server Administrator

The MFT Platform Server Administrator provides an explorer-type interface that you can use to:

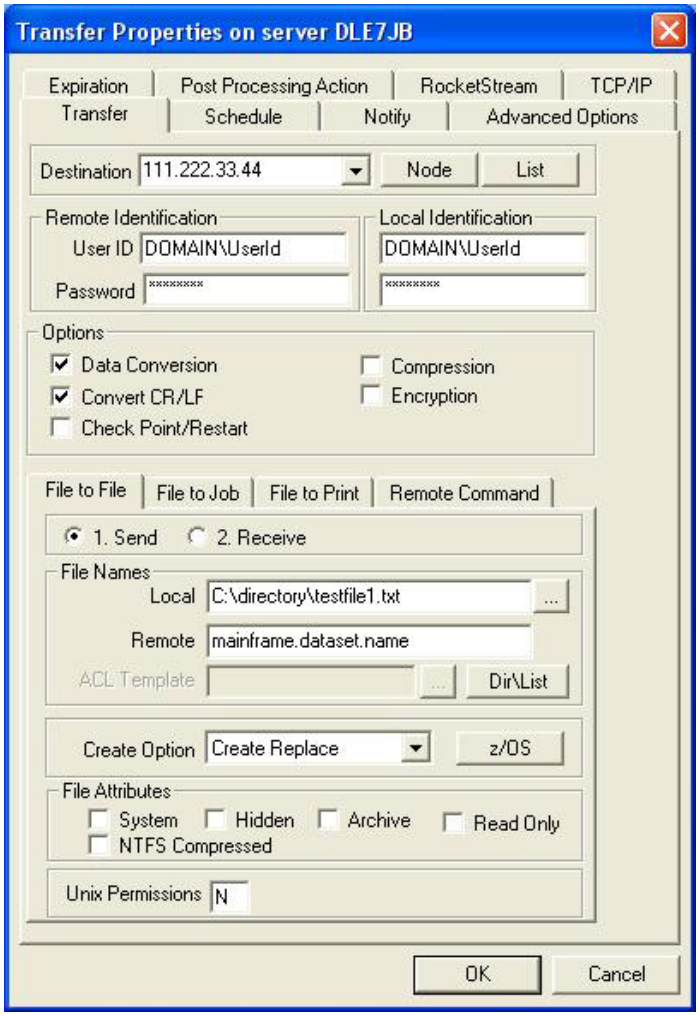
- Directly access the MFT Platform Server
- Define SSL information
- View and administer the queue of transfers
- View listings
- Modify server settings
- Create, modify, and delete transfers, templates, and DNI entries

3.1 Sample Transfer Using MFT Platform Server Administrator

This section is intended to be an overview of how to do a simple File to File transfer using MFT Platform Server Administrator. The buttons and parameters are described more in depth later in this chapter.

When you start the MFT Platform Server Administrator you will automatically be attached to your MFT Platform Server for Windows. To do the transfer, fill in the panel with the information for the transfer that you want to perform. The information in the Destination field, the Remote Identification, Remote File Name and Local File Name are all required fields. Some things that you can change would be the options such as Data Conversion, Check Point/Restart, compression or encryption. Some other options on the panel include the choice of Sending the file or Receiving the file.

One way to get to the Transfer GUI panel is to click on the Transfer icon  and choose **Advanced TCP Transfer**. You will see a blank transfer panel unlike the example that has been filled in:



For simplicity, let's say that you are going to do a File to File transfer from Windows to a mainframe.

The Destination is the remote system where you are going to send the file. This can be an IP Name (or Address) or the APPLID for MFT Platform Server for z/OS running on the mainframe. The information in this field is going to depend on the remote system and the protocol that is being used for the transfer. This information will be kept in a pull-down so that it can be used for future transfers.

The remote system may also be pre-defined by the user. This is referred to as a Node. By clicking on the Choose Node button, a list of your pre-defined nodes is displayed. Select a node from the list and the destination information will be filled in automatically.

The Remote Identification is the user id and password on the remote system. Therefore, if you are sending a file to the mainframe, then you would place your mainframe user id and password in this section. The password will be shown as asterisks as you type it.

Local Identification is your Windows user id and password.

The options in the Options section can simply be clicked on or off. There are four options that have additional parameters, Data Conversion, Check Point Restart, Compression and Encryption. These additional parameters can be found under the Advanced Options tab. Check Point/Restart has an additional parameter called Check Point Interval. This parameter tells MFT Platform Server for Windows how long to wait to take a checkpoint. Compression has an additional parameter called Type. The type of compression can be RLE, LZ, ZLIB1- ZLIB9, Default or None. Encryption has an additional parameter called Method. The method of encryption can be DES, Triple DES (3DES), Blowfish, Blowfish Long, Rijndael, Default or None.

Data Conversion has three additional fields so that data may be converted to or from ASCII or EBCDIC on the local or remote system.

Click on OK to get back to the main panel.

Below the Options section are the tabs for the type of transfer that you would like to do. For this example, we are going to concentrate on the File to File section.

You must decide if this will be a Send or Receive, then select the appropriate radio button. For this example, we will be initiating a Send transfer.

The Local File Name for this example would be the name of a file on your local Windows machine. If you do not remember the name of the file, the box to the right of the Local File Name (the box with 3 dots) will bring up a list of files on your machine (or network). Then you can just double click on the name of the file. This is particularly helpful in eliminating errors while typing the directory or file name.

The Remote File Name in this case would be a mainframe dataset name. This can be an existing dataset name or a new dataset name. For this example, we will enter a new dataset name.

The ACL Template would be used on a Receive. 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.

MFT Platform Server for Windows has the ability to transfer entire directories. The DIR button gives the options for the directory transfer such as the ability to scan sub directories and stop on failure.

The Create Option parameter states if this file exists or not. You have the option to create it if it does not exist, replace it if it does exist, append to an existing file, etc.

The File Attributes would be used to give attributes to a file that you are receiving to Windows.

The z/OS button to the right of the Create Options field, allows you to give attributes to the file that you are sending to the mainframe, such as record format, record length, blocksize, allocation, etc.

All the information for your basic file transfer is located on this panel. However, there are more transfer options under the other tabs along the top of this panel. Information under these tabs includes scheduling, defining the compression to be used (when compression is selected on the main GUI panel), the port to be used with a TCP transfer or the Mode Name to be used with an SNA transfer.

When you have all the information for your file transfer input, you can click on the OK button at the bottom of the panel. This will initiate the transfer. You can then click on the Transfer icon to see the progress of your file transfer.

Please refer to the following sections for more details on each parameter on the MFT Platform Server Administrator GUI.

3.2 MFT Platform Server Administrator Parameters

When you first initialize the Interactive Interface, the MFT Platform Server Administrator panel will automatically connect to your MFT Platform Server for Windows. To get to the Transfer GUI panel, you can either select the icon below the server name or the transfer icon (white with the yellow arrow) on the tool bar.

If you select the icon under the server name, highlight the icon, right click and select a new transfer, then select the protocol that you wish to use for the transfer. This will bring up the transfer panel.

If you select the icon on the tool bar, when you click on the icon you can select the protocol that you wish to use for the transfer. This procedure will bring up the transfer panel.

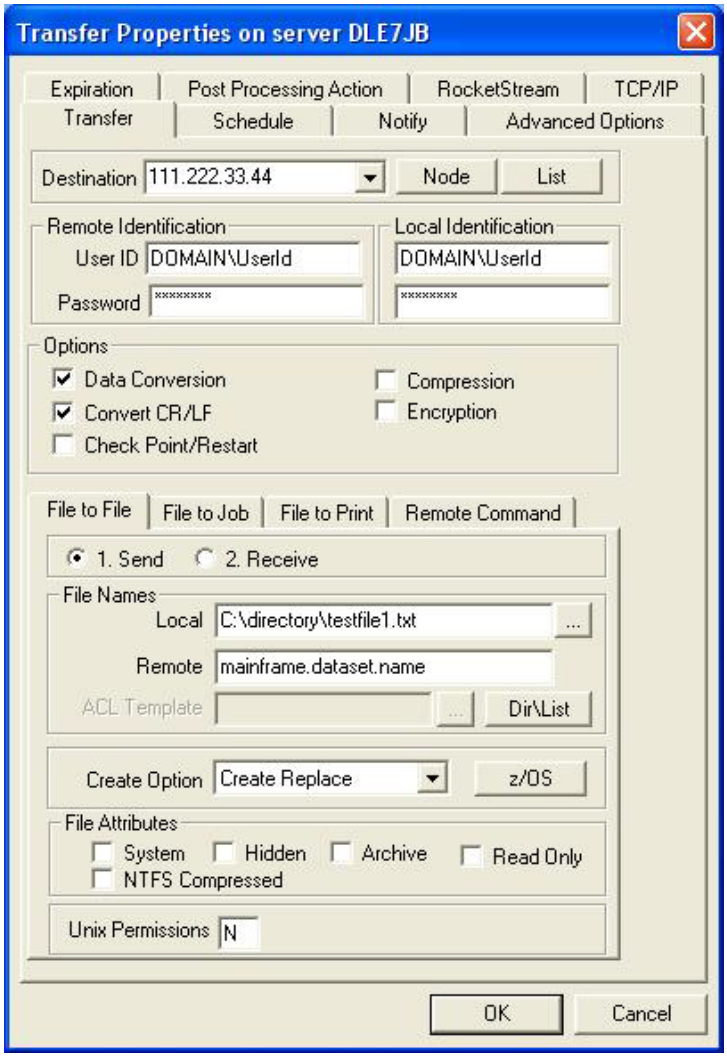
The transfer panel can be described as having two different halves. The top half of the panel has several fields that are universal to all transfer types. The lower half of this panel consists of four tabs. Each tab represents a different type of transfer that is supported by MFT Platform Server for Windows: File to File, File to Job, File to Print and Remote Command.

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

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

3.2.1 Transfer Tab

There are several fields that are common to all file transfers and will appear on the top half of the panel regardless of the transfer type selected. These fields are referred to as Universal Fields and are described below.



3.2.1.1 Universal Fields

The universal fields are the fields located on the top half of the transfer panel. The four tabs on the bottom half of the panel will be explained separately.

- **Destination** - This is the address of the remote system.
  - For TCP/IP transfers, this is the DNS Name, WINS Name, or IP Address (for example 251.250.41.5).
  - For Windows to Windows transfers, this is the LU Name alias or the CPI-C Name configured in HIS.
  - For SNA transfers, this is the VTAM APPLID of the MFT Platform Server started task on the z/OS system.

  The destination field has a pull down list that is designed to keep a list of the remote systems that were used in the past. A pre-defined Node may also be used in the Destination field.

- **Node -** This is the name of the remote system as defined using the cfnode program provided with MFT Platform Server for Windows. If there is a Profile associated with the Node, then the Remote Identification field will be filled in with “Default from node”. If no profile is found, then the field will be blank. You may type a Node in the Destination field and leave the userid and password blank; the information will be picked up from the profile definition if it is defined.

  If any of the transfer settings conflict with the node settings the user will be notified with a pop-up message box. The user can allow the transfer to be modified by clicking the OK button or they can stop the transfer by clicking the Cancel button.

- **List -** Displays a list of distribution lists available to choose from as defined in the **cflist.cfg** file found in the MFT for Windows install directory. Note: The use of a distribution lists is supported for SEND transactions only.

- **Remote Identification -** The information that is specific to the user on the remote system.

  User ID

  The User ID for the remote system, or the name by which the issuer is known to the remote system. The userid may be 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 userid. The userid is generally not case sensitive, unless going to a UNIX system. The User ID defaults to the last Issuer ID entered in this field. If a Node was selected and there is a Profile associated with the Node, then this field will be filled in with “Default from node”. If no profile is found, then the field will be blank.

  Password

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

  A feature called **cached passwords** allows the user to specify a password for a particular remote Windows User ID and store the password in the Windows registry on the remote system. The user will be able to perform MFT Platform transfers to that Windows system without having to specify the password. For more information on this feature please refer to the *Cached Passwords* Appendix.

  Note: If the user’s password on a remote z/OS system has expired, he or she will be unable to access a z/OS file from MFT Platform Server Administrator. In order to change the password, go to the password field on the main panel under the Transfer tab. Specify both the old password and the new password in the password field, separated by a slash (i.e. old/new).This will change the z/OS password to the new one specified.

- **Local Identification -** The Transfer Properties panel also allows you to specify the local authentication credentials for transfers. The userid may be 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 userid. The Local Identification is set to the user id of the logged on user. The default value for the password is "X:" which will cause the MFT Platform Server to read the cached password for this user. If you would like to use this feature, you must first cache your password. The user may override the default and enter a password, or any of the other cached password keys: 'X:password', 'X:DELETE', 'X:DELETEALL', or 'X:'. For more information on cached passwords, please refer to the *Cached Passwords* Appendix.

- **Options -** Options allows the user to specify ASCII to EBCDIC translation, convert Carriage Return/Line Feed, Check Point/Restart, Compression and Encryption. These options are defined as follows:

  Data Conversion - This is used to convert data between ASCII and EBCDIC. Transfers can be either binary or text. If the box is left unchecked, the transfer will be a binary transfer. If the box is checked the transfer will be a text transfer. There are additional parameters under the Advance

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

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

Check Point / Restart - This parameter 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 sender's checkpoint and its own checkpoint in the persistent queue. In the event of a failure, the initiator and the responder negotiate the saved checkpoint information and restart from the last known good checkpoint. Check Point is specified in minutes under the Advanced Options tab.

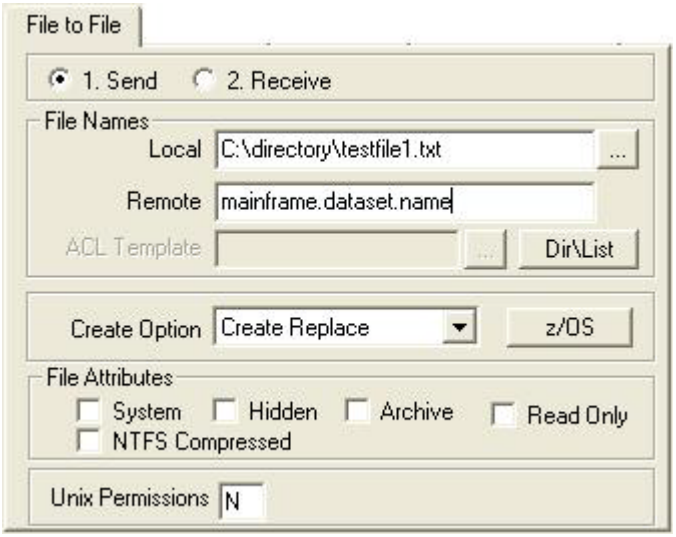
Compression - This parameter allows the user to specify that compression will be used for this transfer. Select the checkbox to turn compression on and then go the Advanced Options tab to select the Type of Compression to be used for the transfer. Compression compresses data on the sender side of the transfer and decompresses the data on the receiver side of the transfer. This 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.

With certain types of data, compressing the data sometimes will result in the compressed data being larger than the original data. *Smart Compression* accounts for this and alleviates the situation by transmitting only the data packets which are smaller than the original. When this occurs, it saves the increased network bandwidth of the larger "compressed" packet, and even more importantly, it saves the CPU cycles on the receiving side which would essentially be wasted.

Encryption - This parameter allows the user to turn encryption on and off. Select the checkbox to turn encryption on, then go the Advanced Options tab to select the Method of Encryption to be used for the transfer.

3.2.1.2 File to File tab

Select the File to File tab to store the contents of the file transfer in a file. The fields of this tab are described below.



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

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

- **Dir/List -** 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.

StopOnFailure	If the current file transfer fails, it will not try to transfer the rest of files.
ScanSubDir	This will cause not only the directory from the file path to be scanned, but all subdirectories as well. (Not available for List transfers.)
Test	Allows the user 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.

- **Create Options -** You must choose one of the following effects for the file being transferred:

Create	Create 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	Replace the contents of the destination file with the contents of the source file.
Append	Append the contents of the source file to the end of the destination file.
Create Replace	If the file does not exist on the system, then 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, then 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	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.

- **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.
Hidden	A file that cannot be seen by the user.
Archive	Select archive if you want to mark a file that has changed since it was last backed up.
Read Only	This 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.

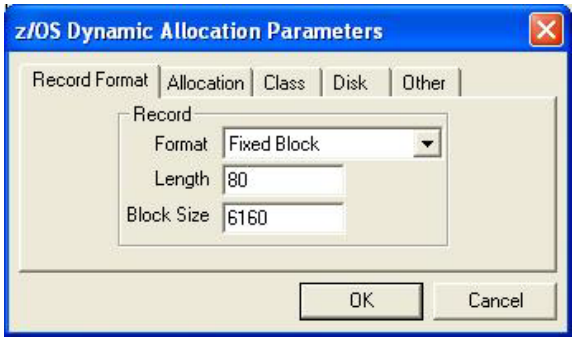
- **UNIX Permissions -** When a file is created on a UNIX system, MFT Platform Server for Windows has the ability to set the UNIX Permissions on the file. UNIX permissions are defined by a three digit number such as 777 (the same as for chmod command). The default value for this parameter is the file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

> Note: Permissions will be set up under the file only if file was created. In other words UNIX Permissions works only with Create, CreateReplace and CreateReplaceNew file options when the file is being created.

3.2.1.2.1 z/OS Options Panel

By clicking on the z/OS button you will reveal the z/OS Dynamic Allocation Parameters window which provides four property pages that contain the fields that are necessary to specify when the user is sending files to a MFT Platform Server for z/OS partner.

3.2.1.2.1.1 Record Format

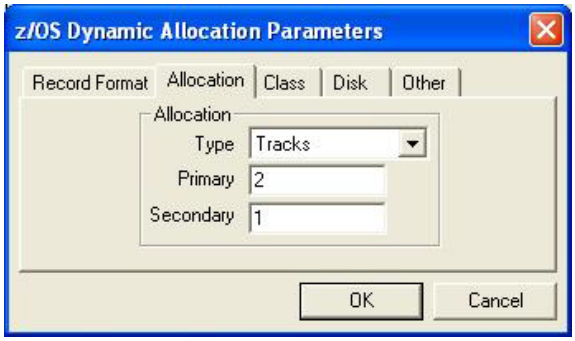


- **Format** - Determines the logical record length (LRECL). Choose one of the following formats:

Fixed	Each string contains exactly this number of characters.
Fixed ASA	Each string contains exactly the number of characters and the use of ASA characters on z/OS.
Fixed Block	All blocks and all logical record are fixed in size. One or more logical records reside in each block.
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.
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.
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.
Variable	The length of each string is less than or equal to this number.
Variable ASA	The length of each string is less than or equal to this number and the use of ASA characters on z/OS.
Variable Block	Blocks, as well as logical record length, can be any size. One or more logical records reside in each block.
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.
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.
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.
Undefined	Blocks are of variable size. There are no logical records. The logical record length will appear as zero. This record format is usually only used in load libraries. Block size must be used if you are specifying Undefined.

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

3.2.1.2.1.2 Allocation



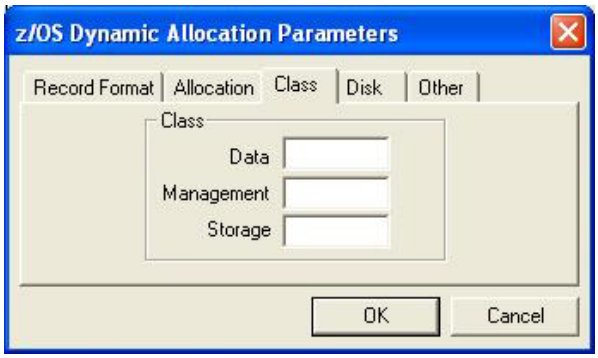
- **Type -** The valid values are as follows:

Tracks
Cylinders
Megabytes
Kilobytes

- **Primary -** Allocation Primary is used by the z/OS partner when creating datasets as the initial number of units of TRACKS, CYLINDERS, etc. to allocate.
- **Secondary -** The secondary allocation quantity is used by the z/OS partner when creating datasets as the next number of units of TRACKS, CYLINDERS, etc. to allocate once the initial space in the dataset has been exhausted.

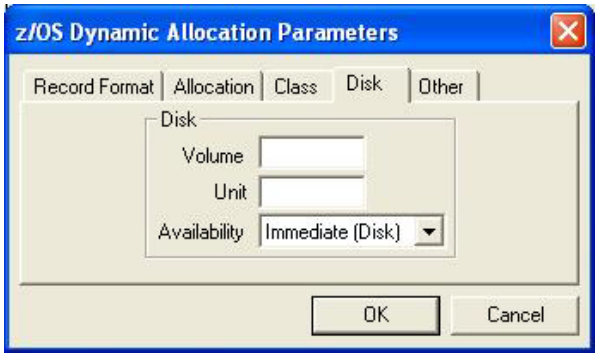
The default is Kilobytes with zero Primary and zero Secondary space. This default configuration will pick up the size of the file being sent to the z/OS system and allocate the appropriate space.

3.2.1.2.1.3 Class



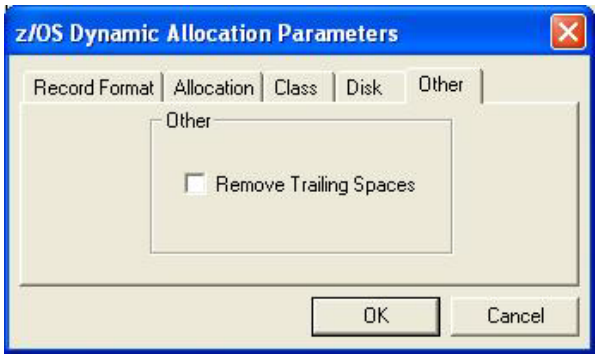
- **Data -** This represents the z/OS Data Class as defined to the Data Facility /System Managed Storage. In addition, it is used to indirectly select file attributes such as Record Format and Logical Record Length. This is a 1–8 character value, which contains numeric, alphabetic, or national characters (in the United States these are $, #, or @). The first character must be an alphabetic or national character.
- **Management -** This represents the z/OS Management Class as defined to the Data Facility /System Managed Storage. This is a 1-8 character value, which contains numeric, alphabetic, or national characters (in the United States these are $, #, or @). The first character must be an alphabetic or national character.
- **Storage -** This is a 1–8 character value which represents the z/OS Storage Class as defined to the Data Facility /System Managed Storage, which is used to indicate the host file's media type and the installation's backup, restore, and archive policies. See your mainframe administrator for more information.

3.2.1.2.1.4 Disk



- **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.
- **Availability -** Indicates when the remote file will be available to users. The two valid values are Immediate (Disk) and Deferred (Tape).

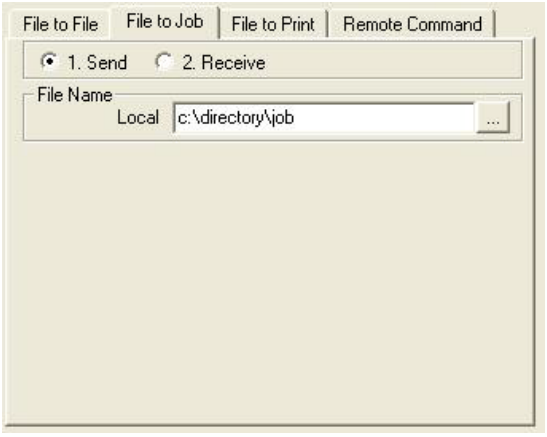
3.2.1.2.1.5 Other



- **Remove Trailing Spaces -** This option removes all spaces or binary zeros at the end of a record when transferred from the z/OS platform.

3.2.1.3 File to Job Tab

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

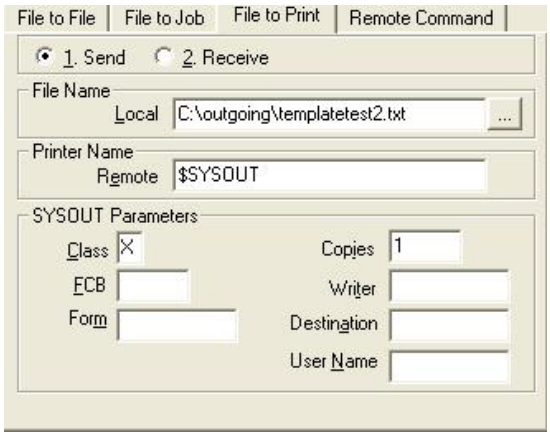


- **1. Send -** Initiates the send request to the remote system.
- **2. Receive -** Initiates the receive request from the remote system.
- **File Name -** The File Name field changes dynamically according to which direction the transfer is going. If the user specifies Send, the Local field appears in the File Name field. If the user specifies Receive, the Remote field appears in the File Name field.

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

3.2.1.4 File to Print Tab

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



- **1. Send -** Initiates the send request from the local system to the remote system.
- **2. Receive -** Initiates the receive request from the remote system.
- **File Name -** The File Name field in the File to Print tab changes according to which direction the transfer is going. If the user specifies Send, the Local field appears in the File Name field. If the user specifies Receive, the Remote field appears in the File Name field.

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

- **Printer Name -** The name of the printer to which the file is to be sent. This allows the user to send the file that is being transferred directly to the print queue or spool on the remote or local side.

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

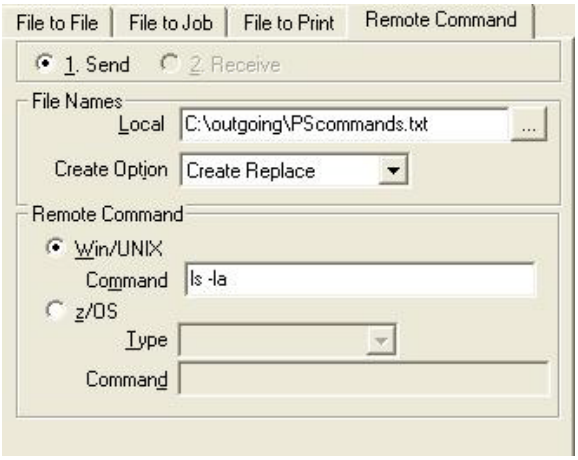
  To specify a *z/OS printer*, type SYSOUT@, where @ is the class to which you want to send the output. When specifying a z/OS printer, you can specify the SYSOUT parameters as described below.

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

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

3.2.1.5 Remote Command

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



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

Create	Create 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	Replace the contents of the destination file with the contents of the source file.
Append	Append the contents of the source file to the end of the destination file.
Create Replace	If the file does not exist on the system, then 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, then 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	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.

- **Remote Command –** In this section fill in the command that you would like executed remotely.

Win/UNIX - If you would like the command to be executed on a Windows or UNIX platform select this.

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

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

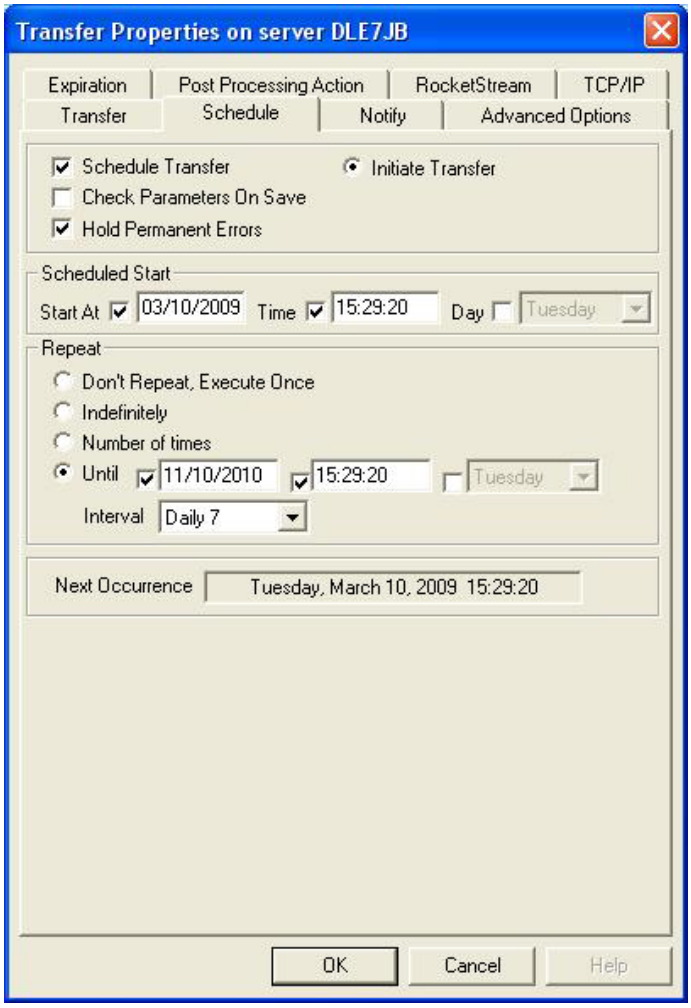
Type - This is the type of z/OS command that you would like executed.

Execute/REXX Exec	Allows the user to specify an exec command or a REXX exec for execution on the remote z/OS system.
Submit JCL	Allows the user to submit a job on the remote z/OS system. This differs from File to Job because the JCL to run actually sits on the remote system.
Call JCL	Allows the user to call a user program defined on the remote z/OS system.
Call Program	Allows the user to call a user program defined on the remote z/OS system.

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

3.2.2 Schedule Tab

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



- **Schedule Transfer -** Add (check) or delete (clear the check) schedules from the transfer. If a transfer is scheduled, that will take precedence over the Check Point/Restart option and what is input under the Expiration tab.
- **Check Parameters On Save -** Instruct the MFT Administrator to verify that the file that you have scheduled to be transferred exists at the time of scheduling.

  When the option is not selected, you can schedule a transfer of a file that does not exist: use this if you expect that the file will exist prior to the scheduled date of transfer execution. If the file does not exist when the scheduled transfer is executed, the server will notify you (as specified in the Notify property page) at the scheduled time of transfer execution.
- **Hold Permanent Errors -** This option will put a scheduled transfer on hold if a permanent error occurs. If this option is not selected the transfer would continue to be attempted even after a permanent error occurred. Examples of permanent errors would be the remote file not existing, bad user id or password, and expired license key.
- **Initiate Transfer -** Indicates that transfer will be initiated when the schedule becomes eligible.

- **Scheduled Start -** In the Scheduled Start fields, you can indicate when you want a file transfer to execute in the future. This parameter has three fields. In the first field, you specify the *date* on which you want to execute the transfer. In the second field, you indicate the *time* at which you want to execute the transfer. In the third field, you indicate the *day of the week* on which you want to execute the transfer.

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

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

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

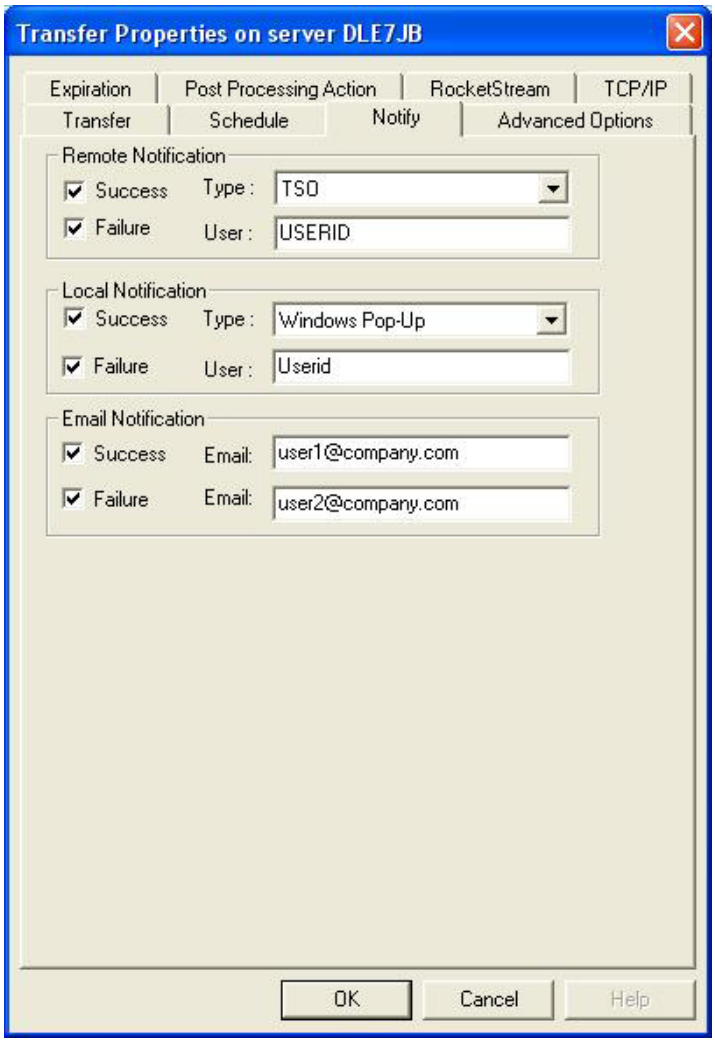
- **Repeat -** Provides information relative to the future execution (if any) of the particular file transfer after it has been executed once.

Don't Repeat, Execute Once	When this option is selected, the file transfer will be executed once, and then no longer attempted.
Indefinitely	When this option is selected, the Interval field appears on the panel. 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 Start At field and in the Interval field. This option specifies the number of times the file transfer can be executed before it is removed from the queue. The range for this field is unlimited with the exception of 0. The default is 10. Similar to the Indefinitely option, the Number of times option invokes the Interval field.
Number of times	You can use this option to specify the date, time and the day of the week up until which you want to execute the file transfer. When this option is selected, fields appear where you can specify the required information (similar to the Start At field).
Until	If you specify a Repeat option (with the exception of Don't Repeat, Execute Once), you can select this parameter. 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.
Interval	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).

If Scheduling is selected along with Check Point/Restart and Try Count, then, if for any reason your scheduled transfer fails during transmission, the transfer will only get sent at the next Scheduled date and time, it will NOT get sent as soon as the problem that caused the failure is resolved.

3.2.3 Notify Tab

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



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

Then set

- **Remote Notification**

  Type - Allows the user to select the means by which the remote user will be informed that the transfer has been completed. The five valid types are TSO, ROSCOE, Windows Pop-Up*, Email and NONE.

  For Mainframe only, when "TSO" type of notification is selected there is not a successful or a failure kind of notification. There is just notification, therefore if you select "TSO" in remote notification and submit the transfer and then look at the properties of the transfer, both success and failure will be checked.

  If NONE is selected, this indicates that there will be no notification upon completion of the transaction. **The USERID specified in the User field will not be notified should the Type field specify NONE.**

  User (or Email) - This is the name of the remote user to notify when a transaction is completed. It lets the user know whether the transaction was successful or not. If using email ensure you have completed the SMTP field under the General tab in the MFT Platform Server Properties panel.

- **Local Notification**

  Type - Allows the user to select the means by which the local user will be informed that the transfer has been completed. The three valid types are Windows Pop-Up*, Email and NONE.

  The USERID specified in the User field will not be notified should the Type field specify NONE.

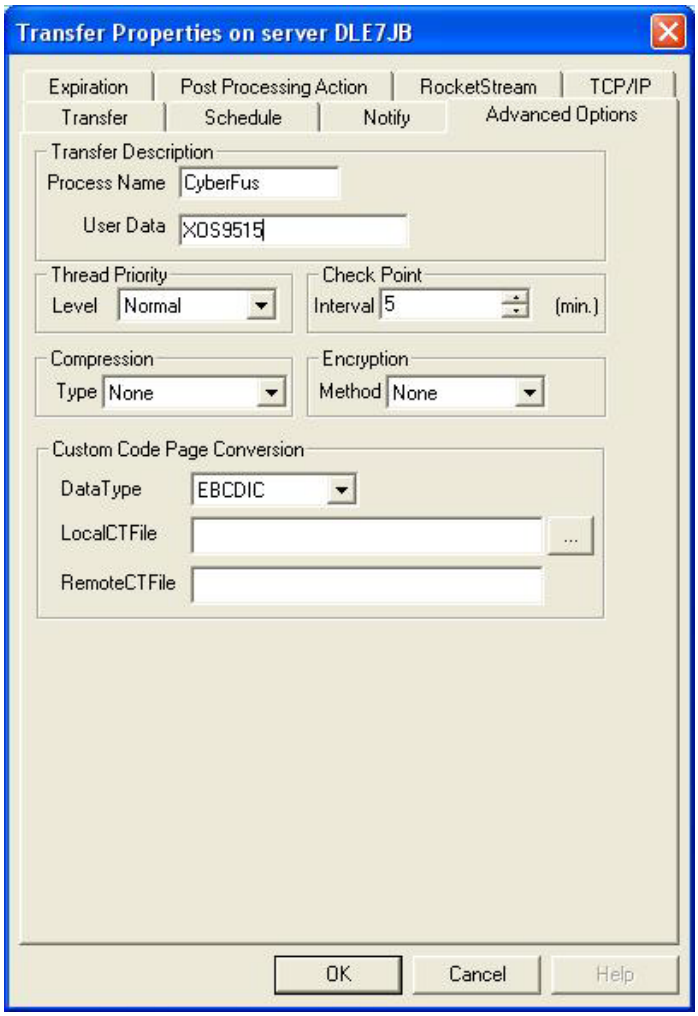
  User (or Email) This is the name of the local user to notify when a transaction is completed. It lets the user know whether the transaction was successful or not. If using email ensure you have completed the SMTP field under the General tab in the MFT Platform Server Properties panel.

- **Email Notification**

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

* Due to Microsoft ending support for the messenger service (messenger.exe) as of Windows Vista & Windows 2008 server; we are no longer able to support the notification type “Windows Popup”. This option will be removed from all future releases.

3.2.4 Advanced Options Tab



- **Transfer Description**

  Process Name - 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.

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

  User Data - Any alpha, numeric or national characters of up to 25 characters that will be logged into the history files that contain information that describe the transfer on the local and remote system. You can omit this parameter.

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

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

  Level - The levels of priority that can be assigned are as follows: highest, above normal, normal, below normal, lowest, and idle.

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

  Interval (in minutes) - The 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.

  Check Point Interval is specified in minutes and is a valid range of 1–90 minutes.

- **Compression -** Compression 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. Here's why:

  When you compress certain types of data, the compressed data will be 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 which would essentially be wasted.

  Type - MFT Platform Server for Windows provides two different compression algorithms: Limpel-Zev (LZ), Run Length Encoding (RLE), and ZLIB1 – ZLIB9. The user can select the algorithm which best suits their network. See the definitions below to make your choice. The default is None.

  MFT Platform Server for Windows will report to the Windows Event Log when the MFT Platform Server for Windows Initiator specifies *Compression* and communicates with a responder that does not support compression.

LZ	LZ 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.
RLE	RLE is more data-dependent than LZ. That is, the compression ratio may vary widely based upon the type of data being compressed. Choose RLE if you network bandwidth is not a critical bottleneck for your network and you need to save CPU cycles.
ZLIB1 through ZLIB9	ZLIB1 through ZLIB9 refer 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 and produces the best quality of compression. Level 7 (ZLIB7) typically offers the best compromise of compression and speed.
None	No compression will be used for this transfer.
Default	If Default is chosen then the type of compression will be taken from the Node setting or it will be set to None for non-Node transfers.

- **Encryption -** This parameter allows the user to turn encryption on and off. The method of encryption can be DES, Triple DES (3DES), Blowfish, Blowfish Long, Rijndael, None or Default.

DES (56 bit encryption)	This parameter allows the user to specify that DES encryption will be used for this transfer. DES (Data Encryption Standard) is a symmetric cryptographic algorithm, in which one secret key is used for encryption and decryption of the data being sent. DES uses a 56 bit encryption key.
Triple DES (112 bit encryption)	This parameter allows the user to specify that Triple DES encryption will be used for this transfer. Triple DES is just DES done three times with two secret keys applied in a particular order giving you 112 bit encryption.
Blowfish (56 bit encryption)	Blowfish is a block encryption algorithm that can use keys from 40 to 448 bits long. The MFT Platform Server implementation of Blowfish uses a 56 bit encryption key.

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.
Rijndael (256 bit encryption)	This parameter allows the user to specify that Rijndael encryption will be used for this transfer. Rijndael 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.
None	No encryption will be used for this transfer.
Default	If Default is chosen then the type of encryption will be taken from a Node that had been configured or it will be set to None for non-Node transfers.

- **Custom Code Page Conversion**

  Data Type - This is used to convert data between ASCII and EBCDIC. Select this when communicating with systems with defined data structures. This would not be necessary if you are communicating from PC to PC. When you change the comtblg.dat file you would need to stop and start the MFT Platform Server for Windows Service for the new conversion table to take effect if you are using MFT Platform Server for Windows.

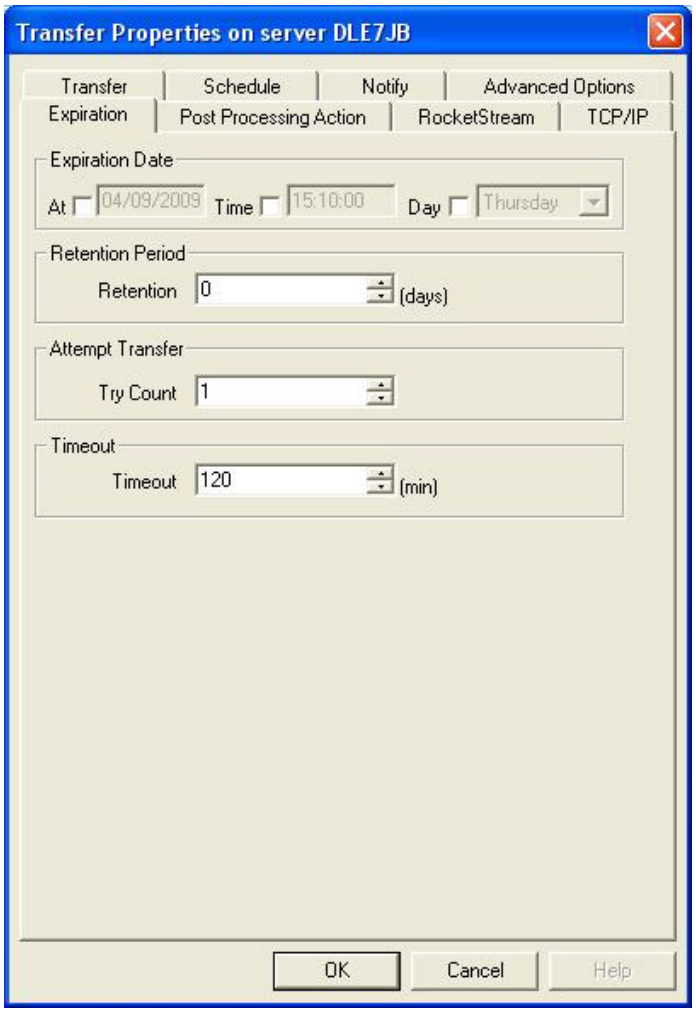
  ASCII - No translation is done from Windows to the remote system.

  EBCDIC - Normally used when transferring data with a z/OS or System i system.

  LocalCTFile - This parameter will contain the name of the file, which will be used to translate on the local side.

  RemoteCTFile - This parameter will contain the name of the file, which will be used to translate on the remote side. *Note: When defining the RemoteCTFile you must also define the LocalCTFile:NULL so no translation takes place locally.*

3.2.5 Expiration Tab



- **Expiration Date -** Specifies the exact date and time when a transfer should no longer be attempted. However, if this transfer was scheduled, that will take precedence over expiration. If Expiration and Retention are used, then whichever value occurs first will take precedence.

  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.

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

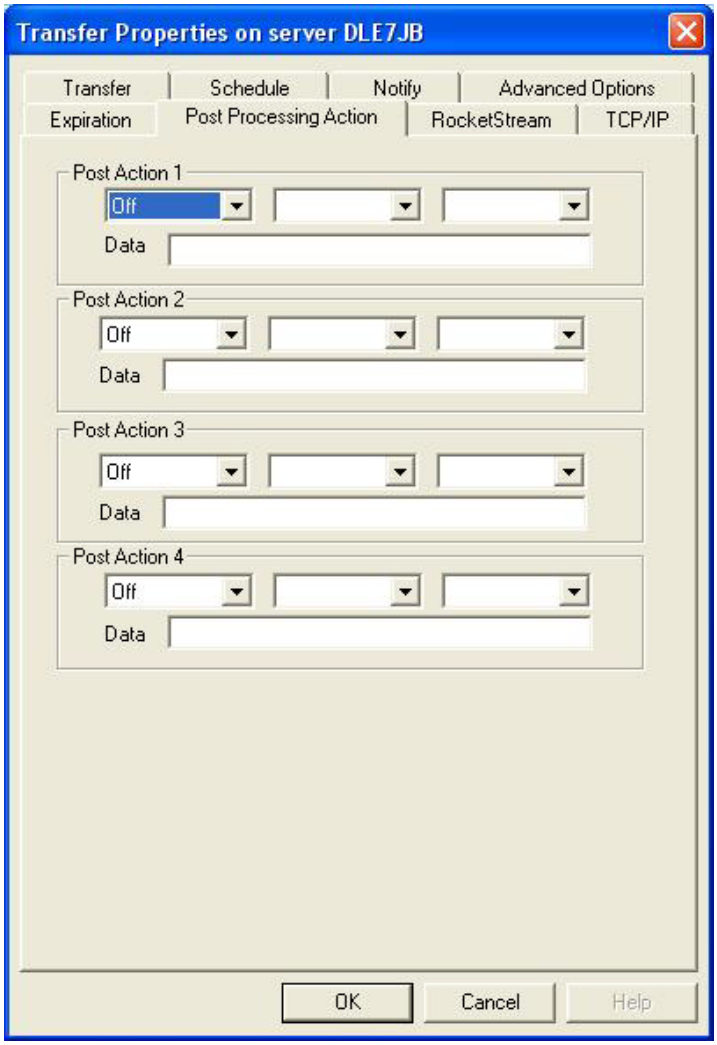
- **Retention Period -** Specifies the number of days that should pass from the transfer's start to the point it should no longer be attempted. If Expiration and Retention are used, then whichever value occurs first will take precedence.

- **Attempt Transfer: Try Count -** Specifies the number of times that MFT Platform Server for Windows will attempt the transfer. When the Try Count is reached, MFT Platform Server for Windows will no longer attempt 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.
- **Timeout -** Specifies the amount of time (minutes) a connection will stay open while waiting for a response from the remote side. Once the time is reached the connection is ended.

  Note - This parameter takes precedence over the Initiator Timeout on the Server Properties Window.

3.2.6 Post Processing Action Tab

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



- **Post Action 1 -** This is a command (.bat, .com, .exe, etc.) that will be executed upon the completion of the transfer.

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

  Field 2 - The values for this field are Initiator or Responder. Post Action 1 will be executed base on the source of the file transfer.

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

  Data - Defines the file that should be executed.

  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. Append a & 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 & sign to the data entered.

For example: C:\MyAction1.exe arg1=true #
C:\MyAction2.exe arg1=false &

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

3.2.6.1 Substitutable Parameters

MFT Platform Server supports Substitutable Parameters to allow users to take full advantage of the 64 character maximum on the command data, and to allow users to not have to copy the filename from the LocalFileName or RemoteFileName parameters. Note that we do 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. Below is a list of the substitutable parameters that are supported.

For our example, assume that we have a file called: C:\a\b\c\d\config.txt

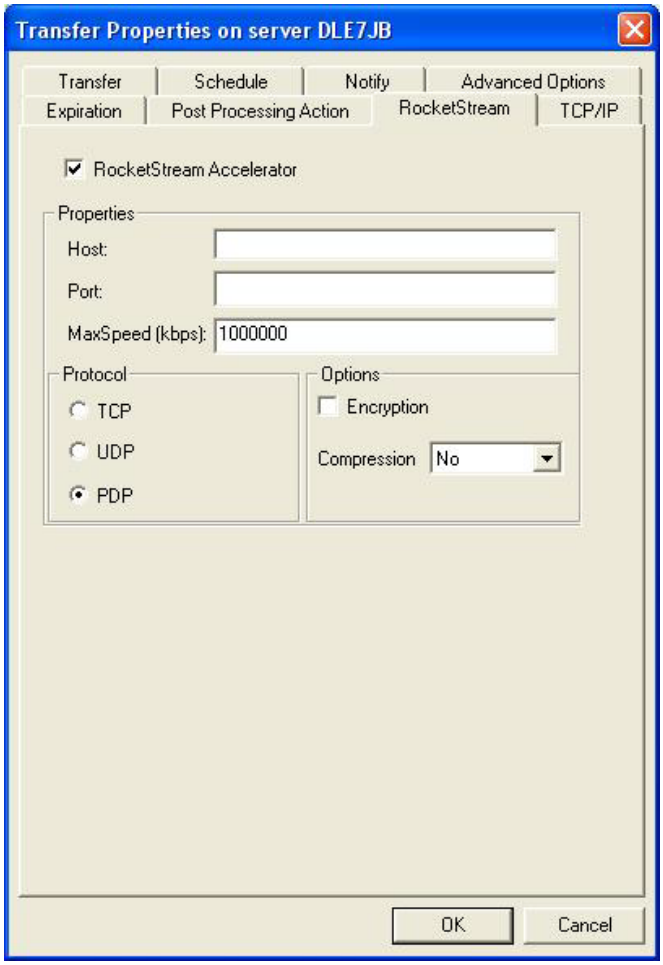
Substitutable Parameter	Description	Resolved Name Example
%DIR	Remote File Name directory without the file name or drive	a\b\c\d
%DRIVE	Remote File Name Drive	C
%NODRIVE	File name without Drive	a\b\c\d\config.txt
%SDIR	The lowest level directory	d
%HDIR	The high level directory	a
%NOSDIR	Directory name 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
%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

Note that 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. Note that 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 % sign. If UNIX detects the %DRIVE parameter, then the value DRIVE should be used as substitution. Similarly, %PROC becomes PROC and %UDATA becomes UDATA if not interacting with a z/OS system.

3.2.7 RocketStream Accelerator Tab

If you are licensed to use the RocketStream Accelerator technology and you want to set transfer requests to be sent using the RocketStream Accelerator protocols of UDP (User Datagram Protocol), PDP (Parallel Delivery Protocol) or TCP you would enable it by clicking on the RocketStream Accelerator box. You will see the properties panel will no longer be grayed out allowing you to configure the RocketStream Accelerator host and port your transfer request will be sent to.

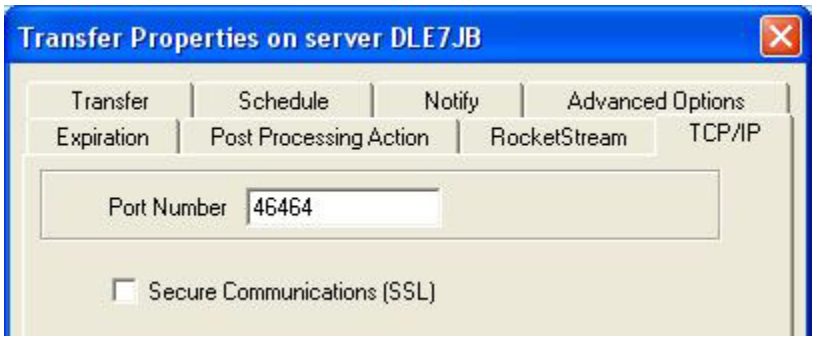
*Note: The standard RocketStream Accelerator port to use is **9099**. It is not recommended to use another port unless instructed by your local administrator.*



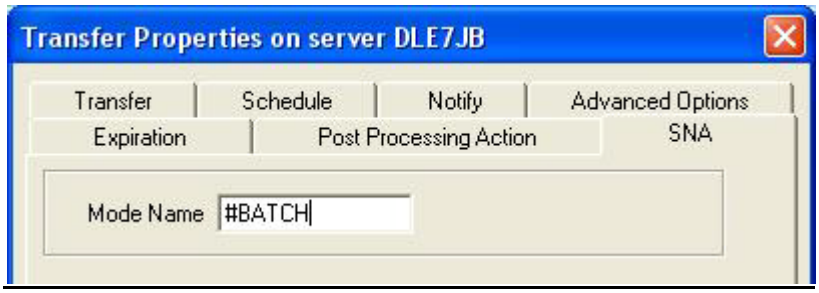
You can also configure the RocketStream Accelerator host to use, Encryption (Blowfish), Compression - [Best, Default, Fast] (This is a proprietary compression compatible with zlib), or a Max Speed in Kilobytes per second your transfer request should be set to use.

Note: It is not recommended to use MFT Platform Server Compression with the RocketStream Accelerator compression. One or the other should be used.

3.2.8 TCP/IP and SNA Tab



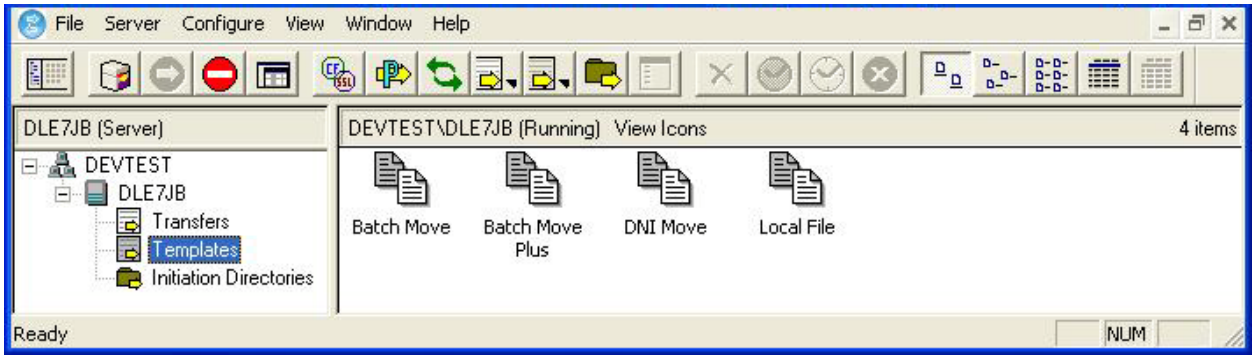
- **Port Number -** This is the secondary network address for the TCP/IP transfer. In TCP/IP networks, applications choose a specific port number for transactions so they do not conflict with other applications at the same TCP/IP address. By default, MFT Platform Server for Windows uses 46464. If other applications on your network use this port number, use a different port for your MFT Platform transfers.
- **Secure Communications (SSL) -** This check box is selected when SSL is to be used. The SSL port number should also be entered in the Port Number field if SSL communication is desired. Please refer to the section on SSL for more information.



- **Mode Name**

  The name used to represent a set of characteristics to be used in an APPC LU-LU session. This mode name must be configured to HIS.

3.3 The Network View



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

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

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

3.3.1 Buttons

3.3.1.1 Create a New Network View button

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

3.3.1.2 Add a New Server button

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

3.3.1.3 Start a MFT Platform Server button

 Click this button to start a server.

3.3.1.4 Stop a MFT Platform Server button

 Click this button to stop a server.

3.3.1.5 The View/Change Server Properties button

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

3.3.1.6 SSL Settings button

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

3.3.1.7 Configure Post Processing button

 This turns on the Configure Post Processing feature. Check the box to turn on this feature and enter the name of the file to be used for the post processing.

3.3.1.8 Refresh View button

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

3.3.1.9 New Transfer button

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

3.3.1.10 New Template button

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

3.3.1.11 New Initiation Directory button

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

3.3.1.12 Update Properties button

 Click this button to view or change the parameters of a specific Transfer, Template, or Directory Named Initiation entry. The Properties panel will display. Modify the fields and select OK.

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 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 (see *The Event Log* Appendix).

3.3.1.13 Delete Selected Objects button

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

3.3.1.14 Hold button

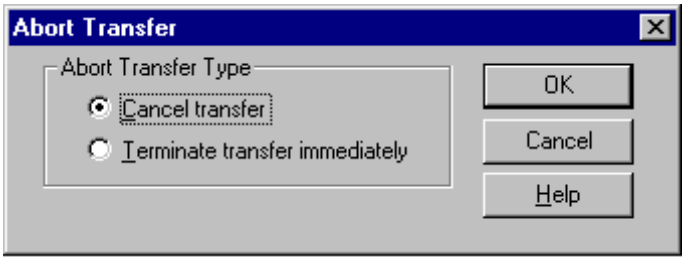
 Click this button to put a hold request on a transfer or Directory Named Initiation entry so that it cannot be dispatched. This action will prevent the Schedule Dispatch Service from initiating the transfer until otherwise notified.

3.3.1.15 Release button

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

3.3.1.16 Abort button

 Click this button to abort a transfer. The Abort Dialog panel will appear.



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

Cancel transfer will notify the remote system that the transfer has been terminated.

Terminate transfer immediately will terminate the transfer and not notify the partner. In certain instances, this selection can stop a transfer that Cancel transfer cannot.

3.3.1.17 View Items as Large Icons button

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

3.3.1.18 View Items as Small Icons button

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

3.3.1.19 View Items in a List button

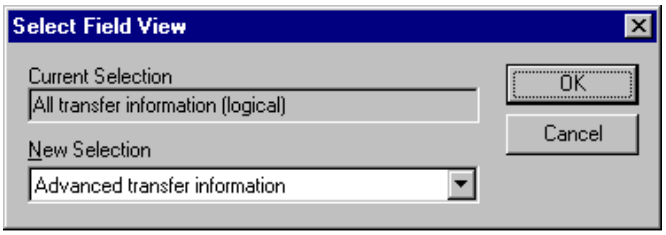
 This button will cause a single small icon for each file transfer to display with its Transfer ID directly next to it. The way the icon appears differs depending upon the file transfer type selected.

3.3.1.20 View Items in Detail button

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

3.3.1.21 Select Field View button

 Click this button to select which fields you would like to view from a predefined group. The panel appears as shown below.



- **Current Selection**

  This field names the predefined group you are viewing.

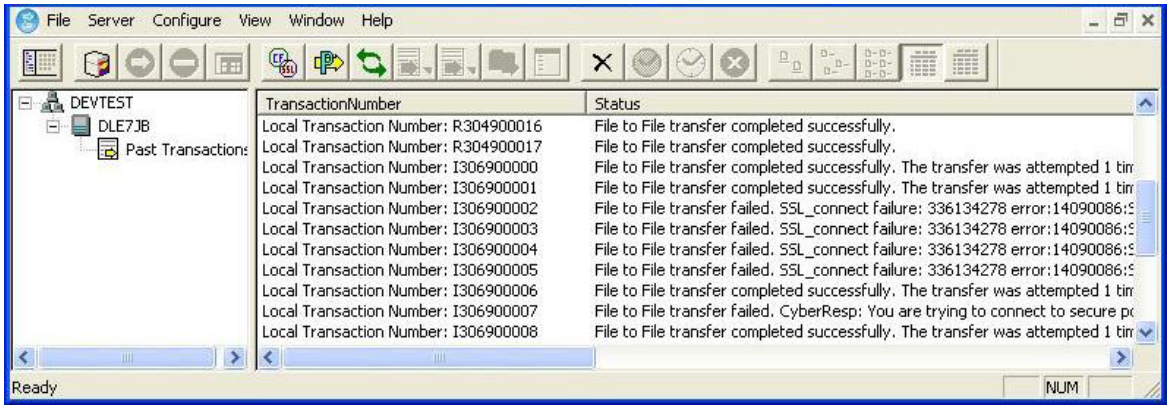
- **New Selection**

  Use this field to change the predefined group. These groups are defined below.

All (alphabetical)	All fields on the queue are arranged alphabetically.
All (logical)	All fields are displayed in a logical sequence.
General	A short list of fields that are common to all file transfers are displayed.
File to File	Only those fields relative to File to File transfers are displayed.
File to Print	Only those fields relative to File to Print transfers are displayed.
File to Job	Only those fields relative to File to Job transfers are displayed.
Remote Command	Only those fields relative to Remote Command executions are displayed.
Advanced	Only those fields that appear on the Advanced panel are displayed.
Status	Only those fields that are seen in the Initiated Transfers window are displayed.
Schedule	Only those fields that appear on the Schedule panel are displayed.
z/OS Parameters	Only those fields that appear on the z/OS options panel are displayed.
SNA Parameters	Only those fields that are SNA specific are displayed.
TCP/IP Transfer	Only those fields that are TCP/IP specific are displayed.

3.3.2 Past Transactions

The MFT Platform Server Administrator Past Transactions feature helps the user to see the status of transfers, which have been completed. The transfers can be viewed by selecting **View** from the drop-down menu then selecting **Past Transactions**. The user needs to add the server to view status of the previously completed transfers on that server. This is exactly the same as adding a server in a network view in the Administrator. The transfers in this particular window cannot be viewed by double-clicking on the transfer. The status of these transfers is pulled from the Event log of the respective server. Therefore if the user clears his event log, the past transactions will also be deleted.



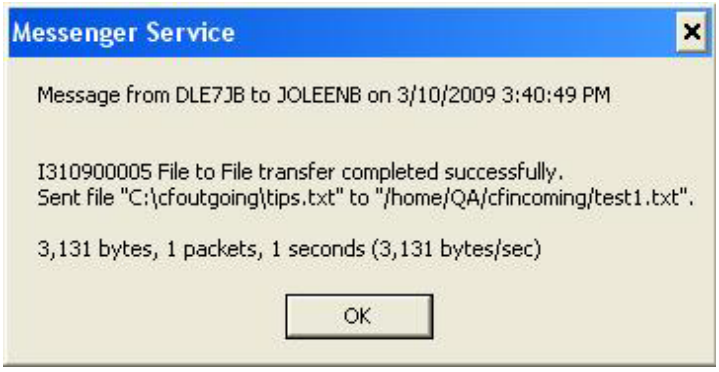
A backup event log on the server is created every time the user opens or refreshes the Past Transactions panel. This backup is in a file called c:\temp\tmp.evt. The user can read from the backup event log by selecting **Open Backup Eventlog** from the **File** drop-down menu in the Administrator. The user has to select the server and only then will this menu-item be enabled. The user can delete this file but in that case there will be no transactions available when the user opens the backup event log. The user can sort the transactions by clicking on any column header.

3.3.3 Notification

Once a file transfer has been submitted and executed, there are several ways that you can be notified about the transfer's success or failure.

3.3.3.1 Server Issued Message to Windows User

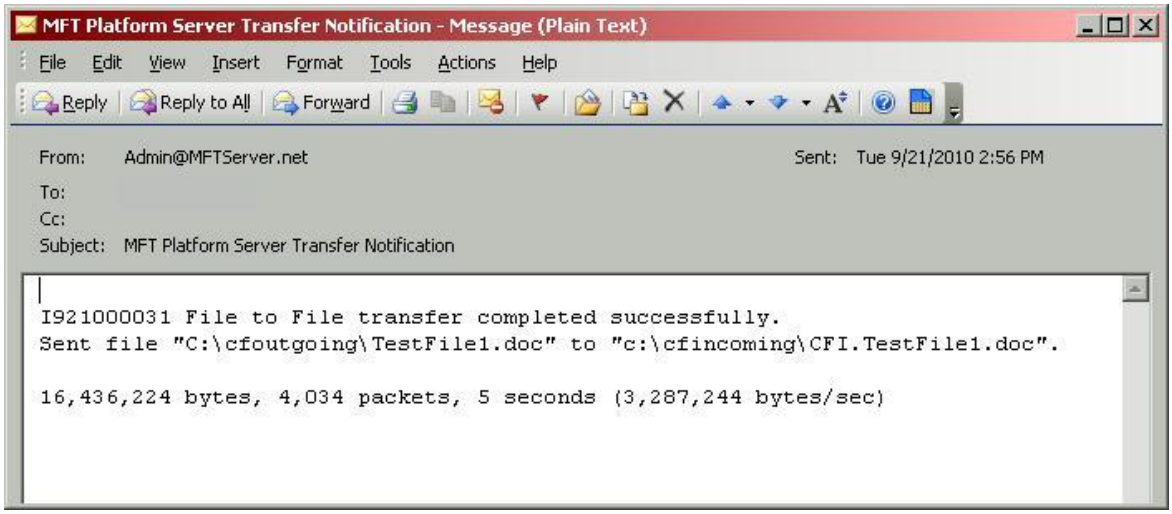
Upon completion of a file transfer, the server issues a pop-up message* to inform the Windows user about the outcome of the file transfer. The message also indicates the name of the local file was that was sent, the name of the remote file to which it was sent, the file size, and the length of time it took to transfer that file.



* Due to Microsoft ending support for the messenger service (messenger.exe) as of Windows Vista & Windows 2008 server; we are no longer able to support the notification type "Windows Popup". This option will be removed from all future releases.

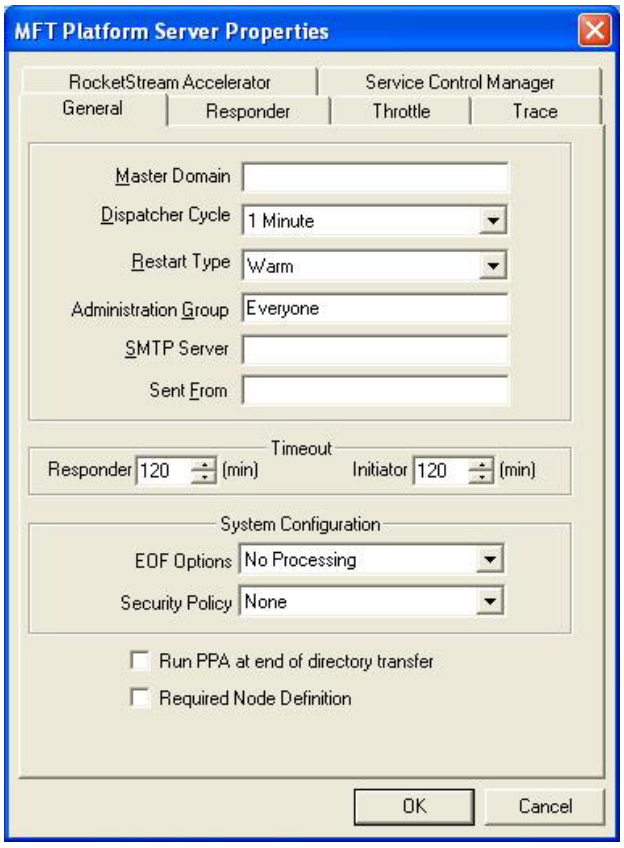
3.3.3.2 MFT Platform Server Email Notification

Upon completion of the file transfer, MFT Platform Server will send an email to the address specified in the Notify tab under the Options button. The email will be similar to the following:



3.4 Server Properties

3.4.1 General properties page

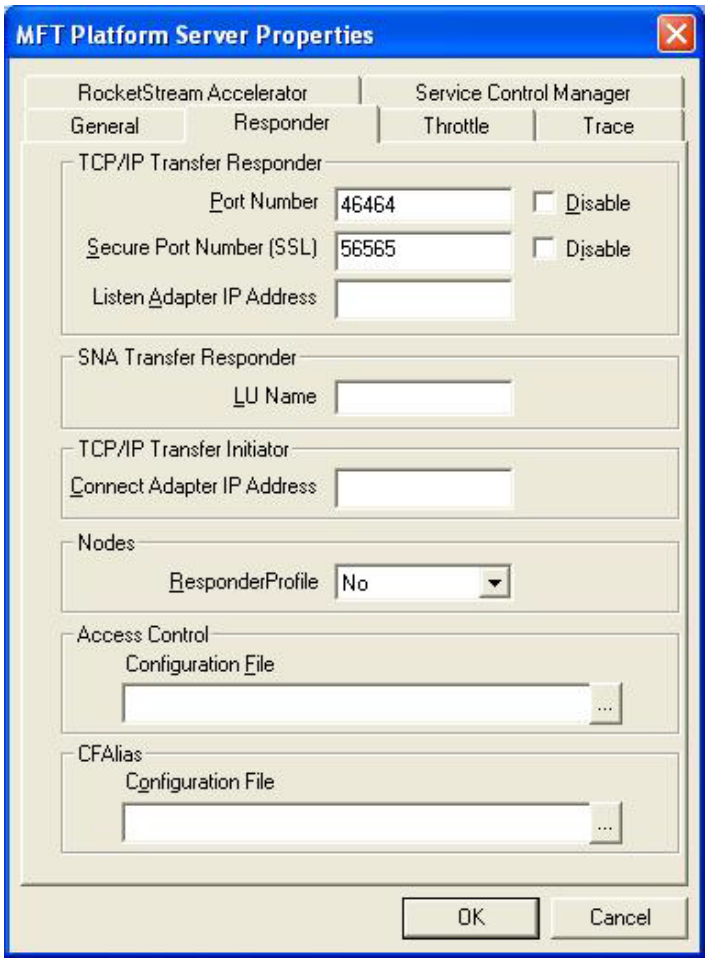


Note: A user who can view the server status but does not have the authority to start or stop the server will see the connection status set to "ServerName - Connect Query Only." This user cannot change the server settings on this panel. The MFT Platform Server Properties panel will appear as Read-Only. The OK and the Cancel button are replaced by a Close button.

Parameter	Description
Master Domain	In this field, enter the name of the domain that you wish 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 your user id without a domain name preceding it.
Dispatcher Cycle	This parameter specifies the time that the scheduled dispatcher service will wait 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, 1hr., 2hr., 4hr., 8hr., 12hr., 24hr. The scheduled dispatcher service writes the date and time to the MFT Platform Server Monitor when it checks schedules for eligibility.
Restart Type	**Warm** - Specifying warm start means that all of the transfers that are in the persistent work queue are retained when MFT Platform Server is restarted. **Cold** - Specifying cold start indicates that all 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.

	Note: A Cold Start will delete your DNI definitions as well as any queued or active transfers
Administration Group	The name of the email server that will be 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
SMTP Server	The name of the email server that will be 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	This field identifies the name displayed in the email notification. This value can not contain any spaces.
Timeout: Responder	Specifies the amount of time (minutes) a connection will stay open while waiting for a response from either the Initiator or the Responder. Once the time is reached the connection is ended. Valid values 1 to 1440: Default 120 (2 hours)
Timeout: Initiator	Specifies the amount of time (minutes) a connection will stay open while waiting for a response from either the Initiator or the Responder. Once the time is reached the connection is ended. Valid values 1 to 1440: Default 120 (2 hours)
System Configurations: EOF Options	This parameter defines what permitted action (whether a Carriage Return Line Feed (CRLF), End of File (EOF), or both will be added to records) will take 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.
System Configurations: Security Policy	This parameter defines whether this MFT Platform Server will enforce HIPAA or FIPS-140 regulations on initiated and responding transfers. **HIPAA** – This setting requires MFT Platform Server to comply with HIPAA standards. At this time the standards require that all files are transferred using encryption key length that will be 128 bits or greater. **FIPS-140** – This setting 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 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 read Section 4.12. To comply with the security policies of HIPAA or FIPS-140 transfer requests configured incorrectly, for example a transfer using an encryption type of DES which is not allowed for either HIPAA or FIPS-140, the encryption would be over ridden and to comply with HIPAA, if you set your encryption to DES a pop-up message would be displayed informing you the encryption will be changed to 3DES, if you set the encryption to Blowfish a pop-up message would be displayed informing you the encryption will be changed to Blowfish Long. If you were using FIPS-140 you would receive a pop-up message informing you the encryption will be changed to Rijndael (AES) when a transfer is initiated.
Run PPA at end of directory transfers (Directory Transfer or Distribution List Transfer)	This parameter defines when a directory transfer is done 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.
Required Node Definition	Required Node Definition offers the ability to restrict the remote systems that the server will accept transfer requests from. In order to be accepted, incoming requests must come from remote systems that are defined in the cfnode.cfg file. All others will be refused. Please refer to the sections on Nodes, User Profiles and Responder Profiles for details.

3.4.2 Responder Property Page



This page provides for configuration of the responder Port Number and Responder LU Name.

- **TCP/IP Transfer Responder**

  Port Number - 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 may 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 may be reserved for standard applications at your installation.

  Disable - This check box will turn the regular TCP/IP port number on or off.

  Secure Port Number (SSL) - This is 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 may be reserved for standard applications at your installation. For more information on SSL, please refer to the SSL section of this document.

  Disable - This check box will turn the SSL port number on or off.

- **SNA Transfer Responder**

  LU Name - MFT Platform Server responders to transfers using SNA which are routed to the machine associated with a Local LU defined in SNA Server as opposed to the network address of the machine. If there is a specific LU Name given here, that server will respond to SNA requests routed to that LU. If the field is left blank, MFT Platform Server for Windows will respond to SNA requests routed to ANY LU. (There is no default for this value.)

- **TCP/IP Transfer Initiator**

  Connect Adapter IP Address - If a machine has more than one IP Address it is possible to bind the connection to a particular one. It can guarantee that all MFT Platform Server transfers will go only through this particular IP Address. The default value for this parameter is ALL what means bind to any IP Address. If this parameter is defined, the initiator will send/receive data for outgoing connections only through this address.

- **Nodes**

  Responder Profile - Responder Profiles define a local username and password that should be 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.

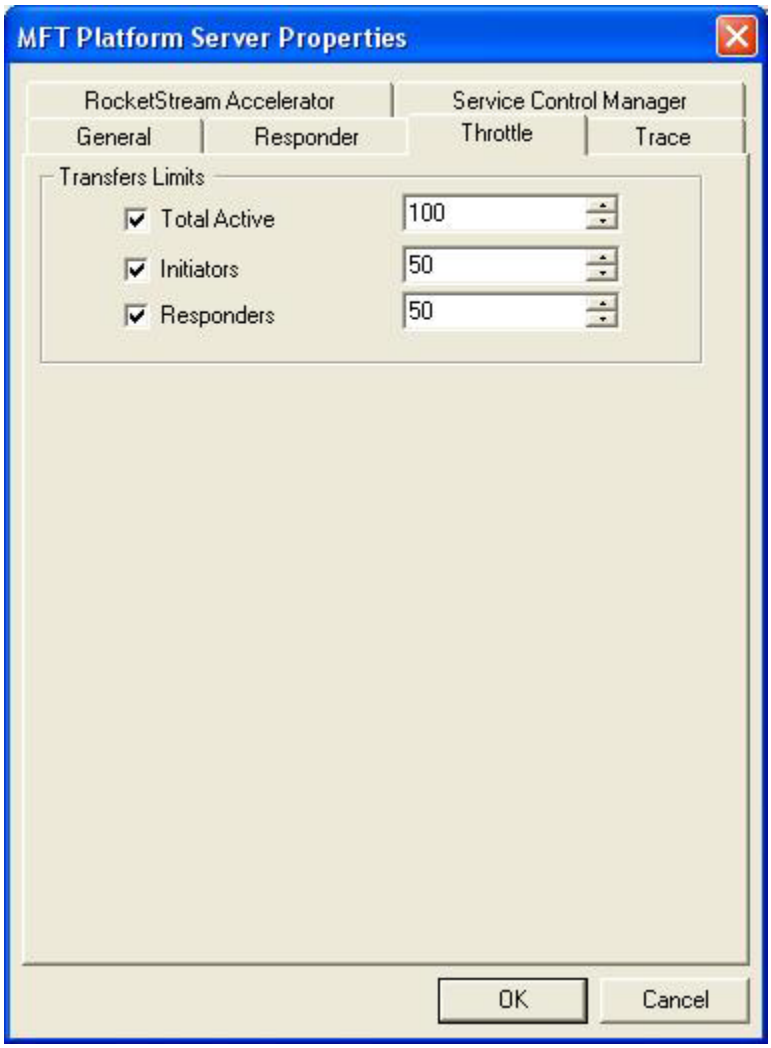
- **Access Control**

  Configuration - Users can now send a file to the Windows platform and it will automatically go a pre-defined directory based on user-defined criteria. The default file name for the Access Control configuration is called AccessControl.cfg. Please refer to the section on Access Control for more information. This is used by the MFT Platform Server Responder only.

- **CFAlias**

  Configuration - Users can now send a file to the Windows platform and it will automatically go a pre-defined directory based on user-defined criteria. The default file name for the CFAlias configuration is called CfAlias.cfg. Please refer to the section on CfAlias for more information. This is used by the MFT Platform Server Responder only.

3.4.3 Throttle Properties Page

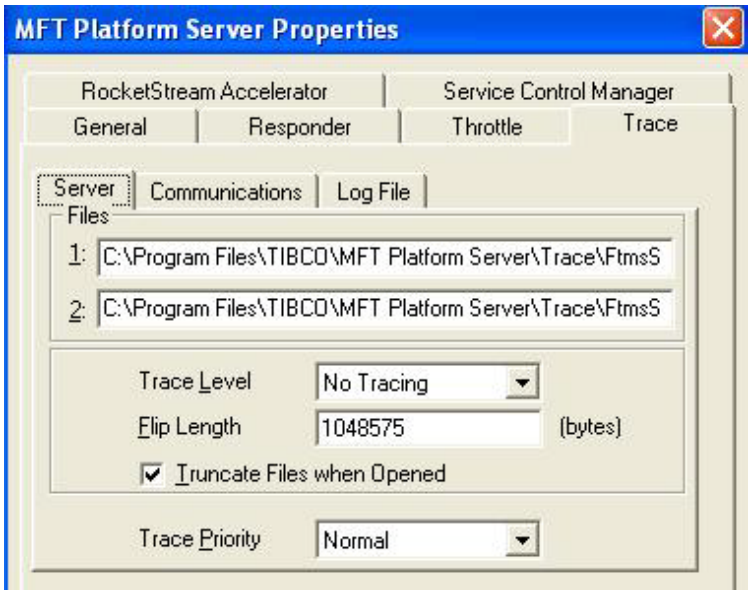


Server throttling limits user activity. You can use this property sheet to limit total active transfers, initiators and responders.

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

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

3.4.4 Trace Property Page



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

Note: Tracing should only be turned on at the request of TIBCO Technical Support.

Each tab contains information for each trace file generated by the MFT Platform Server.

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

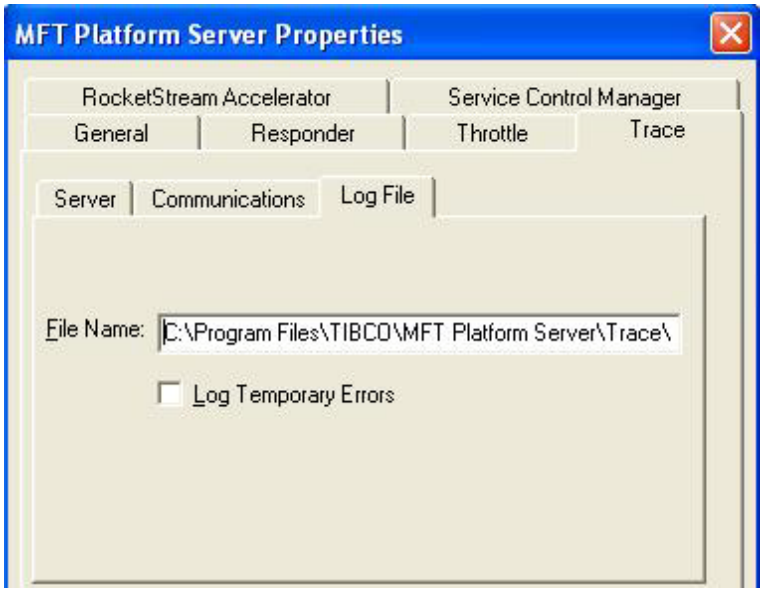
Use the Communications tab to trace the Server and, specifically, the communications layer which is activated during transfers. The information contained in this trace file shows exactly what is being transmitted and received across the network during a transfer.

Use the Log File for web administration. This will allow you to view past transactions through the MFT Command Center. A screen shot of the Log File tab follows the list of parameters below.

- **Trace File 1 -** This field indicates which file to use for the first flip file.
- **Trace File 2 -** This field indicates which file to use for the second flip file.
- **Trace Level -** 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 -** This is the maximum amount of information (in bytes) that will be written before the trace files flip. This value should not be less than 1024.
- **Truncate Files when Opened -** When the application or server starts, it can clear out (truncate) the trace files before it begins to write information. If this option is TRUE, then the trace files are truncated when the program starts. Otherwise, it opens the existing files and appends the information to the end.
- **Trace Priority -** While the fields described above apply separately to each trace file, this field applies to all of the trace files at the same time.

  This field indicates the priority given to the thread that is responsible for receiving and formatting the trace information from the system. Increase this value if it appears that the system is generating trace information that exceeds the system's ability to write the information to the trace files. Tracing should only be turned on at the request of TIBCO Technical Support.

3.4.4.1 Log Tab



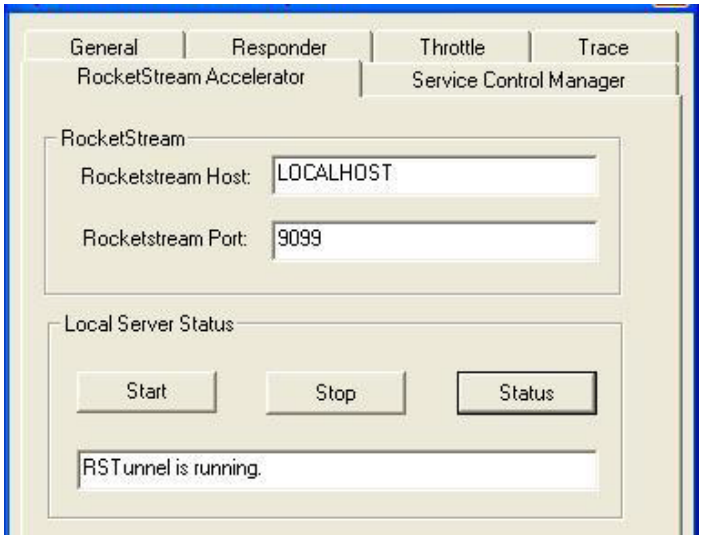
- **File Name**

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

- **Log All Transfer Attempts**

  Check box to set Log All Transfer Attempts to on. Setting this to off (leaving the box unchecked) will cause MFT Platform Server to log only the final transfer attempt in a restart situation.

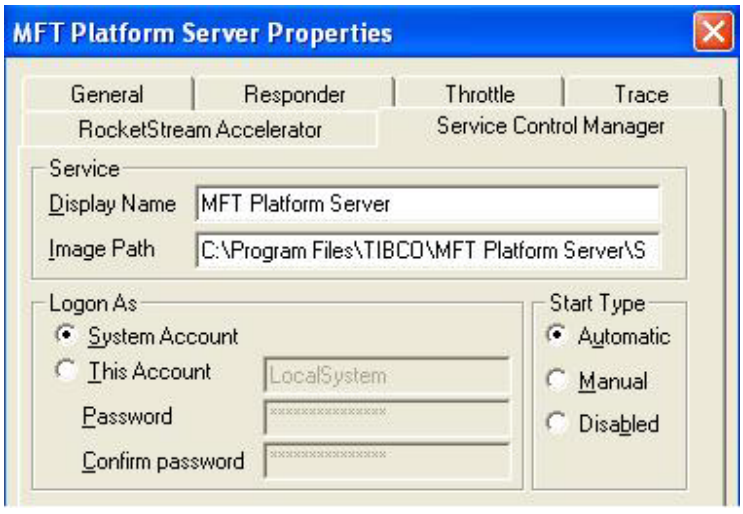
3.4.5 RocketStream Accelerator



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

- **RocketStream Accelerator Host -**This is the Hostname or IP of the RocketStream Host.
- **RocketStream Accelerator Port -** This is the port number the RocketStream Accelerator is listening on. Default is port 9099.
- **Local Server Status –** This section will allow you to Start and Stop the RSTunnel Service as well as allow you to display the current status if the service.

3.4.6 Service Control Manager Property Page



Use the SCM property page to 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 panel allows maintenance of both types of service.

- **Display Name -**This describes the service shown in the Windows Service Control utilities. If not given, those tools display "MFT Platform Server" as the service description.

- **Image Path -** This is the full name to the executable for the service. For MFT Platform Servers, this shows ….\ftmssvr.exe
- **Logon As -** Indicates the user id (local system or specified user) and password that Windows uses to start the MFT Platform Server Service.
- **Start Type -** Indicates how the service should be started.

  Automatic: starts when the system reboots (recommended setting)

  Manual: starts when the administrator tells it to start.

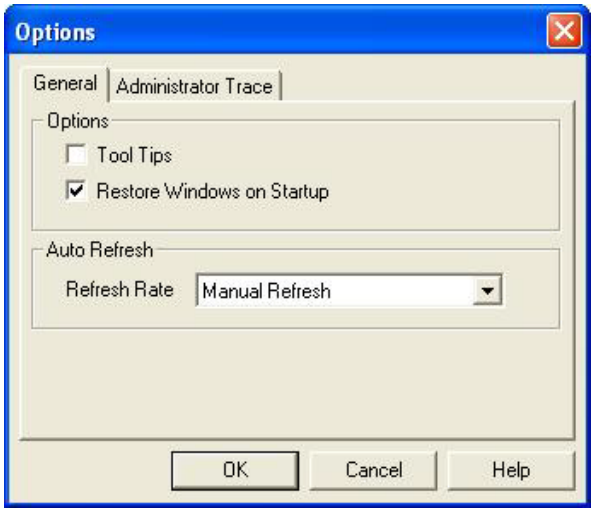
  Disabled: prevents the service from ever starting.

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

3.4.7 View Menu—Options Property Sheet

The Options property sheet is available from the MFT Platform Server Administrator View menu. You can also access this property sheet if you right-click (click the right mouse button) in an un-written window space.

3.4.7.1 General Property Page



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

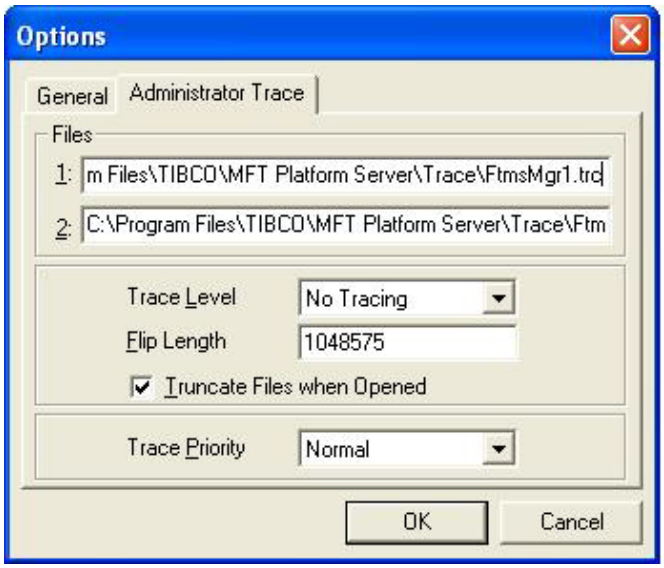
  The available options are:

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

- 5 Minutes—the refresh occurs every 5 minutes
- 10 Minutes—the refresh occurs every 10 minutes
- 30 Minutes—the refresh occurs every 30 minutes
- 60 Minutes—the refresh occurs every 60 minutes

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

3.4.7.2 Administration Trace Property Page



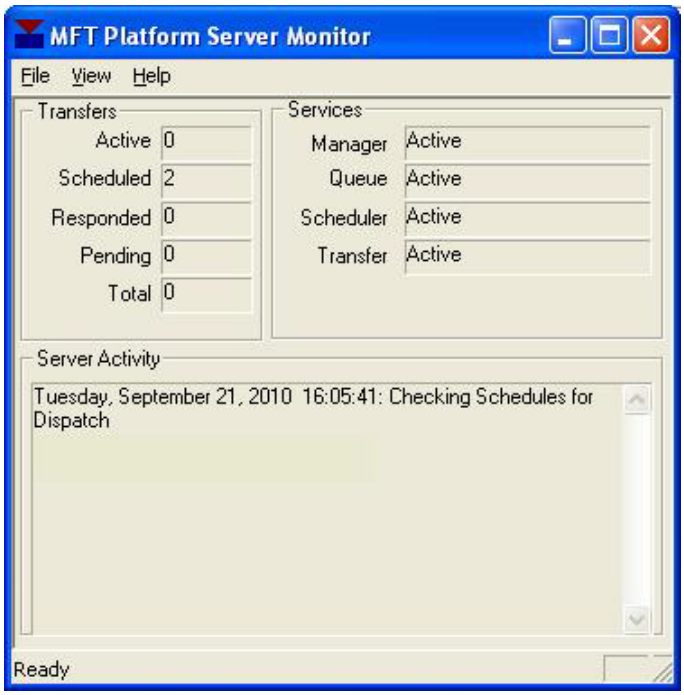
Use this property page to configure the tracing for the MFT Platform Server Administrator application.

Note: This panel is for configuring the MFT Platform Server Administrator *locally*. Tracing should only be turned on at the request of TIBCO Technical Support.

- **Trace File 1 -** This field indicates which file to use for the first flip file.
- **Trace File 2 -** This field indicates which file to use for the second flip file.
- **Trace Level -** 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 -** This is the maximum amount of information (in bytes) that will be written before the trace files flip. This value should not be less than 1024.
- **Truncate Files when Opened -** When the application opens, it can clear out (truncate) the trace files before it begins to write information. If this option is TRUE, then the trace files are truncated when the program starts. Otherwise, the application opens the existing files and appends the information to the end.
- **Trace Priority -** This field indicates the priority given to the thread responsible for receiving and formatting the trace information from the system. Increase this value if it appears that the system is generating trace information that exceeds the system's ability to write the information to the trace files. Tracing should only be turned on at the request of TIBCO Technical Support.

3.5 MFT Platform Server Monitor

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



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

3.5.1 Functions

From the MFT Platform Server Monitor's View menu, you can choose the following:

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

3

4. Command Line Interface

The Command Line Interface allows a user 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
- Environment Variables on the Command Line

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.

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

4.1 Command Line Format

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

FTMSCMD /SEND /FILE [parameters] "c:\local\file\name.txt" "remote.file.name"
In this example, no environmental variables were used. This would require you to specify the mandatory parameter in the parameter section: LU_NAME.

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

SET LU_NAME=*luname*

SET CR_LF=*no*

SET REMOTE_USER_ID=*userid*

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

Here, the mandatory parameters were specified in the environmental variables. You would not have to specify parameters in the parameter section; however, you could still specify any of the additional parameters in the parameter section or in the environmental variables.

The environment variable setting stays active until you change it or remove it using the set command with no value specified (for example: SET CR_LF=).

4.1.1 Specifying Command Line Parameters

To set a command line argument, use this syntax:

FTMSCMD [parameters] "local_file_name" "remote_file_name"
Options can include any number of the following forms:

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

   /option (DOS Standard)

   -option (UNIX Standard)

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

4.2 File to File Transfers

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

- 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

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

Send	Indicates that the file will be sent from the local to the remote system.
Receive	Indicates that the file will be retrieved from the remote system.
Submit	This parameter is used in conjunction with the FS:ServerName parameter to submit a transfer to another MFT Platform Server. Specify the transfer parameters as you normally would on the command line. If the Submit button is selected and a server name is not specified (/fs:ServerName), an error is returned.

File	Store the contents of the file transfer in a file. This is the default.
Print	Send the file being transferred directly to the print queue or spool on the remote side.
Job	Send a local file to the remote system, where the partner will execute it as a batch job.
Remote Command	Executes a command on a remote system. Output will be stored in a local file specified by the user. Note that when the remote system is z/OS, the output is not returned.

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

LU_NAME	A port through which end users of a network communicate with each other. The Logical Unit handles and enforces the protocols required for end user-to-end user communications. Any valid LU name in the network can be used in this field. This is the same as the destination field in the GUI.
LOCAL_FILE_NAME	The name of the file on the local system that is going to be involved in the transfer.
REMOTE_FILE_NAME	The remote filename of the virtual filestore on the remote system that is the subject of the activity. It 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.

Example of Sending a File to a Remote System

```
FTMSCMD /S /F /LU:LUNAME /DT=BINARY /RL=1 /RI=USERID /RW=pswd
“F:\JOHN\QA\ONEX1.BIN” “JTPLM.QAL.BATCHB.ONEX1”
```

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”
```

4.3 File to Job Transfers

This section describes how to transfer a file and have the output of the transfer execute as a job.

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

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

This transfer can be in either direction (receiving a file from the remote side and having it execute on a local side or sending a file and having it execute on the remote side). The file name will depend on which way the transfer is occurring. For example, if you are receiving a file from the remote side and having it execute on the local system, you would specify the name of the remote file in the example above where it says *file name*. There is no need to specify a local file name since the output will not be written to any local file.

If you are sending a file to the remote side and having it execute on the remote system, specify the *local file name* in the example above where it says *file name*.

Example of Sending a Job to a Remote System

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

Example of Receiving a Job to a Remote System

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

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

4.4 File to Print Transfers

This section describes the print transfer function. See the File to Print Tab section for a more detailed description of the available parameters that you can specify when using this function.

To print the output of the transfer to the destination printer, you must specify the positional parameter /P. This is done similarly to the way that the file's positional parameters are specified when you perform a file to file transfer.

FTMSCMD [parameters] /SEND /PRINT /REMOTE_PRINTER_NAME=printername file_name

This example illustrates a file transfer whose output will be directed to a printer. Since 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 file name will depend on which way the transfer is occurring. For example, if you are receiving a file from the remote side and printing it to a local printer, you would specify the *remote file* name in the example above where it says *file name*. You do not have to specify a local file name because the output will not be written to any local file.

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

SET REMOTE_PRINTER=*printername*

FTMSCMD [parameters] *file_name*

In this example, the mandatory parameters were specified in the environmental variables. You would not *have to* specify any parameters in the parameter section; however, you *could* specify any of the additional parameters in the parameter section or in the environmental variables.

4.4.1 How to Specify the Printer Name

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

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

REMOTE_PRINTER_NAME

Default	Not Applicable
Allowable Values	Any combination of up to 255 characters
Minimum	1 character
Maximum	255 characters

This is the name of the printer to which the job will be sent.

SYSOUT_CLASS

Default	NONE
Allowable Values	0–9,A-Z
Minimum	0 or A or a
Maximum	9 or Z or z
Alternate Specification	CL

SYSOUT Class describes to which class the JES output will be routed. On z/OSs, 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 need to be told by the z/OS staff what values to supply.

SYSOUT_COPIES

Default	NONE
Allowable Values	1–999
Minimum	1
Maximum	999
Alternate Specification	SP

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

SYSOUT_DESTINATION

Default	NONE
Allowable Values	1–8 characters
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	SD

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

SYSOUT_FCB

Default	None
Allowable Values	1–4 characters
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	SB

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

SYSOUT_FORM

Default	NONE
Allowable Values	1–8 characters
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	SF

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

SYSOUT_USERNAME

Default	NONE
Allowable Values	1–8 characters
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	SI

This is the username assigned to a job submitted to the internal reader.

SYSOUT_WRITER

Default	NONE
Allowable Values	1–8 characters
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	SW

This indicates the external writer name that will be used to process this printer file on the z/OS. This is the name of a service program on the z/OS, which will be given control when it is time to process this file from the printer queue. The service program, which is written by the customer, decides how it wants to process this print file. Do not specify a value for this parameter unless directed to by the systems analyst on the z/OS.

4.5 Remote Command Transfers

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

To have a command execute remotely, specify the positional parameter /COMMAND 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 will be used to store the output of the remote command if the remote system is Windows or UNIX. z/OS does not send back output.

Example of executing a Remote Command on a Remote System

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

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

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

4.6 Parameters

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

All of the parameters below, with the exception of local filename and remote file name, can be specified both as environmental variables and as parameters on the command line. Each of the parameters can be specified in three different ways, all of which are valid both on the command line and as an environment variable. For example, data type can be specified as **DATA_TYPE**, **DataType,** and **DT**.

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

4.6.1 Optional Parameters

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

ALLOCATION_TYPE={ TRACKS | CYLINDERS | MEGABYTES | KILOBYTES }

Default	TRACKS
Allowable Values	TRACKS, CYLINDERS, MEGABYTES, KILOBYTES
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	AllocationType, AT

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

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

ALLOCATION_PRIMARY

Default	Not Applicable
Allowable Values	numeric value
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	AllocationPrimary, AP

The primary allocation quantity in tracks, cylinders, kilobytes or megabytes as indicated in the allocation type field.

ALLOCATION_SECONDARY

Default	Not Applicable
Allowable Values	numeric values
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	AllocationSecondary, AS

The secondary allocation quantity in tracks, cylinders, kilobytes or megabytes as indicated in the allocation type field.

BLOCK_SIZE

Default	Not Applicable
Allowable Values	numeric values
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	BlockSize, BS

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.

CHECK_POINT_RESTART={ YES | NO | nn }

Default	YES (default is 5 minutes)
Allowable Values	YES, NO, nn
Minimum	1 minute
Maximum	90 minutes
Alternate Specification	CheckPointRestart, CP

*Note: This parameter requires you to **Submit** your transfer.*

When enabled, this parameter 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 sender's checkpoint and its own checkpoint in the persistent queue. In the event of a failure, the initiator and the responder negotiate the saved checkpoint information and restart from the last known good checkpoint. Checkpoint is specified in units of time.

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

COMMAND=

Default	Not Applicable
Allowable Values	Command to be executed
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification*	RC, RemoteCommand, E, EXEC, RE, REXXEXEC, SJ, SUBJCL, CJ, CALLJCL, CPG, CALLPGM

This parameter is used with the File to Remote Command feature.

*The Alternate Specifications for this parameter depend on the remote system that the command will be executed on. Below is a list of 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	NO
Allowable Values	YES, RLE, LZ, Z1 – Z9, NO
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	Compression, CM

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

LZ 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.

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

Z1 through Z9 refer 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 and produces the best quality of compression. Level 7 (ZLIB7) typically offers the best compromise of compression and speed.

CR_LF={ YES | NO }

Default	NO
Allowable Values	YES, NO
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	CrLf, CR

CR_LF indicates that Carriage Return/Line Feed translation should be performed during the transfer. This parameter has no effect when sent with DATA_TYPE=B(Binary).

DATA_ TYPE = { A | B | E }

Default	E
Allowable Values	A, B, E
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	DataType, DT

Data type specifies what format the data should be stored in on the remote system. Binary will indicate that there should be no conversion done.

DATA_CLASS

Default	Not Applicable
Allowable Values	1–8 character value
Minimum	1 character
Maximum	8 characters
Alternate Specification	DataClass, DC

This represents the z/OS Data Class as defined to the Data Facility /System Managed Storage. In addition, it is used to indirectly select file attributes such as Record Format and Logical Record Length. This is a 1–8 character value, which contains either numeric, alphabetic, or national characters (in the United States these are $,#, or @). The first character must be alphabetic or national.

DESTINATION

Default	Not Applicable
Allowable Values	LU name, IP Name or IP Address
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	Destination, DS, LuName, LU, Host, HO

This is the address of the remote system.

ENCRYPTION = {DES | 3DES | BF | BFL | RJ | NONE}

Default	OFF
Allowable Values	DES, 3DES, BF, BFL, RJ, NONE
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	en

This parameter determines the level of encryption that will be used by default in your system. Valid responses are:

- 128 None No encryption is the default for this Fusion system
- 64 DES DES encryption will be used
- 32 3DES Triple DES encryption will be used
- 16 BLOWFISH Blowfish encryption will be used
- 8 BLOWFISHLONG Blowfish Long encryption will be used
- 4 Rijndael Rijndael encryption will be used

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

EXPIRATION_DATE

Default	None
Allowable Values	MM/DD/YYYY, HH:MM:SS, SUN, MON, TUES, WED, THURS, FRI, SAT
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	ExpirationDate, ED

Specifies the exact date and time when a transfer should no longer be attempted. However, if this transfer was scheduled, that will take precedence over expiration. If Expiration and Retention are used, then whichever value occurs first will take precedence.

FILE_AVAIL={ I | D }

Default	I
Allowable Values	I, D
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	FileAvailability, FA

- I Immediate This indicates that the file is available to be transferred immediately. This is the default.
- D Deferred Specifies that 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	NONE
Allowable Values	1–31 character value
Minimum	1 character
Maximum	31 characters
Alternate Specification	FileTransferServer, FS

This parameter is used in conjunction with the Submit parameter in order to submit a transfer to another MFT Platform Server. The MFT Platform Server uses the ServerName to obtain an RPC Binding Handle to the MFT Platform Server that will be processing the file transfer and then submits the transfer to the server's queue.

When the server name specified in this parameter is invalid or there is not an active MFT Platform Server running on the machine, an error is returned.

If a MFT Platform Server has been selected and Submit was not specified MFT Platform Server will accept the request for transfer however it will only perform a two stage client to host transfer.

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

FTMSCMD /send /file /FS:DOMAIN /SERVER

LOCAL_CTFILE

Default	Not Applicable
Allowable Values	1-16 characters
Minimum	1 character
Maximum	16 characters
Alternate Specification	InitiatorCTFile, LCT

This parameter is used to convert data between ASCII and EBCDIC. This parameter will contain the name of the file, which will be used to translate on the local side. This would not be necessary if you are communicating from PC to PC.

LOCAL_DOMAIN

Default	Not Applicable
Allowable Values	1-15 characters
Minimum	1 character
Maximum	15 characters
Alternate Specification	LocalDomain, LD

Provides information about the user who initiated the transfer.

LOCAL_PASSWORD

Default	X:
Allowable Values	1-15 characters
Minimum	1 character
Maximum	15 characters
Alternate Specification	LocalPassword, LW

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

LOCAL_USER_ID

Default	None
Allowable Values	20 characters
Minimum	1 character
Maximum	20 characters
Alternate Specification	LocalUserId, LI

Provides information about the user who initiated the transfer and is not case sensitive.

LIST

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

Assigns the distribution list to use for the transfer request

MGMT_CLASS

Default	None
Allowable Values	1–8 character value
Minimum	1 character
Maximum	8 characters
Alternate Specification	MgmtClass, MC

This represents the z/OS Management Class as defined to the Data Facility /System Managed Storage. This is a 1–8 character value, which contains either numeric, alphabetic, or national characters (in the United States these are $, #, or @). The first character must be an alphabetic or national character.

MODE_NAME= {#BATCH}

Default	#BATCH
Allowable Values	1–8 characters
Minimum	1 character
Maximum	8 characters
Alternate Specification	ModeName, MN

The mode name is used to represent a set of characteristics to be used in an APPC LU-LU session. This must match a mode name that was defined in the Microsoft SNA Server definition.

NOTIFY_LOCAL_USER= userid

Default	NONE
Allowable Values	1–80 character name
Minimum	1 character name
Maximum	80 character name
Alternate Specification	NotifyLocalUser, NLU

This is the 1–80 character name of the local user to NOTIFY when this file transfer has completed, either successfully or unsuccessfully. For this name, it is recommended that you use either your own userid or one of your Operations Support team members.

NOTIFY_LOCAL_USER_TYPE={MAIL | WINDOWS } | : {SUCCESS | FAILURE}

Default	None
Allowable Values	MAIL, WINDOWS
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	NotifyLocalUserType, NLT

This is the type of userid to NOTIFY after a file transfer terminates. This is used in conjunction with the NOTIFY_LOCAL_USER= parameter.

The allowable values are:

- MAIL — To give e-mail notification for both successful and failed transfers.
- MAIL:SUCCESS — To give e-mail notification for only successful transfers.
- MAIL:FAILURE — To give e-mail notification for only failed transfers.
- WINDOWS — To give Windows pop-up notification for both successful and failed transfers.
- WINDOWS:SUCCESS — To give Windows pop-up notification for only successful transfers.
- WINDOWS:FAILURE — To give Windows pop-up notification for only failed transfers.

NOTIFY_USER= userid

Default	NONE
Allowable Values	1–20 character name
Minimum	1 character name
Maximum	20 character name
Alternate Specification	NotifyUser, NU, NotifyRemoteUser, NRU

This is the 1–20 character name of the remote user to NOTIFY when this file transfer has completed, either successfully or unsuccessfully. For this name, it is recommended that you use either your own userid or one of your Operations Support team members.

NOTIFY_USER_TYPE={ NONE | T | R | W | M } |: { S | F }

Default	None
Allowable Values	NONE, T, R, W, M
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	NotifyUserType, NT, NotifyRemoteUserType, NRT

This is the type of userid to NOTIFY after a file transfer terminates. This is used in conjunction with the NOTIFY_USER = parameter. When TSO or ROSCOE is selected, both successful and failed transfers will get a notification. If WINDOWS or MAIL is selected, then the user can decide if he would like notification for successful or failed transfers only.

The allowable values are:

- NONE — No notification.
- T TSO — TSO notification for both successful and failed transfers.
- R ROSCOE — ROSCOE notification for both successful and failed transfers.
- W WINDOWS — To give Windows pop-up notification for both successful and failed transfers.

- W:S WINDOWS:SUCCESS To give Windows pop-up notification for only successful transfers.
- W:F WINDOWS:FAILURE To give Windows pop-up notification for only failed transfers.
- M MAIL To give e-mail notification for both successful and failed transfers.
- M:S MAIL:SUCCESS To give e-mail notification for only successful transfers.
- M:F MAIL:FAILURE To give e-mail notification for only failed transfers.

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

Default	None
Allowable Values	S, H, A, R, C, Z, E, T
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	PermittedActions, PA

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

PORT

Default	46464
Allowable Values	1–65535

This parameter names the port number for a TCP/IP transfer. The default IP Port Number for MFT Platform Server is 46464, but you can change it to any number between 1 and 65535, inclusive. However, some lower port numbers may be reserved for standard applications at your installation. For an SSL transfer, the SECURE parameter must also be used.

PRIORITY= { 3 | n }

Default	3
Allowable Values	1–6
Minimum	1
Maximum	6
Alternate Specification	Priority, PR

This is the priority that will be applied when the thread for the file transfer is created. This does not indicate the priority that the job will have in the MFT Platform Server work queue. *n* is a decimal

number from 1–6 which indicates the priority of the file transfer. The priorities are 1 to 6, high to low respectively. It is recommended that you let this default to 3.

PROCESS_NAME

Default	CyberFus
Allowable Values	1–8 characters
Minimum	1 character
Maximum	8 characters
Alternate Specification	ProcessName, PN

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

REMOTE_CTFILE

Default	Not Applicable
Allowable Values	1-16 characters
Minimum	1 character
Maximum	16 characters
Alternate Specification	ResponderCTFile, RCT

This is used to convert data between ASCII and EBCDIC. This parameter will contain the name of the file, which will be used to translate on the remote side. This would not be necessary if you are communicating from PC to PC.

RECORD_FORMAT={ F | FB | V | VB | U }

Default	FB
Allowable Values	F, FB, V, VB, U
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	RecordFormat, RF

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

- F Fixed — Each string contains exactly the number of characters defined by the string length 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 string length 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.
- U Undefined — Blocks are of an undefined size. There are no logical records. The logical record length will appear as zero. This record format is usually only used in load libraries. Block size must be used if you are specifying Undefined.

RECORD_LENGTH={ nnnnn | 0 }

Default	1 (F or FB) 4 (V or VB)
Allowable Values	1*–32760
Minimum	1 (F or FB) 4 (V or VB)
Maximum	32760
Alternate Specification	RecordLength, RL

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

*****If RecordFormat=F or FB, then the allowable values are 1–32760. If RecordFormat=V or VB, then the allowable values are 4–32760.

REMOTE_DOMAIN

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

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

REMOTE_PASSWORD

Default	None
Allowable Values	Password up to 20 characters
Minimum	1 character
Maximum	20 characters
Alternate Specification	RemotePassword, RW

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

REMOTE_PRINTER_NAME

Default	Not Applicable
Allowable Values	Any combination of up to 255 characters
Minimum	1 character
Maximum	255 characters
Alternate Specification	RemotePrinterName, RP

This is the name of the remote printer to which the job will be sent when using File to Job.

REMOTE_USER_ID

Default	None
Allowable Values	1–20 characters
Minimum	1 character
Maximum	20 characters
Alternate Specification	RemoteUserId, RI

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

REMOVE_TRAILING_SPACES

Default	N
Allowable Values	Y, N

Minimum	N/A
Maximum	N/A
Alternate Specification	RemoveTrailingSpaces, RTS

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

RETENTION_PERIOD

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

Specifies the number of days that should pass from the transfer's start to the point it should no longer be attempted. If Expiration and Retention are used, then whichever value occurs first will take precedence.

RSAccelerator

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

Setting this parameter to Y will force a transfer to be conducted through a Windows MFT Platform RocketStream Accelerator server using the RocketStream Accelerator technology which allows you to greatly improve data transfer speeds over IP networks with high latency. *Note: You must be licensed for RSA to use this technology.*

RSCompression

Default	N
Allowable Values	Y|Best, Default, Fast
Minimum	N/A
Maximum	N/A
Alternate Specification	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 compression to Default your file will receive the greatest compression and may take slightly longer to transfer then if you used Fast which will result in your file being less compressed but sent out faster.

RSEncryption

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

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

RSHost

Default	None
Allowable Values	Host, N
Minimum	N/A
Maximum	N/A
Alternate Specification	RSH

This is the IP or Hostname of the Windows MFT Platform RocketStream Accelerator server. By defining a host on the command line or in a transfer template you will be overriding the RSHost value configured in the config.txt if it is defined. If the value is N and you have RSAccelerator set to Yes then the value configured for RSHost in the config.txt will be used.

RSMaxSpeed

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

When conducting file transfers through an RSAccelerator (RSA) you can set the Max Speed in Kilobytes per second to be used by the RSA server when you set this parameter in your command line or transfer template.

RSPort

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

This is the port number the Windows MFT Platform RocketStream Accelerator server is listening on for transfers using the RocketStream Accelerator technology. By defining a port number on the command line or in a transfer template you will be overriding the RSPort value configured in the config.txt. Default is 9099. If the value is N and you have RSAccelerator set to Yes then the value configured for RSPort in the config.txt will be used.

RSProtocol

Default	None
Allowable Values	TCP, UDP, PDP
Minimum	N/A
Maximum	N/A
Alternate Specification	RSP

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

SCHEDULE_AT

Default	None
Allowable Values	MM/DD/YYYY, HH:MM:SS
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	ScheduleAt,SAT

*Note: This parameter requires you to **Submit** your transfer.*

Specifies the date and time when a transfer will be executed.

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

Default	None
Allowable Values	N, I, T:x, U
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	ScheduleRepeat,SRE

Specifies the rate at which the schedule should be repeated.

The allowable values are:

- N NO Do not repeat the transfer.
- I INFINITE Repeat the transfer forever.
- T:x TIMES Repeat the transfer x amount of times.
- U UNTIL Repeat the transfer until the specified date and time. Format – MM/DD/YYYY, HH:MM:SS

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

Default	None
Allowable Values	D7, WK, 2WK, MON, 2MON, QTR, 2QTR, YR, 2YR, E:n:u
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	ScheduleInterval, SRI

Specifies the interval in which the transfer should be repeated. This parameter should be used only if you are scheduling the transfer.

The allowable values are:

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

SECURE

Default	0
Allowable Values	0, 1

Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	SecureCommunication, SSL

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

SECURITY_ATTRIB_FILENAME

Default	None
Allowable Values	1–8 character value
Minimum	1 character
Maximum	8 characters
Alternate Specification	SecurityAttribFileName, SA

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

StopOnFailure

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

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

STOR_CLASS

Default	None
Allowable Values	1–8 character value
Minimum	1 character
Maximum	8 characters
Alternate Specification	StoreClass, SC

This represents the z/OS Storage Class as defined to the Data Facility /System Managed Storage, which is used to indicate the host file's media type and the installation's backup, restore, and archive policies. This 1–8 character 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	N
Allowable Values	Y, N
Minimum	N/A
Maximum	N/A

Allows the user 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. This is used when running directory transfer requests and transfers using a distribution list.

TRACE_LEVEL

Default	1
Allowable Values	1–9
Minimum	1
Maximum	9
Alternate Specification	TraceLevel, TL

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

TRY_COUNT= { nn | 1 }

Default	1
Allowable Values	1–10 or unlimited (or 0)
Minimum	1
Maximum	unlimited
Alternate Specification	TryCount, TC

Where *nn* is a decimal number from 0–10 that indicates the *maximum* number of times that this file transfer can be attempted before it is purged from the MFT Platform Server work queue.

RETRY=0 indicates no limit. It is best not to specify this parameter; instead, use the default.

UNIT = SYSDA

Default	SYSDA
Allowable Values	1–8 character name
Minimum	1 character
Maximum	8 characters
Alternate Specification	Unit, UN

This is the 1–8 character name of the Unit where the Host data set is to be allocated.

USER_DATA= User Data/Description

Default	none
Allowable Values	any string of up to 25 characters
Minimum	0 or none
Maximum	25 characters
Alternate Specification	UserData, UD

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

VOL_SER

Default	none
Allowable Values	1–6 character name
Minimum	none
Maximum	6 characters
Alternate Specification	VolumeSerialNumber, VS

Indicate the default volume serial to use for new datasets created by the MFT Platform Server Responder. If you leave this parameter blank, it will use the VOLSER that was specified in the

GLOBAL parameters on the z/OS system. This parameter is ignored when sent to MFT Platform Server for Windows.

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

Default	CR
Allowable Values	C, R, A, CR, CA, CN
Minimum	Not Applicable
Maximum	Not Applicable
Alternate Specification	WriteMode, WM

Indicates the effect on the remote file.

- C Create — Create the remote file. Abort the transfer if it already exists.
- R Replace — Replace the remote file only. If it does not yet exist, then abort the transfer.
- A Append — Append to the remote file.
- CR Create Replace — Create the remote file or Replace it if it already exists.
- CA Create Append — Create the remote file or Append it if it already exists.
- CN Create Replace New — Create the remote file or replace it with new attributes. When specified for transfers to Windows, CN indicates that the system should create directory paths as needed.

4.7 Use of Errorlevel with FTMSCMD

FTMSCMD passes back return codes in order to assist the programmer when writing batch jobs. The following example batch job will execute a transfer. A message will be displayed to the screen indicating the success or failure of the transfer.

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

if errorlevel 1   goto ERROR
if errorlevel 0   goto SUCCESS

:ERROR
      echo transfer failed
      goto END

:SUCCESS
      echo transfer successful
      goto END

:END
      echo batch program complete
```

4.7.1 Overview of our Sample Batch Program

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

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

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

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

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

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

4

5. Extended Features

The MFT Platform Server for Windows provides many features including:

- Access Control
- User Id Alias
- Directory Named Initiation
- Custom Code Page Conversion
- Directory Transfers
- Wildcard Support
- Secure Sockets Layer (SSL) Support
- Remote Nodes and User Profiles

5.1 Access Control

This section explains the concept of MFT Platform Server's Access Control. Users would like to be able to change the default location for a file based on the **USERID**, **NODE** and/or **IPADDR**. Users can now send a file to the Windows platform and it will automatically go to a pre-defined directory based on user-defined criteria. The Access Control configuration file, called **AccessControl.cfg** by default, must be selected under the Responder tab under Server Properties. This feature only is used for the MFT Platform Server for Windows Responder.

5.1.1 Sample of AccessControl.cfg File

Below is a sample of the Access Control configuration file, called AccessControl.cfg by default. MFT Platform Server does not look for the best match; it looks for the FIRST match. Therefore, it is suggested that you 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.

5.1.2 Parameters

A sample of the Access Control file, **AccessControl.cfg**, is located in the **MFT Platform Server** directory. Each parameter is defined below.

USERID	Defines the local userid. Either this or **NODE/IPADDR** must be specified. Both **USERID** and **NODE/IPADDR** can be specified. A value of **DEFAULT** indicates that this is the default value for a system.
NODE	Defines the node definition. Either the **NODE/IPADDR** or **USERID** must be specified. Both **USERID** and **NODE/IPADDR** can be specified. A value of **DEFAULT** indicates that this is the default value for a system. This parameter is mutually exclusive with the **IPADDR** parameter. When defining nodes in this file, make certain that you use the proper case as these files are case-sensitive.
IPADDR	Defines the IP Address in dotted decimal notation. Either the **NODE/IPADDR** or **USERID** must be specified. Both **USERID** and **NODE/IPADDR** can be specified. This parameter is mutually exclusive with the **NODE** parameter.
DESCRIPTION	Allows the user to enter a 32 byte description or comment.
SEND_DIR	Defines the default directory for files to be sent to another system. If this parameter is not defined, then there is no default value for files sent.
RECEIVE_DIR	Defines the default directory for files to be received from another system. If this parameter is not defined, then there is no default value for files sent.
COMMAND_DIR	Defines the default directory for commands executed on this system. If this parameter is not defined, then there is no default value for files sent.
SUBMIT_DIR	This parameter is valid only for MFT Platform Server on z/OS. It defines the default directory for files to be submitted into the z/OS internal reader. MFT Platform Server on z/OS also allows this parameter to be specified as

	SUBMIT_HLQ. This parameter is required if SUBMIT_OPTION is set to ROOT or FORCE.
SEND_OPTION	Defines the options for Sending files. There are six valid values: **ROOT -** If a directory is specified, the directory will be appended to the directory defined by the SEND_DIR parameter. **FORCE -** If a directory is specified, the directory will be 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. **ALLOW -** If a directory is specified, the directory will be used. If a directory is not defined, it will be changed to the directory defined by the SEND_DIR parameter. **REJECT -** If a directory is specified on a Send, the file transfer will terminate with errors. Otherwise, data will be processed from the SEND_DIR directory. **NEVER -** The **NODE** or **USERID** is not allowed to Send a file. **USE -** The directory name specified in the file transfer request will be used. If no directory name is defined in the file transfer request, the Windows default directory will be used.
RECEIVE_OPTION	Defines the options for Receiving files. There are six valid values: **ROOT -** If a directory is specified, the directory will be appended to the directory defined by the RECEIVE_DIR parameter. **FORCE -** If a directory is specified, the directory will be 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. **ALLOW -** If a directory is specified, the directory will be used. If a directory is not defined, it will be changed to the directory defined by the RECEIVE_DIR parameter. *Note: By setting ALLOW will allow files to be written to directories other than that which is defined in the RECEIVE_DIR..If a relative path (directory without a slash in the beginning. i.e. tmpdir\filename.txt) is used for a remote file name in the transaction coming in, MFTPS will place files in the current directory where Platform Server is executing if the user has access rights. This is the MFTPS System directory.* **REJECT -** If a directory is specified on a Send, the file transfer will terminate with errors. Otherwise, data will be processed from the RECEIVE_DIR directory. **NEVER -** The **NODE** or **USERID** is not allowed to Receive a file. **USE -** The directory name specified in the file transfer request will be used. If no directory name is defined in the file transfer request, the Windows default directory will be used.
COMMAND_OPTION	Defines the options for executing Commands. There are three valid values: **ROOT -** If a directory is specified, the directory will be appended to the directory defined by the COMMAND_DIR parameter. **NEVER -** The **NODE** or **USERID** is not allowed to execute Commands. **USE -** The directory name specified in the file transfer request will be used. If no directory name is defined in the file transfer request, the Windows default directory will be used.
SUBMIT_OPTION	Defines the options for Submitting jobs. There are two valid values: **ALLOW –** The user is allowed to submit jobs. **NEVER -** The **NODE** or **USERID** is not allowed to Receive a file.

5.1.2.1 Directory Name Used in Request

If the directory name is defined in the RECEIVE_DIRECTORY and the **FORCE** parameter is defined, then 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 1:

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

If the directory name is defined in the RECEIVE_DIRECTORY and the **ROOT** parameter is defined, then 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 will be: c:\sales\2005\accounting\tax.xls

5.1.2.2 Continuation and Comments

Parameters can be entered on a single line or on multiple lines. Parameters are delimited by a comma. If a space follows the comma, the parameter is continued on the next line. If there is a special character in the parameter, it should be enclosed in double quotes.
Example:

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

Is the same as :

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

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

5.1.2.3 Default Entries

The user can specify default entries for the USERID and NODE parameters by using the value DEFAULT. This will provide 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
```

5.1.2.4 Parameter Validation

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

On z/OS MFT Platform Server will validate all CFACCESS parameters at startup and whenever the CFACCESSREFRESH command is executed. Only valid entries will be saved into memory. When file transfer requests are received, the entries in memory will be checked.

5.2 CFAlias

Some architectures do not want users knowing 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 filename, 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 filenames. The CFAlias configuration file, called CfAlias.cfg by default, must be selected under the Responder tab under Server Properties. This feature only is used for the MFT Platform Server for Windows Responder.

5.2.1 CFAlias Parameters

The following parameter values will be supported. The syntax is similar to the AccessControl syntax. Parameters are 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, make certain that you use the proper case as these files are case-sensitive.
IPADDR	Defines the name of the IPAddress that initiated the transfer.
TYPE	Valid values are **SEND**, **RECEIVE** or **BOTH**. This parameter is relative to the Responder. Therefore if the Initiator issues a **SEND** request, the CFAlias feature will consider this a **RECEIVE** request because it is operating as the Responder.
FILE	Defines the actual fully qualified file name to be used.
ALIAS	Defines the name of the file that is sent by the initiator.
ALLOW	Valid values are **YES** or **NO**. When specified as **YES**, a match indicates that the user is allowed to define the actual file name if a match is not made on an alias grouping. When defined as **NO**, the request will fail if no match is made with an alias grouping. **NODE/IPADDR** and/or **USERID** must be defined. When **ALLOW** is not defined, then **FILE** and **ALIAS** must be defined. When **ALLOW** is defined, the **FILE** and **ALIAS** parameters are not allowed. If a sender's parameters do not match any entry in the alias config file, then the transfer will be rejected.

5.2.2 Substitutable Parameters

The MFT Platform Server administrator can define substitutable parameters in the FILE parameter of the CFAlias file. Substitutable parameters are defined by a % followed by the parameter name. The following Substitutable parameters are allowed:

%JDATE	Julian Date (YYDDD)
%JDATEC	Julian Date (CCYYDDD)
%GDATE	Gregorian Date (YYMMDD)
%GDATEC	Gregorian Date (CCYYMMDD)
%TIMET	Time (HHMMSST)
%TIME	Time (HHMMSS)
%NODE	Node Name (if no node 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

5.2.3 Example of How CFAlias Could Be Used

Say the architecture is a UNIX file server with people using MFT Platform Server on a Windows machine. Say that the user JohnDoe needs to send a daily report (**report.doc**) to the server every day, but the administrator wants to keep a record of each previous day's report around. Also, the administrator does not want JohnDoe to know/bother with the architecture on the server. Thus, on the client side, JohnDoe would specify his remote file name as **report.doc**, because he knows no other name for it, and the CfAlias.cfg file would contain the following groupings:

```
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 will send his daily report every day exactly the same way. Each time he sends his report to the server, it will be put in the **c:\JohnDoe\DailyReports\report.%GDATE.doc** file, so each day it will have a different filename based on the current date. For example, if the date is July 18th, it would be 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**.

5.2.4 Sample of CfAlias.cfg File

```
************************************************************************
*   This file contains the CFAlias Configuration parameters.
*   This file will be searched for parameters that match the
*   userid and/or node.
*
*   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, RECV 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 IP address
*   must have both file and alias
*   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 user id and/or node is used.
*
************************************************************************
USERID=Admin,
NODE=DEFAULT,
IPADDR=111.222.34.56,
TYPE=SEND,
FILE=c:\johnd\files\remotefile,
ALIAS=monthlysales

IPADDR=22.33.44.55,
TYPE=SEND,
ALLOW=no
```


5.3 CFINQ

This section describes the logging function and how to query past transactions. The log file stores the basic parameters of a transfer, but little information about how the transaction went 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.

5.3.1 Log Files

MFT Platform Server for Windows has comprehensive logging to provide information about transfers that were initiated as well as transfers that were received on the Windows platform. There is one common log to record this information from both the initiator and the responder. A new log file (Log.txt.yyyymmdd) will be created each day with the date appended to the end of the filename entered in the configuration file. This file will be accessed when inquiring on transactions 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. A sample for the daily log is shown below:

```
VersionNumber=6.1 Build 985,Priority=N/A,LocalTranNumber=I729500019, RemoteTranNu
mber=R729500020,TransferStartTime=095158, TransferStartDate=20050729,TransferEndTi
me=095201,TransferEndDate=20050729,TransferDirection=Send,TransferWork=File,TransferCommand=N/
A,TransferProcessName=N/A,TransferScheduleDate=N/A,TransferScheduleTime=N/A,TransferExpirationD
ate=N/A,TransferExpirationTime=N/A,CompressionType=None,CompressedBytes=0,ConvertCRLF=no,EB
CDICTranslate=no,SSL=no,SSLPortNumber=N/A,EncryptionType=Blowfish_448,RecordFormat=FixedBlo
ck,FileCreateOptions=CreateReplaceNew,FileAttributes=N/A,UNIXFilePermissions=666,AllocationType=N
/A,AllocationDirectory=N/A,AllocationPrimary=N/A,AllocationSecondary=N/A,Volume=N/A,Unit=N/A,N
odeClass=N/A,StorClass=N/A,MgtClass=N/A,DataClass=N/A,BlockSize=0,RecordLength=80,UserData=N/
A,LogonDomain=N/A,LocalFileName=c:\localfile.txt,LocalUserid=abc,RemoteFileName=/home/remotefile,
RemoteUserid=xyz,RemoteNodeName=111.22.33.44,RemoteNodeType=N/A,RemotePortNumber=46464,Tr
yCount=1,TryMaxCount=1,GoingToRetry=0,ByteCount=27,RecordCount=N/A,MemberCount=N/A,CheckP
ointCount=N/A,CheckPointRestart=no,CheckPointInterval=0,StatusMsg=File
TransferComplete,CrlMsg=N/A,StatusDiagCode=0,StatusSev erity =
00,StatusReturnCode=N/A,TransferStatus=Success,LocalCTFile=N/A,RemoteCTFile= N/A
```

5.3.2 CFINQ Program

The MFT Platform Server Inquiry program, CFINQ, provides two ways of showing the audit information to a user in a more convenient and user friendly 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. Below is a list of the parameters that may be utilized for the execution of the CFINQ program. An equal sign or a colon may be used to separate the parameter from the value.

In order to obtain all transactions in the specified query, a user must be either the Administrator, or be in the **cfbrowse** or **cfadmin** group. If a user is not in either of the groups or is not the Administrator user then they will be able to view only their own transactions.

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

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

5.3.2.1 CFINQ Parameters

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

Parameter Name	Alternate Parameter Names	Short Description
DAYS		Number of days to search
DESCRIPTION	DESCR	MFT Platform Server UserData
ENDDATE	EDATE	End date
ENDTIME	ETIME	End time
EXCEPTIONS	EXC	Status of the transaction
INITRESPFLAG	IRF	Initiator or Responder records
LOCALFILE	LF	Log file directory
LOCALUSER	LUSER	Local User Id
LOCTRANSNUM	LTRN	Local Transaction number
LOGDIR	LOGD	MFT Platform Server log files directory
MAXXFER	MAX	Number of transactions to list
PROCESS	PRO	MFT Platform Server Process name
REMHOST	RHOST	Remote System name
REMTRANSNUM	RTRN	Remote Transaction number
STARTDATE	SDATE	Start date
STARTTIME	STIME	Start time
TEMPERROR	TMPERR	Temporary Error Records

DAYS

If SDATE and EDATE are both defined, this field is ignored. DAYS must not exceed 1826 (5 years). If SDATE is not defined, the Start Date = Current Date - # of Days. If SDATE is not defined and EDATE is defined CFINQ program will start searching (EDATE - # of DAYS) and ends at the EDATE date.

DESCRIPTION

Defines the MFT Platform Server UserData. By providing the DESCRIPTION parameter the user shall expect the CFINQ program to search the MFT Platform Server log files and present the detailed information for any transfers matching that description. A message will be displayed on the screen if there are no transactions for the DESCRIPTION specified.

ENDDATE

The ENDDATE defines the end date in format: *yyyymmdd.*
EDATE=TOD or EDATE=TODAY means today's date.
EDATE=YES or EDATE=YESTERDAY means yesterday.
If EDATE is not defined the default is TODAY.

ENDTIME

Defines the End Time in the 24 hour format of hhmmss. The default is 240000. If STIME is not defined the CFINQ program will search for the MFT Platform Server transaction only within 000000 - ETIME period.

EXCEPTIONS

Defines the type of transfers to select.
U = Unsuccessful
S = Successful
Default = Successful and Unsuccessful

INITRESPFLAG
Defines Initiator or Responder records to select.
B = Both
I = Initiator
R = Responder
Default = Both

LOCALFILE
Defines the MFT Platform Server Local File Name.

LOCALUSER
Defines the Local User Name (userid). If you specify a user name other than your own, you must have Security Authorization.

LOCTRANSNUM
The LOCTRANSNUM parameter defines the unique local transaction number of the MFT Platform Server transfer. By providing the LOCTRANSNUM parameter the user shall expect the CFINQ program to search the MFT Platform Server log files and will present the detailed information for that transaction number. A message will be displayed on the screen if there are no transactions for the LOCTRANSNUM specified.

LOGDIR
Defines the MFT Platform Server log files directory.

MAXXFER
Defines the maximum number of requests that will be returned. The default is 500. The valid values are 1 to 100,000.

PROCESS
Defines the MFT Platform Server Process name.

REMHOST
Defines the MFT Platform Server remote system name. This can be a NODE name, SNA LUNAME, IPName or IPAddress in dotted decimal notation. Generic selection cannot be used for IP Addresses.

REMTRANSNUM
The REMTRANSNUM parameter defines the unique remote transaction number of the MFT Platform transfer. By providing the REMTRANSNUM parameter the user shall expect the CFINQ program to search the MFT Platform Server log files and will present the detailed information for that remote transaction number. A message will be displayed on the screen if there are no transactions for the REMTRANSNUM specified.

STARTDATE
The STARTDATE defines the start date of the search in format: *yyyymmdd*.
SDATE=TOD or SDATE=TODAY will use today's date.
SDATE=YES or SDATE=YESTERDAY means yesterday.
If SDATE is not defined, the default is TODAY.

STARTTIME
Defines the Start Time in the 24 hour format of hhmmss. The default is 000000. If ETIME is not defined the CFINQ program will search for the MFT Platform transaction only within STIME – 24 hour period.

TEMPERROR
Defines if temporary error records are to be selected.
Y = Yes
N = No

Default = Yes

5.3.2.2 CFINQ Example

Below is an example how a user would use CFINQ on the command line. In this example, the start date is July 9, 2005 looking 20 days forward with a start time on 9:01am and an end time of 3pm. The local user is abc. We want to list successful transfers only, with a maximum of 1000 records listed. Below is the command that would be entered:

```
cfinq sdate:20050709 days:20 stime:090100 etime:150000 luser:abc
lf:"c:\cfserver\log.txt" EXC:S max:1000
```

An equal sign or a colon may be used to separate the parameter from the value.

Output:

```
*************************************************************************
        YOU HAVE ENTERED THE FOLLOWING VALUES FOR YOUR INQUIRY:

        LOCTRANSNUM.............[]
        REMTRANSNUM.............[]
        LOGDIR..................[]
        STARTDATE...............[20050709]
        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 menu options.

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

To view the next 20 transactions for the sample above, select **n** and you will see the following:

```
INDEX TRANSACTION STATUS  IPADDRESS             LOCALFILENAMEDIRECTORY
*************************************************************************
   1  I729500007  Success 111.22.33.44:46464   c:\a.txt
   2  I729500009  Success 111.22.33.44:46464   c:\a.txt
   3  I729500011  Success 111.22.33.44:46464   c:\a.txt
   4  I729500012  Success 111.22.33.44:46464   c:\a.txt
   5  I729500013  Success 111.22.33.44:46464   c:\a.txt
   6  I729500014  Success 111.22.33.44:46464   c:\a.txt
   7  I729500015  Success 111.22.33.44:46464   c:\a.txt
   8  I729500017  Success 111.22.33.44:46464   c:\a.txt
   9  R729500020  Success 111.22.33.44:46464   c:\remotefile
  10  I729500019  Success 111.22.33.44:46464   c:\cfserver\log.txt
  11  I729500021  Success 111.22.33.44:46464   c:\a.txt
  12  R729500025  Success 111.22.33.44:46464   c:\temp\CG.DAT
  13  R729500027  Success 111.22.33.44:46464   c:\temp\CLEAN.EXE
  14  R729500029  Success 111.22.33.44:46464   c:\temp\RUN.BAK
  15  R729500032  Success 111.22.33.44:46464   c:\temp\HOOK.REG
  16  R729500028  Success 111.22.33.44:46464   c:\temp\RUN.BAK
  17  R729500033  Success 111.22.33.44:46464   c:\temp\WAKE.EXE
  18  R729500002  Success 111.22.33.44:46464   c:\temp\EXPRESS.INI
  19  R729500000  Success 111.22.33.44:46464   c:\temp\HOOK.REG
  20  R729500034  Success 111.22.33.44:46464   c:\temp\ENGLISH
  21  R729500031  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) was selected to show the details of transaction I729500019:

```
*************************************************************************
RECORD:10                        INITIATOR
*************************************************************************
Version..................... 7.1.0 Build 1030
Priority.................... N/A
Local Transaction Number.... I729500019
Remote Transaction Number... R729500020
Transfer Start Time......... 095158
Transfer Start Date......... 20050729
Transfer End Time........... 095201
Transfer End Date........... 20050729
Transfer Direction.......... Send
Transfer Work............... File
Transfer Command............ N/A
Transfer Process Name....... N/A
Transfer Schedule Date...... N/A
Transfer Schedule Time...... N/A
Transfer Expiration Date.... N/A
Transfer Expriation 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 Primary.......... N/A
Allocation Secondary........ N/A
Volume...................... N/A
Unit........................ N/A
Node Class.................. 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.................. c:\cfserver\log.txt
Local User ID............... abc
Remote File................. /home/remotefile
Remote UserId............... xyz
Remote Node Name............ 111.22.33.44
Remote Node Type............ N/A
Remote Port Number.......... 46464
Try Count................... 1
Try Max Count............... 1
Byte Count.................. 27
Record Count................ N/A
Member Count................ N/A
Check Point Count........... N/A
Check Point Restart......... no
Check Point Interval........ 0
LocalCTFile................. N/A
RemoteCTFile................ N/A
Status Msg.................. File Transfer Complete
Crl Msg..................... N/A
Status Diag Code............ 0
Status Severity............. N/A
Transfer Status............. Success
```

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

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

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

NOTE:

- Navigation commands are case sensitive.
- The latest MFT Platform Server transfers information will be gathered first and, in case the total number of records will exceed 10000, the information close to the SDATE will not be included into the presented transactions list.
- The entered date must be in the following format: *YYYYMMDD*.

- The CFINQ program will not accept any negative values.
- There must not be any space between the parameter name, equal or colon sign, and parameter value.

5.4 Configured Post Processing

Configured post processing allows the user to trigger any executable command (.bat, .com, .exe, etc.) 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, will be searched upon the completion of a transfer. If the properties of the transfer match the parameters, the executable command will be triggered.

5.4.1 Configuration Parameters

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

COMMAND	Defines the file that should be executed. This is a required parameter.
TYPE	Defines the type of the file transfer request. There are three values that can be defined by the TYPE parameter: 1. SEND 2. RECEIVE 3. BOTH
SOURCE	Defines the source of the file transfer request. There are three values that can be defined by the SOURCE parameter: 1. INITIATOR 2. RESPONDER 3. BOTH
STATUS	Defines whether a transfer request was successful or unsuccessful. There are three values that can be defined by the STATUS parameter: 1. SUCCESS 2. FAILURE 3. BOTH
FILENAME or DSN	This defines the fully qualified file name. This field is compared against the local file name in the file transfer request.
PROCESS	Defines the PROCESS name associated with the transfer request.
IPADDR	Defines the IP Address of the machine that is communicating with MFT Platform Server.
NODE	Defines the NODE name of the transfer request. For initiator requests, this parameter is used when the NODE parameter is used on a request. For responder requests, MFT Platform Server scans the list of node for matches on the IP Address. These entries are then matched against the value specified in the NODE parameter. When defining nodes in this file, make certain that you use the proper case as these files are case-sensitive.

Example of CfgPostProc.cfg File

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

SUBMIT,COMMAND=cmdfile.txt,TYPE=SEND,
        STATUS=BOTH,SOURCE=INITIATOR,
```

FILENAME=infile.txt,
IPADDR=111.222.1.2

5.4.2 Argument substitution

Transfer properties can be passed to the executable command as substitutable command line arguments. Enter any of the argument names listed below after the COMMAND entry in the configuration file.
For example: COMMAND=cmdfile.txt &FILENAME &TYPE,
The filename and type of the transfer request will be substituted for &FILENAME and &TYPE and passed to the executable 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 found)
&IPADDR	IP Address(or IPADDR if not IP)
&TRN	local transaction number

5.5 Custom Code Page Conversion

This feature allows the user to convert text files between various character-set specifications. With MFT Platform Server, we provide the following four conversion tables:

Comtblg.classic	The old comtblg.dat shipped with prior versions. (Prior to v7.1)
Comtblg.cp037	Extended ASCII table that is based on IBM Code page 037
Comtblg.cp1047	Extended ASCII table that is based on IBM Code page 1047
Comtblg.dat	ASCII/EBCIDIC table used by MFTPS at run time (Default is copy of Comtblg.cp037)

Comtblg.dat, which contains information on how to convert between the ASCII and EBCDIC character sets looks like this:

```
00010203372D2E2F16050A0B0C0D0E0F
101112133C3D322618193F27221D351F
405A7F7B5B6C507D4D5D5C4E6B604B61
F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F
7CC1C2C3C4C5C6C7C8C9D1D2D3D4D5D6
D7D8D9E2E3E4E5E6E7E8E9BAE0BBB06D
798182838485868788899192939495 96
979899A2A3A4A5A6A7A8A9C04FD0A107
9F000000000000000000000000000000
000000000000000000000000000000000
41AA4AB100B26AB5BDB49A8A5FCAAFBC
908FEAFABEA0B6B39DDA9B8BB7B8B9AB
6465626663679E687471727378757677
AC69EDEEEBEFECBF80FDFEFBFCADAE59
4445424643479C485451525358555657
8C49CDCECBCFCCE170DDDEDBDC8D8EDF
002E2E2E2E2E2E2E2E2E0A2E2E0D2E2E
2E2E2E2E2E0A2E2E2E2E2E2E2E2E2E2E
2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E
2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E
20A0E2E4E0E1E3E5E7F1A22E3C282B7C
26E9EAEBE8EDEEEFECDF21242A293BAC
2D2FC2C4C0C1C3C5C7D1A62C255F3E3F
F8C9CACBC8CDCECFCC603A2340273D22
D861626364656667686 9ABBBF0FDFEB1
B06A6B6C6D6E6F707172AABAE6B8C680
B57E737475767778797AA1BFD0DDDEAE
5EA3A5B7A9A7B6BCBDBE5B5DAFA8B4D7
7B41424344454647484 9ADF4F6F2F3F5
7D4A4B4C4D4E4F505152B9FBFCF9FAFF
5CF7535455565758595AB2D4D6D2D3D5
303132333435363738 39B3DBDCD9DA2E
```

The first sixteen lines are the ASCII-EBCDIC translation table (bolded above), and the next 16 lines are the EBCDIC-ASCII translation table. Below is an explanation of how the ASCII to EBCDIC format works, a method which can be generalized to whatever text conversions are needed.

5.5.1 ASCII to EBCDIC Conversion Table Example

Each ASCII or EBCDIC character is represented by 2 hexadecimal digits. For example ASCII character E is hexadecimal 45 or X'45'. Below is the ASCII to EBCDIC translation table. The first hexadecimal digit of ASCII character E is 4, so we go down the table to the row marked 4x, the second hexadecimal digit is 5 so we 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 across to column x0 and in the box is X'D7', therefore X'50' will be 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 across to column x9 and in the box is X'5A', therefore X'E9' will be 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 across to column xD and in the box is X'29', therefore X'5D' will be 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

5.5.2 Making your own tables

For other conversions besides standard ASCII to EBCDIC, new tables can be defined. The provided table can be altered, or a completely new table can be defined. For example, say a user wants to change the EBCDIC to ASCII translation of X'DE', in the bottom half of the default table this translates to X'FA'; however, the user wants X'A3'. After the change the table becomes:

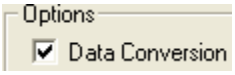
	X0	X1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	00	2E	2E	2E	2E	2E	2E	2E	2E	2E	0A	2E	2E	0D	**2E**	2E
1x	2E	2E	2E	2E	2E	0A	2E	2E	2E	2E	2E	2E	2E	2E	**2E**	2E
2x	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	**2E**	2E
3x	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	2E	**2E**	2E
4x	20	A0	E2	E4	E0	E1	E3	E5	E7	F1	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

Where Dx and xE meet we change X'FA' to X'A3'. If the user also wants the reverse conversion, we change the ASCII to EBCDIC section. So, in the top half of the table we 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

5.5.3 Additional Information

To activate the conversion tables, the Data Conversion box must be checked on the main transfer panel:



This will use the **Comtblg.dat** file for conversion. **Comtblg.dat** is located in the install directory where MFT Platform Server was installed. If the LocalCTFile and RemoteCTFile parameters are filled in under the Advanced Options tab, then they will be used instead. MFT Platform Server will not convert the file twice. The **Comtblg.dat** file is unaffected by the RemoteCTFile parameter.

In individual parameters, it is possible to specify two conversion tables; one on the local side, and one on the remote side. 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 InitiatorCTFile parameter at the command line. Similarly, the remote conversion table is specified with the RemoteCTFile parameter in GUI, and the ResponderCTFile parameter at the command line.

The **InitiatorCTFile** and **ResponderCTFile** parameters are capped at a 16 character max for purposes of shrinking the number of bytes sent per transfer. However, they support filenames relative to the current working directory on the local side. For Windows, MFT Platform Server looks in the MFT Platform Server working directory.

Nodes can also support both local and remote conversion tables associated with them. These will be used whenever that node is specified unless the parameters are overridden on the command line.

Always replace a 2-digit hexadecimal number with a 2-digit hexadecimal number. If the table is invalid translation will not be performed. Remember, the table is comprised of two sections that are 16 lines each. (each character is a 2 digit hexadecimal number). Thus the entire file should have 32 lines across and 32 lines down. If it has anything else it will not work!

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

Remember which table translates for Sends and which translates for Receives. TIBCO recommends during editing placing a few lines between the two tables to help remember which is which.

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 then rename the existing comtblg.classic file to now become the new comtblg.dat file. The conversion tables presently available do not support wide or multibyte character sets at present.

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

5.6.1 DNI GUI Interface

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. Using the Directory Initiation property sheet, you determine directory attributes and create and modify the DNI schedule. When the DNI entry dispatch is complete, you can leave the local file where it is, copy it to another directory, move it to another directory, or delete it.

Among its features and uses:

- DNI can scan network volumes shared from Novell NetWare, UNIX, AS/400, and any network drive that can be viewed using UNC (Universal Naming Convention). Note: Mapped drives are not supported.
- A DNI scan directory can be a single directory *or* a directory and all of its sub-directories.
- DNI directories are put on a flexible schedule: you can scan the directory at a time that you determine (for example, 5:00pm on Fridays, the first day of every month, or every 30 seconds).
- DNI provides store and forward capabilities where the DNI scan directory is the destination of a MFT Platform Server transfer. DNI will scan the directory and forward the file to another destination.
- DNI provides for sequential distribution, where the copy or move disposition targets another DNI directory.
- When the DNI entry dispatch is complete 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 MFT 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 Directory Initiation Property Sheet).

For optimum performance, an excessive number of DNI directories will require another MFT Platform Server for Windows. To improve performance, the dispatch time will also have to be adjusted accordingly.

5.6.1.1 Transfer Templates

A Transfer Template is a collection of any or all of the parameters needed to perform a transfer.

Each DNI entry known to MFT Platform Server is associated with a Transfer Template. The Transfer Template describes the name of the remote system (with a DNS name, remote LU Alias, 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 may associate any number of DNI entries with a Transfer Template. Do this if you want multiple DNI Directories to communicate to the same remote system.

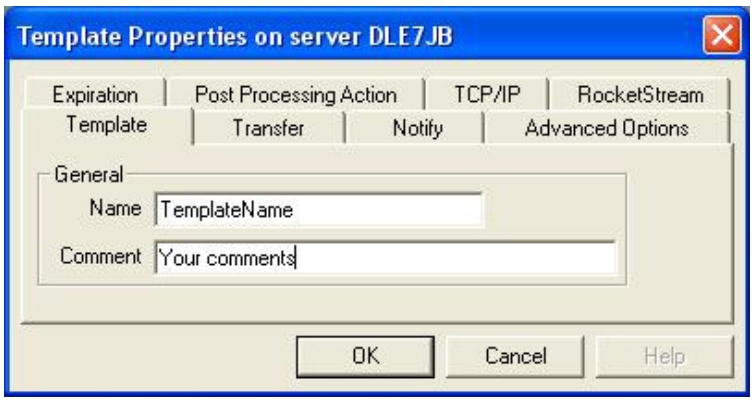
The Transfer Template also describes the name of the remote dataset or file, but if the Remote File Name is not dynamic (for example, ‘SYS1.USERDATA’), every DNI entry would overwrite the previous transfer’s data. MFT Platform Server for Windows provides for dynamic file name creation through the use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

To create a Transfer Template:

1. Click the **New Transfer Template** button to display the Administrator panel.
2. From the menu, choose **Advanced TCP Template** or **Advanced SNA Template**.
3. Complete any or all of the property pages for this transfer template as you would for any other transfer, and then complete the **Directory Initiation Properties** sheet.

There are 3 different types of templates that can be defined. They are TCP templates, SNA templates and Batch templates. The TCP and SNA templates are basically the same, except for the tab stating the protocol. The Batch template allows the user to execute a batch job instead of doing a transfer.

5.6.1.2 TCP/SNA Template Tab



- **Name**

  Name used to identify this template.

- **Comment**

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

Select the Transfer tab.

5.6.1.3 Transfer Tab

The Template Properties Transfer tab is identical to the Transfer Properties Transfer tab. Please refer to the MFT Platform Server Administrator Parameters, Transfer Tab section of the MFT Platform Server Administrator chapter for more information on this tab's parameters.

5.6.1.3.1 z/OS Options Panel

The z/OS Options Panel is identical to the Transfer Properties z/OS Options Panel. Please refer to the MFT Platform Server Administrator Parameters, z/OS Options Panel section of the MFT Platform Server Administrator chapter for more information on the Record Format, Allocation, Class, Disk, and Other tabs.

5.6.1.4 Notify Tab

The Notify tab is identical to the Transfer Properties Notify tab. Please refer to the MFT Platform Server Administrator Parameters, Notify Tab section of the MFT Platform Server Administrator chapter for more information on this tab's parameters.

5.6.1.5 Advanced Options Tab

The Advanced Options tab is identical to the Transfer Properties Advanced Options tab. Please refer to the MFT Platform Server Administrator Parameters, Advanced Options tab section of the MFT Platform Server Administrator chapter for more information on this tab's parameters.

5.6.1.6 Expiration Tab

The Expiration tab is identical to the Transfer Properties Expiration tab. Please refer to the MFT Platform Server Administrator Parameters, Expiration Tab section of the MFT Platform Server Administrator chapter for more information on this tab's parameters.

5.6.1.7 Post Processing Action Tab

The Post Processing Action tab is identical to the Transfer Properties Post Processing Action tab. Please refer to the MFT Platform Server Administrator Parameters, Post Processing Action Tab section of the MFT Platform Server Administrator chapter for more information on this tab's parameters.

5.6.1.8 TCP/IP Tab

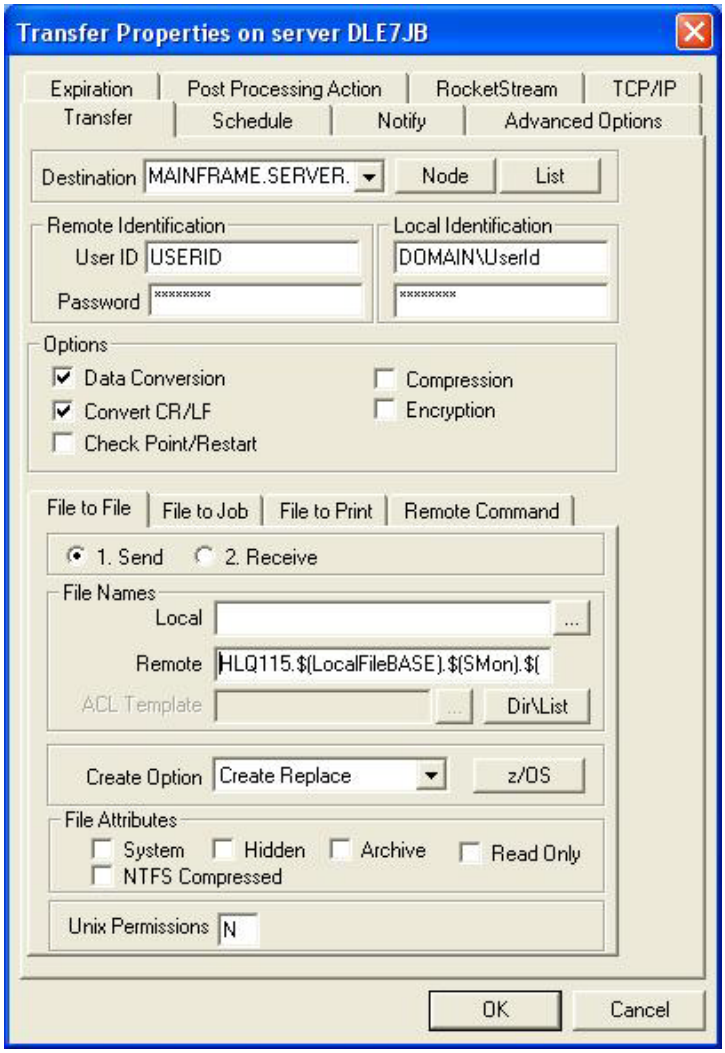
The TCP/IP and SNA tabs are identical to the Transfer Properties TCP/IP and SNA tabs. Please refer to the MFT Platform Server Administrator Parameters, TCP/IP and SNA Tabs section of the MFT Platform Server Administrator chapter for more information on these tabs' parameters.

5.6.1.9 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, we will use File Name Tokens to create a Remote file name. Leave the Local file name blank because the name and path of the file in the DNI directory will be substituted in the Local file name field automatically. The entire Remote file name is:

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

The MFT Platform Server will resolve the File Name Tokens into a file name based on the base name of the DNI file, current month and day. Assuming that the DNI file was named file001.dat and today's date is August 10, the generated file name would be:

prx0115.file001.Aug10

For a full discussion of File Name Tokens, see the *File Name Tokens* Appendix.

5.6.2 Batch Template

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

Note: The job is executed in the \WINNT\SYSTEM32 directory. Remember this when writing your batch jobs in the event that you need to change the directory in which the batch job should execute.

5.6.2.1 Batch Template Tab

- **Name**

  Name used to identify this template.

- **Comment**

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

Select the Batch Job tab.

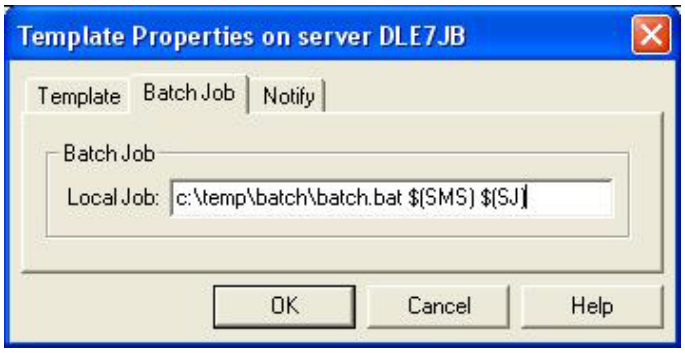
5.6.2.2 Batch Job Tab

- **Batch Job**

  Local Job
  This is the job that the user would like executed when a file is placed in the defined DNI directory.

Below is a sample of a job being executed through the Batch Job Template.

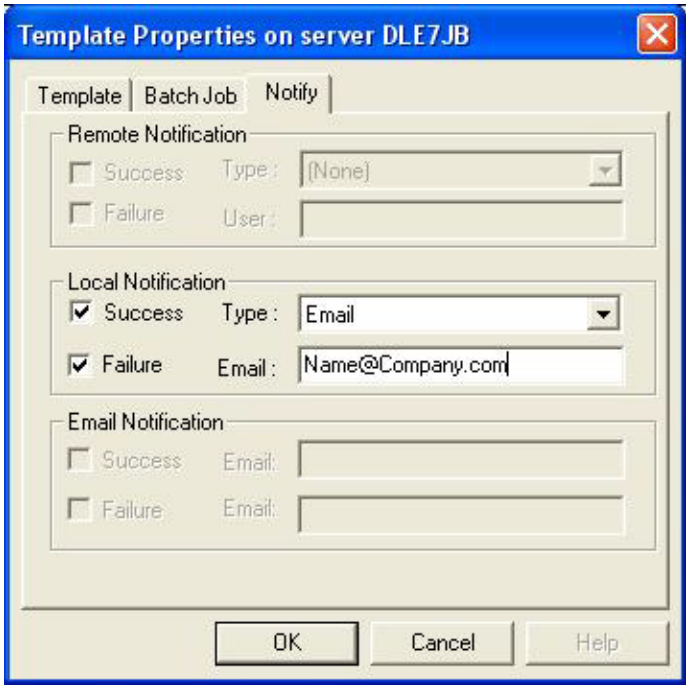
The user is required to input the whole command line for a particular job. The user can specify the file name tokens as input to the command line. Below is another example of the Batch Job Template being used to execute a job with variable parameters. As you can see here the parameters that are being 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 then specify $(LocalFilePath)\$(LocalFileName).

Select the Notify tab.

5.6.2.3 Notify Tab



- **Local Notification**

  Success

  Select this check box to send notification to the local user when the transfer is completed successfully.

Failure

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

Type

Allows the user to select the means by which the local user will be informed that the transfer has been completed. The three valid types are Windows Pop-Up, Email and NONE.

The USERID specified in the User field will not be notified should the Type field specify NONE.

User (or Email)

This is the name of the local user to notify when a transaction is completed. It lets the user know whether the transaction was successful or not.

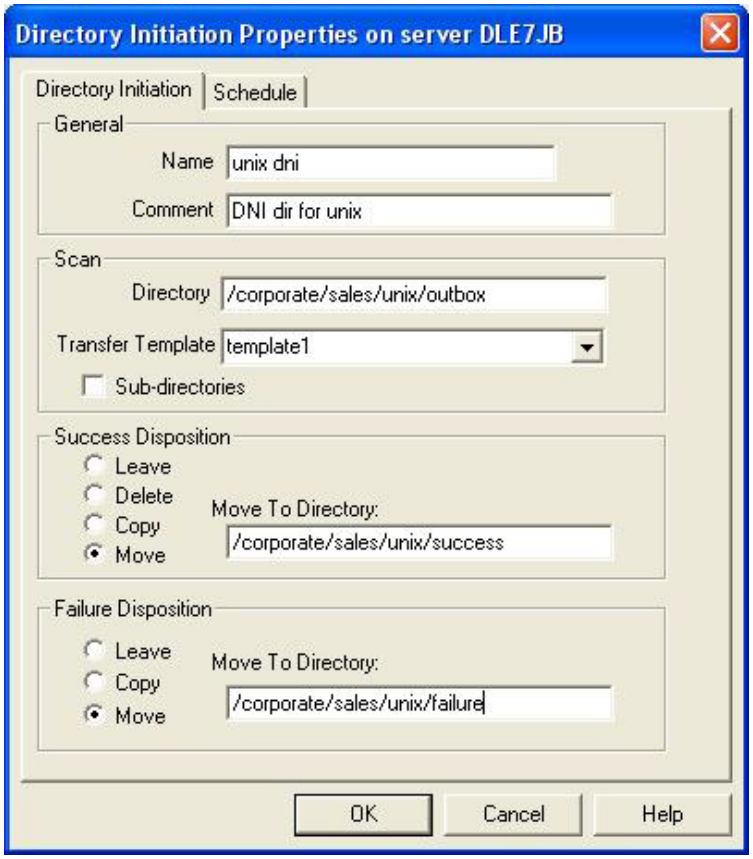
5.6.3 The Initiation Directories Properties Sheet

Use this property sheet to create and maintain DNI information. There are 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 MFT Platform Server's dispatcher becomes active.

When creating a new DNI definition, if a directory does not exist, then a dialog box pops up asking the user if he wants to create the directory. If the user hits on "Yes" then the directory is created, if not the DNI definition is not created because a directory is essential for creating a DNI definition.

5.6.3.1 Directory Initiation Property Page



- **General**

  Name

  A character string which uniquely identifies the DNI entry.

  Comment

  A free text description of the entry.

- **Scan**

  Directory

  The name of the directory to scan for files. This may be the local directory for a Send or a Remote directory for a Receive. *Note: File Name Tokens are not supported in this field.*

  Transfer Template

  The name of the transfer template which will be used to create DNI entries using the file found in the directory. The template must exist in the same MFT Platform Server where the DNI entry will be stored. The Transfer Template drop down box contains a list of the templates that are defined on that MFT Platform Server.

  Sub-directories

  When selected, the service will scan the DNI directory for files as well as all of the sub-directories beneath the DNI directory.

- **Success Disposition**

  Describes what should happen to the scanned file after the DNI entry is dispatched. You may:

  - **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, then the **Copy To / Move To Directory** field will become available.

  Note: Any time the MFT Platform Server service starts, any files remaining in the DNI directory will be re-sent to the pre-defined remote host. Under this condition, files will be re-sent if the Leave or Copy disposition is used. Therefore, if your company policy does not allow for the re-transfer of files, the Move or Delete disposition should be used.

  Note: By default MFTPS will append the file name of the file being transferred to the end of the directory defined in the Copy To Directory or the Move To Directory path. When File Name Tokens are used as part of the Copy To Directory or the Move To Directory MFTPS requires the token(s) used to set the file name.

  Copy To / Move To Directory

  This field is shown when the Disposition of Copy or Move is selected. This field indicates the directory where the source file will be placed once the DNI entry is dispatched. This is especially useful for DNI entries which are configured to receive files.

- **Failure Disposition**

  Describes what should happen to the scanned file after the DNI entry is dispatched. You may:

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

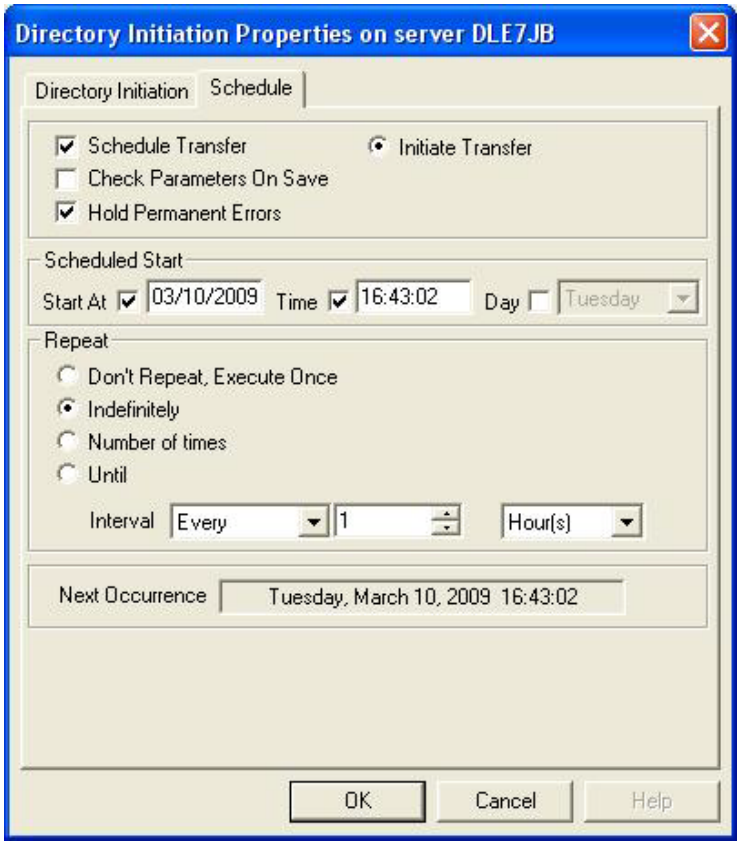
If the disposition is Copy or Move, then the **Copy To / Move To Directory** field also becomes available.

> **Note:** Any time the MFT Platform Server service starts, any files remaining in the DNI directory will be re-sent to the pre-defined remote host. Under this condition, files will be re-sent if the Leave or Copy disposition is used. Therefore, if your company policy does not allow for the re-transfer of files, the Move disposition should be used.

Copy To / Move To Directory

This field is shown when the Disposition of Copy or Move is selected. This field indicates the directory where the source file will be placed once the DNI entry is dispatched. This is especially useful for DNI entries which are configured to receive files.

5.6.3.2 Schedule Property Page



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

- **Schedule Transfer**

  Adds a schedule to the DNI entry.

- **Check Parameters On Save**

  Instructs MFT Platform Server to verify that the file you have scheduled to be dispatched exists at the time of scheduling. This option must be selected in order for this verification to occur.

- **Initiate Transfer**

  Indicates that a DNI entry will be dispatched when the schedule becomes eligible.

Following is a description of the different fields that appear on the Schedule property page.

- **Scheduled Start**

Schedule start provides the information necessary for dispatching a DNI entry in the future. This parameter has 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 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 Start At (date) field.

Note: In order for the information in this field to be applied to the DNI entry, you must select the box that appears to the left of each of the fields.

- **Repeat**

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

  Don't Repeat, Execute Once

  When this option is selected, there will be only one attempt to dispatch this DNI entry.

  Indefinitely

  When this option is selected, the repeat **Interval** field appears 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 repeat Interval field.

  Until

  This option allows the user to specify the date and time or the day of the week until which the DNI entry will be dispatched. When this option is selected, the panel dynamically changes by inserting the fields where you can specify the required information (similar to the Start At field).

  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 5 (Monday to Friday), Daily 7 (Sunday to Saturday), Weekly, Bi-Weekly, Monthly, Bi-Monthly, Quarterly, Semi-Annually, Annually, Bi-Annually, or Every.
  The panel changes if the option Every is selected. The Repeat Interval parameter adds two additional fields that you can use to specify the frequency with which you want to dispatch the DNI entry. 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).

- **Next Occurrence**

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

To save the settings that you specified, click OK.
When new templates or new DNI are created they are backed up to a file called "FTMSSVR.BAK" which resides in the directory where MFT Platform Server is installed. This file is not deleted during the uninstallation process hence the user 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.

5.6.4 DNI Command Line Interface (CLI)

Directory Named Initiation (DNI) allows you to detect the existence of files that have been placed within a directory and/or sub directories and automatically transfer those files to one or more targeted MFT Platform Server remote systems.

When you set up a DNI transfer, it will scan a pre-defined local or remote directory at a user-defined interval. It will return and save a list of all files in that directory. Any files that have changed between the scans are eligible to be transferred. While the DNI CLI was developed for both UNIX and Windows systems the UNIX systems have no standardized way of locking files. So even though Windows supports file locking, the UNIX logic is used on Windows. This is how MFT Platform Server can tell whether the file is in use or not. When a file transfer is complete, DNI allows you to delete the original file, move it to a new directory, or leave the file where it is.

Transfer requests that can be processed using DNI are:

DNI Send
DNI Send reads the directory/s defined and executes a command when it detects a file exists within the directory/s and sends the files to a remote system.

DNI Receive
DNI Receive contacts a remote MFT Platform Server system to extract a list of files in the directory defined. Based on this list, DNI Receive will execute a command to transfer the files to the local machine.

DNI FTP Receive via FTP
DNI FTP Receive contacts the remote FTP system to extract a list of files in the directory defined. Based on this list, DNI RecvFTP will execute a command to transfer the files to the local machine. In most cases this is a **"get"** command.

DNI processing is done using a Perl script called **dni**. As such, in order to use DNI your Windows systems must have a version of Perl installed. The Perl program directory should 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 web site: www.perl.org.

Note: The MFT Platform Server DNI perl script will run on both Windows and UNIX platforms and as such we provide DNI templates for both the Windows and UNIX platform. In this manual we will only discuss the DNI templates for the Windows platform. The DNI templates for the UNIX platform do not work with or complement the MFT Platform Server for Windows DNI feature in any way. For information using MFT Platform Server for UNIX DNI, please refer to the MFT Platform Server for UNIX documentation.

There are many uses for DNI. Some of the more obvious include:

1. To copy entire directory structures from one system to another - You can tell MFT Platform Server DNI to copy all files from SystemA to SystemB and create the same directory structure that exists on SystemA on SystemB using the LocalFileName token. In this example, the DNI template on SystemA can be configured with the PPA SuccessAction set to "leave", the write mode set to CRN and SubDirectory parameter set to Yes for the directory structure to be created. *Note: It is important that the DNI is only from SystemA to SystemB. A second DNI configured similarly to copy from SystemB to SystemA could create an infinite loop.*
2. Copy files to another MFT Platform Server computer. As DNI detects that a file on SystemA has changed, DNI will send the file to SystemB. In this case when the transfer has completed, you can move the file to a backup directory, or delete the file.
3. Execute any Windows command based on the existence of a file. As DNI detects that a file on SystemA has changed, DNI will execute a pre-defined command. This command can include information on the

local file that was the source of the DNI request. It is up to the DNI Administrator to define the actual command that is executed.

5.6.4.1 Installing the DNI Program

DNI is distributed as a tar file called **dni.tar**. It can be found in the MFT Platform Server installation directory. To extract the files, execute the following tar command on the Windows platform

```
tar xvf dni.tar
```

Upon successful execution of this command, you will find the following seven files:

```
dni                 The MFT Platform Server DNI perl script program
dnitemplate.send    The DNI template for UNIX DNI Send (Will not be discussed)
dnitemplate.recv    The DNI template for UNIX DNI Receive (Will not be
  discussed)
dnitemplate.recvftp The DNI template for UNIX DNI Receive via FTP
dnitemplate.wsend   For Windows DNI Send
dnitemplate.wrecv   For Windows DNI Receive
dnitemplate.wrecvftp For Windows DNI Receive via FTP (Will not be discussed)
```

The dnitemplate.wsend, dnitemplate.wrecv, and dnitemplate.wrecvftp scripts should be used when the script execution is on a Windows platform.

Note a few things about the dni template files:

1. Any data in a line after the # is considered a comment.
2. If a line consists of all blanks, then the line is ignored.
3. All parameters are in the format of xxx: yyy where xxx is the parameter name, and yyy is the parameter value.
4. Parameter names and values are not case sensitive, although we suggest leaving them mixed case for easier reading.
5. The LocalFileName (lf) and RemoteFileName (rf) parameters must be specified in the DNI template. If you are using a transfer template within your DNI template the LocalFileName (lf) and RemoteFileName (rf) parameters defined in the DNI template will override these values defined in a transfer template.

5.6.4.2 Running the DNI Perl Script

The following command is used to execute the DNI perl script:

```
perl dni t:dnitemplatename {optional parameters}
```

If the dnitemplate is not in the current working directory, then you must define the fully qualified template name so that DNI can read the contents of the template file.

DNI Script Optional Parameters:

Parameter	Description
cold	The “Cold” option bypasses the reading of the “Leave” file that contains the list of files already transferred that is saved when the SuccessAction and/or FailureAction is set to “leave”. As defined in the warm option description. As such, DNI will transfer any file in the scan directory when the script is started. This option should be used with great care.
warm	This parameter is only used when a DNI SuccessAction or FailureAction is defined as “Leave”. Due to this DNI will keep track of files that have been transferred and store the information in a file. The warm option tells DNI to read the “Leave” file and at start up and process the files normally as if MFT Platform Server did not come down. This makes sure that a file is not transferred multiple times by mistake. (This is the default when no option is set.)
hot	When defined, DNI will add all files detected on the first directory scan to the “Leave” file. Therefore the contents of the directory when DNI is started will not be transferred. Note

Parameter	Description
	that a Hot start creates the "Leave" file with the contents of the directory after the first scan. If DNI Receive is unable to get this list from the remote system due to a network error, then Hot start processing will not be performed.
debugon	Turns on DNI debugging. This parameter overrides the "debug" parameter in the MFT Platform Server template. This parameter should only be used when instructed to do so by TIBCO Technical Support.
debugoff	Turns off DNI debugging. This parameter overrides the "debug" parameter in the MFT Platform Server template. This parameter should only be used when instructed to do so by TIBCO Technical Support.
version (-v)	This option will display the dni version and PTF level.

5.6.4.3 DNI Template Parameters

DNI templates define the source directory to scan and the information necessary to run a transfer. Enter the following command for more details on the DNI template parameters:

perl dni help template

5.6.4.4 DNI Tokens

DNI Tokens: For details on the various tokens supported enter the following command:

perl dni help tokens

5.6.4.4.1 Special SubDir Token Use

Note that there is a special use for the $(SubDir) token. Let's say that you have 5 Windows systems running MFT Platform Server, and that you want to send files to each of the 5 systems. You could set up 5 DNI executions, each of which has a template configured especially for that system. A second option is to set up your directory structure in such a way that the subdirectory name matches the Node name for the remote systems.

As an example, let's say that you have 5 Windows systems running MFT Platform Server. They are called:

WINRed, WINBlue, WINGreen, WINBlack, WINWhite

You could set up your directory structure in the following manner:

Main directory	**Sub Directories**
DNI	WINRed
	WINBlue
	WINGreen
	WINBlack
	WINWhite

You could set up your DNI transfer fields as such, so that a single DNI task could handle transfers to all 5 of the Windows systems.

LocalDirectory:	**/DNI**
DNICommand:	**ftmscmd /tcp /send /file /node:$(SubDir) /rd:domainname "c:\DNI\Source\$(LocalFileName)" "c:\DNI\Target\$(LocalFile)**

Note that the DNICommand will override the Node parameter (n:) with the name of the subdirectory. In this case, you must define MFT Platform Server nodes for each of the subdirectories using the cfnode program. In other words, you must execute the cfnode program 5 times to add node definitions for WINRed, WINBlue, WINGreen, WINBlack and WINWhite.

When file "AcctData" is added to the WINRed directory, DNI will detect the file, and execute the DNICommand. Before executing the DNICommand, it will perform token substitution, and replace the value $(SubDir) with the value WINRed. Therefore, the DNICommand will be changed to the following:

```
ftmscmd /tcp /send /file /node:WINRed "c:\DNI\WINRed\AcctData"
"c:\DNI\Target\AcctData"
```

Another use for the $(SubDir) token is to specify a subdirectory name when performing Post Processing actions. If you want to move files based on success or failure and you want to retain the directory structure, you can use the $(SubDir) token.

5.6.4.5 DNI Help Screens

There are four help screens that give information on how the MFT Platform Server DNI script can be used. This can be executed by entering the following commands:

perl dni help	General help information
perl dni help template	Template configuration information
perl dni help tokens	Substitutable Token information
perl dni help encrypt	Creates an encryption $(Password) token

5.6.4.6 Using DNI Receive FTP with MFT Internet Server

DNI Receive FTP has some special advantages when used with the MFT Internet Server FTP server. MFT Internet Server can be configured to allow a single user to access multiple Servers transparently within a single session. Because of this support, DNI Receive FTP can then access multiple FTP servers and automatically download data from these servers within a single process.
MFT Internet Server contains both an FTP server as well as an FTP client. A user sitting at their desk can use the Windows FTP client to connect to the MFT Internet Server FTP server. After the user logs in, the MFT Internet Server FTP Client can connect to multiple FTP servers at customer sites to extract data. IN this case, DNI Receive FTP performs the functions of the Windows FTP Client automatically without user intervention.

In order to understand how this works, you need a solid understanding of MFT Internet Server definitions. This information is not contained in this manual. Refer to the "MFT Command Center, Internet Server and Platform Servers Documentation" manual for more information on how to configure MFT Internet Server. We will be discussing the following MFT Internet Server definitions:

User Definition	Create a MFT Internet Server user and assign rights to that user
Server Definition	Define the parameters needed to communicate with a remote server
Transfer Definition	Defines parameters needed for the User to transfer with the defined Server definitions

Below is a diagram that shows how a single user can automatically download data from multiple FTP servers using the MFT Internet Server, in order to automatically download data from each of the customer FTP servers, you must make the following MFT Internet Server definitions.

User definition
You must add a User definition for the DNI Receive FTP process to use. You must assign TransferRight to this user. Remember the userid and password assigned to this user because you must configure this information into the DNI template FTPUser and FTPPassword parameters.

```
User=FTPUSER
```

Server Definitions
You must add five Server definitions; one for each customer server. You must define the TCP information necessary to access these machines. Since MFT Internet Server supports multiple server protocols, you must define the Server type. Additionally, you need to define the security credentials necessary to logon to each customer server. This can be done within the server definition, or Server Credentials can be added for the user and server to provide a finer level of security control. Note that the server definitions can be for FTP servers (with or without SSL), SSH Servers, Platform Servers or data stored on the local hard disk.

```
Server=Customer1,IPAddr=10.1.1.100,IPPort=21,ServerType=FTP
```

Server=Customer2,IPAddr=10.1.2.100,IPPort=22,ServerType=SSH
Server=Customer3,IPAddr=10.1.3.100,IPPort=21,ServerType=FTP
Server=Customer4,IPAddr=10.1.4.100,IPPort=21,ServerType=FTP
Server=Customer5,IPAddr=10.1.5.100,IPPort=46464,ServerType=Platform Server

Transfer Definitions

You must add five Transfer definitions; one for each customer FTP server. The transfer definition defines the following information:

: The user authorized to use this Transfer Definition
: The Server that will process this transfer definition
: The Starting point (i.e. Home Directory) within the server where transfers should start.
: The name that is shown in the client software. This is referred to as the FTPAlias

Transfer1,User=FTPUSER,Server=Customer1,FTPAlias=Directory1,ServerFile=/dniftp/download
Transfer2,User=FTPUSER,Server=Customer2,FTPAlias=Directory2,ServerFile=/dnissh/download
Transfer3,User=FTPUSER,Server=Customer3,FTPAlias=Directory3,ServerFile=/dniftp/download
Transfer4,User=FTPUSER,Server=Customer4,FTPAlias=Directory4,ServerFile=/dniftp/download
Transfer5,User=FTPUSER,Server=Customer5,FTPAlias=Directory5,ServerFile=/dnicfi/download

DNI Receive FTP Configuration

The following parameters must be defined in the Template file:

FTPHost: 192.168.100.1
FTPPort: 9021
FTPUser: FTPUSER
FTPPassword: xxxxxxx
FTPRemoteDirectory /
FTPLocalDirectory c:\download
FTPSlash /
FTPSubdirCount: 1
SubDirectory: Yes

DNI FTP Processing

Below is a description of what happens when the above configurations are defined.

1. DNI connects to the FTP Server using the FTPHost and FTP Port parameters
2. DNI logs onto the FTP Server using the FTPUser and FTPPassword parameters
3. DNI issues a command to extract all files and directories within directory"/".
4. MFT Internet Server will then connect to each of the five Servers defined, and extract the files and directories within the directory defined by the ServerFile parameter. MFT Internet Server returns this information to DNI.
5. DNI in this case will detect 5 directory entries and 0 files:
   : Directory1
   : Directory2
   : Directory3
   : Directory4
   : Directory5
6. Because DNI was configured to process subdirectories (SubDirectory=Yes and FTPSubdirCount=1), DNI then extracts the files and directories within the 5 directories shown above.
7. Because the FTPSubdirCount parameter is defined as 1, no further subdirectories will be processed.
8. DNI will then review each file against the "leavefile". This leave file contains the date, time and file size of each file that has been successfully downloaded. If the file has been changed since the last time it was processed or if it has never been processed, it will be downloaded; otherwise, it will not be downloaded.
9. When DNI detects that a file has changed, it will be downloaded via the FTP Get command. When the file has been successfully downloaded, the "leavefile" will be updated with the file date, time and size.
10. When all possible files have been processed, DNI will sleep for the interval defined by the ScanInterval parameter. When it wakes up again, processing will start again at Step 1.

To summarize, DNI will make a single connection into the MFT Internet Server FTP Server. MFT Internet Server will then connect into each Customer FTP server to extract the data needed by DNI. This is all done automatically without the user (in this case DNI) being aware that it happened.

5.7 Directory Transfer and Wildcard Support

MFT Platform Server can send and receive Directories. If Local File Name or Remote File Name contains * and/or ? after last / in file path, then a Directory Transfer will be performed

5.7.1 Directory Transfer Parameters

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

StopOnFailure
If the current file transfer fails, MFT Platform Server will not try to transfer the rest of files. The default for this is Yes.

Test
Allows the user 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. The default for this is No.

5.7.2 Tokens for Local and Remote File Names

$(SDIR)
This case sensitive token may be used with a Receive as part of the LocalFileName path, and with a Send as part of the RemoteFileName path.

Example:
For a Receive, you would set:

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

RemoteFileName:
C:\TransferFiles\data*

The text before this token is assumed to be a base directory.
If **ScanSubDir** is checked on and there are files in both the remote directory (**C:\TransferFiles\data**) and in the remote subdirectories, then the same subdirectories will be created in the local directory (**C:\johndoe\data**) and local file names will be given as **$(RemoteFileName)** token.

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

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

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

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

$(MEMBER)

This token should be 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 than directory names.

So, this token allows you 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, this token will not be used. For example, if the path was **C:\TransferFiles\$(MEMBER)\whatever**, it will become **C:\TransferFiles\whatever**.

5.7.3 Wildcard Information

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

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

Only those file names that satisfy the selection criteria will be transferred.
For example, the name **c:\johndoe\r?t*** will match the files **c:\johndoe\returns** and **c:\johndoe\ratelist** but not the name **c:\johndoe\report.** To transfer an entire directory, a single * should be used.

5.7.4 General Information

- The File Path can contain both / and \ slashes.
- Be aware of the **Creation Option** parameter. If a directory is transferred without scanning the subdirectories and there are no files with the same names specified in the receiving directory, the **Creation Option** must be **CreateReplace** or **CreateReplaceNew**.
- If subdirectories are transferred as well and there are file names that match the specified names in the receiving directory, the **Creation Option** must be **CreateReplaceNew**; if it is **CreateReplace**, the subdirectories will **NOT** be created.
- Selected files are sent to or received from the remote side sequentially, each with a separate Local and Remote Transaction numbers and thus separate entries in the log files. Only after a file transfer is complete will the next file be transferred.
- With the exception of the new parameters listed in the Directory Transfer Parameters section of this document, all parameters are applicable per file transfer.

- **Try Count** will be the same for all files in the directory and the next file will not be transferred until all attempts to transfer the previous one are made.
- **CheckPointRestart** works for the directory/wildcard transfers as well. In other words, if something occurs during the directory transfer and the MFT Platform Server Service is stopped, when the MFT Platform Server Service is restarted, it will finish the checkpointed file, and then continue with the rest of the files specified in the directory transfer.

5.8 FUSPING Utility

The FUSPING Utility is used to find out if MFT Platform Server is running on a remote platform.

5.8.1 Format of fusping commands

Below is the usage of the fusping command:

```
usage: fusping [parameters:][values]
[parameters]:
r: or Request:       - Type of Request. Only "Version" is supported
h: or Host and Port: - IpAddress:PortNumber or IpName:PortNumber
t: or Trace:         - if Yes MFT Platform Server Trace is turned on
?:                   - Help
```

5.8.1.1 Examples:

Below are some examples of how to use the **fusping** utility to check whether MFT Platform Server is running on the remote system as well as the version of MFT Platform Server.

This example is checking a remote mainframe platform:

```
fusping  r:version  h:11.22.33.55:46464
```

Output:

```
Host:             11.22.33.55
Port:             46464
System Name:      Name=A390,STC=CFUSN60  ,CPUType=1234,CPUID=5555
Key Expiration:   20110516
Version:          MFT Platform Server for z/OS,Version=70 ,PTFLevel=CZ01852
```

This example is checking a remote Windows platform:

```
fusping  r:version  h:11.22.33.44:46464
```

Output:

```
Host:             11.22.33.44
Port:             46464
System Name:      WIN44
Key Expiration:   20121019
Version:
                  Ftms32.DLL, Version 7.1.0 (Build 1030 UNICODE)
                  FtmsDni.DLL, Version 7.1.0 (Build 1030 UNICODE)
                  FtmsTcpS.DLL, Version 7.1.0 (Build 1030 UNICODE)
                  FtmsVer.DLL, Version 7.1.0 (Build 1030 UNICODE)
                  FusionMs.DLL, Version 7.1.0 (Build 1030 UNICODE)
                  HoLib.DLL, Version 7.1.0 (Build 1030 UNICODE)
                  HOTrace.DLL, Version 7.1.0 (Build 1030 UNICODE)
                  InstSvc.DLL, Version 7.1.0 (Build 1030 UNICODE)
                  SMTPDll.DLL, Version 7.1.0 (Build 1030 UNICODE)
                  FtmsMgr.EXE, Version 7.1.0 (Build 1030 UNICODE)
                  FtmsCmd.EXE, Version 7.1.0 (Build 1030 UNICODE)
                  FtmsMon.EXE, Version 7.1.0 (Build 1030 UNICODE)
                  FtmsSvr.EXE, Version 7.1.0 (Build 1030 UNICODE)
                  FusionVer.EXE, Version 7.1.0 (Build 1030 UNICODE)
```

5.9 FUSUTIL: Delete, Rename, Exist Utility

When a file transfer completes, the user may 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. This utility will allow us to use a common interface to rename or delete a file. Additionally, it will give us the ability to see if a file exists on a remote platform.

5.9.1 Description

There are three functions of the **fusutil** utility.

1. Delete File
2. Rename File
3. Determine if a file exists

When a request is received by the MFT Platform Server, the request should be converted to the proper request for that operating system. The following table shows the relationship between the **fusutil** command and the operating system command.

Function	Alternate Parameter	Windows equivalent command
RENAME	R	move
DELETE	D	erase
EXIST	E	N/A

Note that there must be one or more spaces between the command parameters.

5.9.2 Format of fusutil commands

The command **fusutil** should be configured as a post processing action using the COMMAND option.

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

The second parameter is required and is the request type:

Function	Alternate Parameter	Description
RENAME	R	Rename a file
DELETE	D	Delete a file
EXIST	E	See if a file exists

The third parameter is required and defines the file name for each option. The fourth parameter is required only for Rename; it defines the new file name.

5.9.2.1 Examples:

Below are some examples of how to use 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>
```

5.9.3 Special Processing

When processing the EXIST function, you should also check if the file is available for use. This should be done on all platforms except UNIX, since there is no standard call to accomplish this on UNIX.

5.9.4 Return Codes

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

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

5.10 Node Definitions and User Profiles and Distribution Lists

Node definitions and User Profiles and distribution lists are used to define all information needed to interact with a single MFT Platform Server remote system or multiple. This frees the user from needing to constantly provide this information to MFT Platform Server.

Node definitions are used to define information about a remote system (node). They are stored in a clear text 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.

User Profile definitions are used to define local users and corresponding remote users per node. User Profiles are stored in a clear text file named cfprofile.cfg located in the MFT Platform Server directory. Passwords are stored in an encrypted format to ensure maximum security. The MFT Platform Server cfprofile command is to be used to add and update profile definitions in the cfprofile.cfg file.

Responder Profiles define a local username and password that should be 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.

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

The following section of the documentation describes Nodes. Profiles may also be defined to be associated with these nodes. User Profiles and Responder Profiles are described later in the chapter.

5.10.1 Node Definition

Node definitions define default parameters needed by MFT Platform Server to interact with a remote system (node). This information includes:

- IP address for TCP/IP transfers
- Port number for TCP/IP transfers
- LU for SNA transfers
- Mode Name for SNA transfers
- Use of SSL for secure communications
- Default type of compression to use for transfers
- Default type of encryption to use for transfers
- HIPAA or FIPS-140 compliance
- Remote and Local Translation tables
- Command Center Support

Once a node definition is created, a user may specify the name of the node to be used when executing a transfer. MFT Platform Server will consult the definition for the specified node to obtain the parameters needed to execute a transfer.

Node definitions are stored in a file named cfnode.cfg located in the MFT Platform Server directory. Although node parameters may be added and updated using a text editor, it is strongly suggested that all node definitions be created and updated using the MFT Platform Server cfnode maintenance command. Before cfnode updates any information in cfnode.cfg, a backup of this file is created called cfnode.bak. When defining nodes in various file, such as CfAlias.cfg, make certain that you use the proper case as these files are case-sensitive.

Below is a sample of a node definition created using the cfnode command.

```
[dataServerA]
   SystemType                  = Windows
   Protocol                    = sna
   RemoteLocation              = LuName
   ModeName                    = #BATCH
   Compression                 = RLE
   Encryption                  = NO
   RemoteCTFile                = rmttrans.txt
   Description                 = This is a sample SNA node definition

[dataServerB]
   SystemType                  = Windows
   Protocol                    = tcpip
   HostName                    = 111.222.33.44
   Server                      = 46464
   SSL                         = N
   Compression                 = RLE
   Encryption                  = BFL
   SecurityPolicy              = HIPAA
   RemoteCTFile                = remotetrans.txt
   ResponderProfile            = D
   Description                 = Sample TCP node
   CommandSupport              = PING
```

5.10.2 Node Parameters

The following parameters may be specified in the cfnode command line. If you do not specify these parameters, and the **prompt:** parameter is specified, the user will be prompted for all information needed to successfully execute cfnode.

action

This is an optional parameter. The action parameter is used to specify the action to be taken. Valid values are Delete, List and Add.

```
ex:     C:\Program Files\TIBCO\MFT Platform Server\System>cfnode action:delete
n:192.168.20.53
```

command line option: a

commandsupport

This is an optional parameter. The actions this node will allow MFT Command Center to perform. Valid Command Center values are:

- ALL - NODE, PROFILE, AUDIT, ALTER, PING and TRANSFER will be supported on this node
- NONE - NO command Center functions will be supported on this node. This is the default if the parameter is not defined
- AUDIT - This node will allow requests that inquire on the MFT Platform Server AUDIT file.
- NODE - NODE List and Update functions will be accepted on this node.
- PING - This node will support MFT Platform Server PING requests.
- PROFILE - List and Update functions will be accepted on this node.

- TRANSFER - This node will support the Command Center Transfer function that initiates file transfers.

The cfnode command will not require the Command Center Support parameter be defined if the prompt:NO parameter is supplied.
ex: ccc:PING
commandsupport:TRANSFER

command line option: ccc

compress

This is an optional parameter. The compress parameter is used to specify the default compression type to use for all transfers with this node. Valid compression values are:

- LZ
- RLE
- ZLIB
- NO – No default compression
- NEVER – Never use compression

All default compression types may be overridden on the command line when transferring a file. The cfnode command will not require the compress parameter be defined if the prompt:NO parameter is supplied.
ex: compress:LZ
c:NEVER

command line option: c

description

This is an optional parameter. The description parameter is used to specify a text description of the node definition. The description may be up to 256 characters and may contain spaces. If the description contains spaces, then it **must** be encapsulated in double quotes. The cfnode command will not require the description parameter defined if the prompt:NO parameter is supplied.
ex: description:"This is a sample description"
d:"This is a sample description"

command line option: d

encrypt

This is an optional parameter. The encrypt parameter is used to specify the default encryption type to use for all transfers with this node. Valid encryption values are:

- DES
- 3DES – Triple DES
- BF – Blow Fish Encryption
- BFL – Blow Fish Long
- RIJN - Rijndael
- NO – No compression
- NEVER – Never use compression

All default encryption types may be overridden on the command line when transferring a file. The cfnode command will not require the encrypt parameter defined if the prompt:NO parameter is supplied.
ex: encrypt:DES
e:NEVER

command line option: e

hostName

This is a required parameter for TCP. The hostName parameter is used to specify the IP address of the node. This value should be the dotted IP address of the remote machine, but may be a resolvable host name. If this parameter is not supplied and the prompt:NO parameter is supplied, cfnode will fail.

ex: hostname:11.22.33.44
h:computer.domain.com

command line option: h

lct

Local Conversion Table (also referred to as the Local Translation File) is an optional parameter. This parameter will contain the name of the file, which will be used to translate the data on the local side.

ex: lct:convert.txt

command line option: lct

lu

This is a required parameter for SNA. This is the LU Name of the remote system. This would be the name that would be used in the Destination field on a MFT Platform Server transfer.

ex: lu:RMTLU
l:RMTLU

command line option: l

modeName

This is a required parameter for SNA. This is the Mode Name that is associated with the LU Name of the remote system.

ex: modeName:#BATCH
m: #BATCH

command line option: m

netMask

This is an optional parameter for TCP. NetMask for remote IpAddress.

ex: netMask:255.255.255.0
net:255.255.255.128

command line option: net

node

This is a required parameter. The node parameter is used to specify the name of the node to be added or updated to the cfnode.cfg file. The node name may be up to 256 characters long and may not contain any spaces. If this parameter is not supplied and the prompt:NO parameter is supplied, cfnode will fail.

ex: node:dataserver
n:dataserver

command line option: n

port

This is a required parameter for TCP. The port parameter is used to specify the port number on which the remote node is listening. If this parameter is not supplied and the prompt:NO parameter is supplied, cfnode will fail.

ex: port:46464
p:46464

command line option: p

prompt

The prompt parameter should be used to put cfnode into an interactive mode. If prompt:YES is supplied, cfnode will prompt the user for all information needed to create a node. Using the prompt: parameter will also ask if the user would like to create cfnode.cfg if it could not be found. Prompting is turned on by default. If the user does not wish to be prompted, then prompt:NO should be used.

rct

Remote Conversion Table (also referred to as the Remote Translation File) is an optional parameter. This parameter will contain the name of the file, which will be used to translate the data on the remote side.

ex: rct:convert.txt

command line option: rct

responder

This is an optional parameter. This parameter is whether a responder profile should be used for this node. Valid values are Yes, No or Dual. A value of D (Dual) means that the substitution of a real userid will occur only if the cfrprofile exists and a match is found. If there is no match found, then MFT Platform Server will attempt to login remote user with the userid and password they sent, rather than generate an error message that cfrprofile does not exist or the information does not match. On the other hand, a value of "Yes" will not try to login the user with the userid and password they sent, and a value of "No" will not check the responder profiles at all.

ex: responder:Yes
r:Y

command line option: r

ssl

This is an optional parameter. The ssl parameter is used to specify whether SSL should be used for TCP/IP communications. The cfnode command will not require the ssl parameter be defined if the prompt:NO parameter is supplied.

ex: ssl:Y

command line option: ssl

security

This is an optional parameter. This parameter will determine whether the node is HIPAA or FIPS-140 compliant. If set, only HIPAA or FIPS-140 compliant encryption types will be listed. See System Configurations: Security Policy for more information.

ex: security:HIPAA

command line option: sl

systemType

This is a required parameter. The systemType parameter is used to specify the type of system represented by this node definition. Valid system types are:

- HPUX
- SUN/SOLARIS
- AIX
- LINUX
- Windows
- AS/400
- OS/390
- z/OS
- Other

If this parameter is not supplied and the prompt:NO parameter is supplied, cfnode will fail.

ex: systemType:Windows
s:Windows

command line option: s

/?

The /? (or -?) parameter should be used to display the online help for cfnode. Online help is as follows:


```
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.)
   l: or lu:                - LU Name of the remote node. (This parameter is only required
                              for SNA transfers.)
   m: or modeName:          - Mode Name used by SNA. (This parameter is
                              only required for SNA 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.
   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)
   e: or encrypt:           - Type of default encryption to use duringtransfers. Valid
                              encryption types:
                              (DES | 3DES | BF | BFL | RJ | NO)
   sl: or security          - Security Compliance level to use during
                              transfers. Valid security types:
                              (Default | None | HIPAA)
   lct:                     - Local translation file. Valid values:
                              file path.
   rct:                     - Remote translation file. Valid values:
                              file path.
   v: or verify:            - Either Yes or No depending on whether or not to
                              AcceptVerifiedUsers from this node.
   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)
   prompt:                  - Prompts the user for corrections when errors are found.
                              Valid values:(YES|NO). Default is YES.
   /? or -?                 - Online help.
```

5.10.2.1 Examples using cfnode

Below is a sample of how cfnode is used with the command line options:
At the command prompt the following is done:

```
C:\>cd Program Files\TIBCO\MFT Platform Server\System
C:\Program Files\TIBCO\MFT Platform Server\System>cfnode n:dataServerA s:Windows
lu:LuName m:#BATCH c:RLE e:NO rct:rmttrans.txt d:"This is a sample SNA node definition"
prompt:NO
```

Below is a sample of cfnode 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: AS/400
7: OS/390
8: z/OS
```

```
9: Other
: 5
What should be the transfer protocol for this node?
1: TCP/IP
2: SNA
:1
Would you like to specify netmask for remote IpAddress:
1: Yes
2: No
: 2
Enter a valid IP address for Node [dataServerB]: 111.222.33.44
Enter the port for which Node [dataServerB] is configured to use:46464
Should SSL be used:
1: Yes
2: No
: 2
What should be the default compression used:
1: LZ
2: RLE
3: No default compression
4: Never use compression
: 2
Enter the Security Compliance level for file transfers:
1: Default ( use Security Policy from Server Property )
2: None
3: HIPAA
: 1
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
: 4
Would you like to specify local translation file:
1: Yes
2: No
3: None
: 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 Internet Server parameters this node will support:
1: All
2: None
3: Audit
4: Node
6: Profile
7: Transfer
99: No more parameters
: 99

A Node definition was created for:
[dataServerB]
   SystemType               = Windows
   Protocol                 = tcpip
   HostName                 = 111.222.33.44
   Server                   = 46464
   SSL                      = N
   Compression              = RLE
   Encryption               = BFL
   SecurityPolicy           = HIPAA
   RemoteCTFile             = remotetrans.txt
   ResponderProfile         = D
   Description              = Sample TCP node
   CommandCenterSupport     = PING
```

The prior samples using the cfnode commands updated a **cfnode.cfg** file with the following contents:

```
[dataServerA]
   SystemType               = Windows
   Protocol                 = sna
   RemoteLocation           = LuName
   ModeName                 = #BATCH
   Compression              = RLE
   Encryption               = NO
   RemoteCTFile             = rmttrans.txt
   Description              = This is a sample SNA node  definition

[dataServerB]
   SystemType               = Windows
   Protocol                 = tcpip
   HostName                 = 111.222.33.44
   Server                   = 46464
   SSL                      = N
   Compression              = RLE
   Encryption               = BFL
   SecurityPolicy           = HIPAA
   RemoteCTFile             = remotetrans.txt
   ResponderProfile         = D
   Description              = Sample TCP node
   CommandSupport           = PING
```

Node definitions may be modified and deleted from **cfnode.cfg** by using a text editor and deleting the entire node definition from **cfnode.cfg**.

The nodes may also be listed or deleted using the “action” parameter. A sample of how to list nodes is shown below:

```
cfnode a:list
[dataserverA]
   SystemType               = Windows
   Protocol                 = sna
   RemoteLocation           = LuName
```

```
   ModeName              = #BATCH
   Compression           = RLE
   Encryption            = NO
   RemoteCTFile          = rmttrans.txt
   Description           = This is a sample SNA node definition
 [dataServerB]
   SystemType            = Windows
   Protocol              = tcpip
   HostName              = 111.222.33.44
   Server                = 46464
   SSL                   = N
    Compression          = RLE
   Encryption            = BFL
   SecurityPolicy        = HIPAA
   RemoteCTFile          = remotetrans.txt
   ResponderProfile      = D
   Description           = Sample TCP node
   CommandSupport        = PING
```

5.10.3 User Profiles

User Profiles define a remote login and remote password to be used per local user and each node definition defined in **cfnode.cfg**. When a node is supplied on the Node tab, a user profile is chosen for the node based on the current logged in user and the information in that user profile is used to log on to the remote system. A MFT Platform Server User Profile contains the following information:

- Node for which the User Profile is valid.
- Local User Name who will use this User Profile.
- Remote User Name to use to log on to the node.
- Remote Password to use to log on to the node (in an encrypted format).

Profile definitions are stored in a clear text file named cfprofile.cfg located in the MFT Platform Server directory. User Profiles that are to be added and updated must be done so through the MFT Platform Server cfprofile command. Before cfprofile updates any information in cfprofile.cfg, a backup of this file is created called cfprofile.bak.

5.10.3.1 User Profile Parameters

You can specify the following parameters in the **cfprofile** command line. If you do not specify these parameters, and the prompt: parameter is specified, the user will be prompted for all information needed to successfully execute **cfprofile**.

Parameter Name	Command Line Option	Short Description
action	a	Optional. Actions allowed
localUser	l	Optional. Local User Name to Assume
node	n	Required. Name of Node
password	p	Required. Remote Password
prompt		Optional. Prompts user to supply all valid
user	u	Required. Remote User Name
/?	-?	Online help.

action

This is an optional parameter. The action parameter is used to specify the action to be taken. Valid values are Delete, List and Add. The **cfprofile** command will not require the action parameter be defined if the **prompt:NO** parameter is supplied.

ex: action:delete
a:delete

command line option: a

localUser

This is an optional parameter. The localUser parameter is used to assume the identity of a different local user on the local system. This allows userA to add a User Profile for userB without having to be logged in as userB. Only an Administrator or a member of the cfadmin group may use this option. If the localUser parameter is not supplied and the prompt:YES parameter is supplied and the logged in user is an Administrator or a member of the cfadmin group, the user will be prompted as to whether they would like to assume another local user.

The Administrator or a member of cfadmin may use the localUser option to create a User Profile that can be used by all local users who wish to command transfers with a particular node. In this case the localUser option should be coupled with the *ALL option. If there is no User Profile for the current user on a given node but there is an *ALL entry defined, MFT Platform Server will use the *ALL User Profile for transfers.

ex: localUser:john
l:*ALL

command line option: l

node

This is a required parameter. The node parameter is used to specify the name of the node to which the User Profile to be updated/added. The node name may be up to 256 characters long and may not contain any spaces. If this parameter is not supplied and the prompt:NO parameter is supplied, cfprofile will fail. A node must already exist in cfnode.cfg in order to successfully add or update a user profile.
ex: node:dataserverA
n:dataserverB

command line option: n

password

This is a required parameter and is case sensitive. The password parameter is used to specify a password to be used to log on to the remote node. If this parameter is not supplied and the prompt:NO parameter is supplied, cfprofile will fail.
ex: password:apple
p:computer

command line option: p

prompt

The prompt parameter should be used to put cfprofile into an interactive mode. If prompt:YES is supplied, cfprofile will prompt the user for all information needed to create or update a user profile. Using the prompt:YES parameter will also ask the user if they would like to create cfprofile.cfg if it could not be found. Prompting is turned on by default. If the user does not wish to be prompted they should supply prompt:NO.

user

This is a required parameter. The user parameter is used to specify the user name to be used to log on to the remote system. The userid may be up to 31 characters in length which includes fifteen characters for a machine name, a slash and up to fifteen characters for the userid. The userid is generally not case sensitive, unless going to a UNIX system. If the remote node is a Windows system, the domain must also be specified using either of the following formats: domain\\username or domain/username

If this parameter is not supplied and the prompt:NO parameter is supplied, cfprofile will fail.
ex: user:kenny
u:bob

command line option: u

/?

The /? (-?) parameter should be used to display the online help for cfprofile. Online help is as follows:

```
usage cfprofile [required-parameters] [optional-parameters]
[required-parameters]:
   n: or node:            - Name of Node
   u: or user:            - User ID to be used on the remote node. If the remote node
                            is a Windows machine the domain must also be specified
                            using the following format: domain\\userID or
                            domain/userID
   p: or password:        - Password to be used on the remote node.
[optional-parameters]:
   a: or action:          - Following values are allowed:
                                  : Delete  Nodename is required
                                            localUser is Admin option
```



```
                                  : List    Nodename is optional
                                            localUser is Admin option
                                  : Add (Default value)
   l: or localUser:       - Specifies the use of a local user name other than the one
                            currently logged in. Note: Only root or a member of the
                            cfadmin group may use this parameter.
   prompt:                - Prompts the user for corrections when errors are found.
                            Valid values: (YES | NO). Default is YES.
   /?                     - Online help.
```

5.10.3.2 Examples Using cfprofile

Below is a sample of 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.
```

Below is a sample of cfprofile 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 updated a **cfprofile.cfg** file with the following contents:

```
[dataserverA]
   Admin = Secure       kenny
8eb26af8131f0634820482c79c83ff1b68584c8aa2f549eb10e984155eef


[dataserverB]
   *ALL    = Secure            bob
84e053ab10463b6ea6c105e2c9bdbaadebc11b1ab9ba58774343702fbff
```

A User Profile may be deleted from **cfprofile.cfg** by using a text editor and deleting the desired User Profile line from the file.

The user profiles may also be listed or deleted using the “action” parameter. A sample of how to list profiles is shown below:

```
cfprofile a:list

[dataserverA]
 Local User        = root
 Remote User       = kenny

[dataserverB]
```

```
Local User        = *ALL
Remote User       = bob
```

5.10.4 Responder Profiles

Responder Profiles define a local username and password that should be 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. A responder profile contains the following information:

- Remote User Name – Userid to be supplied by the remote system initiating the transfer. (Does not have to be a valid username on the local system.)
- Remote Password – Password to be supplied by the remote system initiating the transfer. If the remote user is an already verified user, this parameter should be *VER.
- Local User Name – Userid to be used by MFT Platform Server when processing a transfer on your local machine from the specified Remote User.
- Local Password – Local password associated with the Local userid.

Responder profile definitions are stored in a clear text file named cfrprofile.cfg located in the MFT Platform Server System directory. Inside the cfrprofile.cfg file all password information is encrypted. Responder profiles that are to be added or updated must use the MFT Platform Server cfrprofile command. Before cfrprofile updates any information in cfrprofile.cfg, a backup of this file is created called cfrprofile.bak.

5.10.4.1 Responder Profile Parameters

You can specify the following parameters in the cfrprofile command line. If you do not specify these parameters, and the prompt: parameter is specified, the user will be prompted for all information needed to successfully execute cfrprofile.

Parameter Name	Command Line Option	Short Description
lPass	lp	Required. Local Password
lUser	l	Required. Local User ID
node	n	Required. Name of Node
prompt		Optional. Prompts user to supply all valid
rPass	rp	Required. Remote Password
rUser	r	Required. Remote User ID
/?	-?	Online help.

lPass

This is a required parameter. The lPass parameter is used to specify the local password associated with the local userid. This must be a valid username on the local system. If the parameter is not supplied and the prompt:NO parameter is supplied, cfrprofile will fail.

command line option: lp

lUser

This is a required parameter. The lUser parameter is used to specify the local username to be mapped to the incoming remote user name. This must be a valid username on the local system. If the parameter is not supplied and the prompt:NO parameter is supplied, cfrprofile will fail.

ex: lUser:john
l:john

command line option: l

node

This is a required parameter. The node parameter is used to specify the name of the node to which the Responder Profile to be updated/added will be coupled. The node name may be up to 256 characters long and may not contain any spaces. If this parameter is not supplied and the prompt:NO parameter is supplied, cfrprofile will fail. A node must already exist in cfnode.cfg in order to successfully add or update a responder profile.

ex: node:dataserverA
n:dataserverB

command line option: n

prompt

The prompt parameter should be used to put cfrprofile into an interactive mode. If prompt:YES is supplied, cfrprofile will prompt the user for all information needed to create or update a responder profile. Using the prompt:YES parameter will also ask the user if they would like to create cfrprofile.cfg if it could not be found. Prompting is turned on by default. If the user does not wish to be prompted they should supply prompt:NO.

rPass

This is a required parameter. The rPass parameter is used to specify the remote password that is sent by the remote MFT Platform Server installation initiating the transfer. If this parameter is not supplied and the prompt:NO parameter is supplied, cfrprofile will fail. If this responder profile is to be in conjunction with an already verified user, rPass should be set the *VER.

ex: rPass:apple
rp:*VER

command line option: rp

rUser

This is a required parameter. The rUser parameter is used to specify the remote username that is sent by the remote MFT Platform Server installation initiating the transfer. If the remote user resides on a mainframe, then this parameter should not be longer than 8 characters. If this parameter is not supplied and the prompt:NO parameter is supplied, cfrprofile will fail.

ex: rUser:kenny
r:kenny

command line option: r

/?

The /? (or -?) parameter should be used to display the online help for cfrprofile. Online help is as follows:

```
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.
```

5.10.4.2 Examples Using cfrprofile

Below is a sample of 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        = ****************
```

Below is a sample of cfrprofile 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 updated a **cfrprofile.cfg** file with the following contents:

```
[dataServerA]
      RemoteUser=kenny
            RemotePassword= 24c89e105efee2f3d2d84988a4140652b45d7345
      LocalUser=john
         LocalPassword= 40562eb4d4fd437ab7d7b256221267b6c43da8fb8
```

A Responder Profile may be deleted from cfrprofile.cfg by using a text editor and deleting the desired Responder Profile record from the file.

The responder profiles may also be listed or deleted using the “action” parameter. A sample of how to list responder profiles is shown below:

```
cfrprofile a:list

[dataserverA]
 Local User        = john
 Remote User       = kenny
```


5.10.5 Distribution Lists

Distribution lists define multiple nodes and can hold a default directory to be used when conducting send file transfer requests. A distribution list name, the nodes to be used, and a distribution directory get defined in **cflist.cfg**. When a Distribution List is selected from the Transfer window by using the List button the destination information is pulled from your node configurations. Note: when running a receive request the List button will be grayed out.

Below are some examples that could be defined 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=Store3, Store4
Directory=c:\tmp\prod\data
Node=Store5
```

5.10.5.1 Distribution Parameters

[xxxxxxx]

This is a required parameter where xxxxxxx defines the Distribution List name which can be from 1 to 32 characters configured between 2 brackets containing no spaces. Any name longer than 32 characters will be 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 chosen to be used. Multiple nodes defined on 1 line must be delimited by a comma.

Directory

The Directory parameter is used to define the directory for the nodes to use to read or write to. If the parameter is defined without a directory then the directory defined in the transfer window or command line will be used. However, if there is a directory defined in the distribution list it will override a directory that is defined in the transfer window or on the command line.

5.11 RocketStream Accelerator

RocketStream Accelerator provides greatly improved data transfer speeds over high bandwidth/high latency IP networks. Tests have shown transfers completing up to 10 to 100 times faster than FTP, overcoming the slowness due to latency problems. We have added the RocketStream Accelerator file transfer technology to MFT Platform Server in order to provide a faster way to send files to remote destinations, where there are normally latency problems in long distance connections.

Note: You must be licensed to use the RocketStream Accelerator technology. If you are not currently licensed for RocketStream Accelerator please contact TIBCO Software Inc. See the Information Page of this document for contact information.

RocketStream Accelerator uses its own version of User Datagram Protocol (UDP) and RocketStream Accelerator's parallel implementation of TCP, called Parallel Delivery Protocol (PDP).

5.11.1 RocketStream Accelerator Ports

By default, the RocketStream Accelerator listens on port 9000 for requests coming in using TCP or UDP protocol and listens on port 9002 for requests using the PDP protocol.

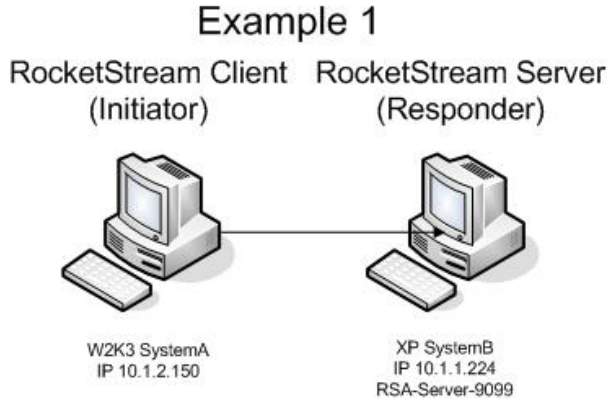
When the request has been received the RocketStream Accelerator Client will then set a port number to be used for the data transmission between the RocketStream Accelerator Server and the Responder in the port range of 9100 – 9199.

Note: Ports 9000, 9002 and 9100-9199 must be opened in the firewall to allow the RocketStream Accelerator Client to access the RocketStream 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.

5.11.2 Using RocketStream Accelerator within MFT Platform Server

Currently RocketStream Acceleration is available using MFT Platform Server for Windows. Your Windows MFT Platform Server installation can act both as a RocketStream Accelerator Client and/or a RocketStream Accelerator Server. It is possible to 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 RocketStream Accelerator service (RsTunnel.exe). Two example diagrams and configuration instructions are presented below:

5.11.2.1 Example 1 - Windows to Windows



This example depicts a file being sent from a Windows MFT Platform Server (SystemA) to a Windows MFT Platform Server (SystemB) using RocketStream Acceleration. This is the simplest RocketStream Accelerator transfer to configure.
First, verify SystemB has the RocketStream Accelerator service running and is listening on the default port 9099. To verify this, open SystemB's MFT Platform Server Administrator window and then display the Server Properties, click on the RocketStream Accelerator Tab as seen in Section 2.4.5 of this manual. Here you can check the status.

Next, on SystemA you can setup an Advanced TCP Transfer by filling in the necessary transfer detail information in the various Transfer Property tabs as seen in Section 2.2.1 through Section 2.2.8 of this manual except for when you get to the RocketStream Accelerator tab as seen below:

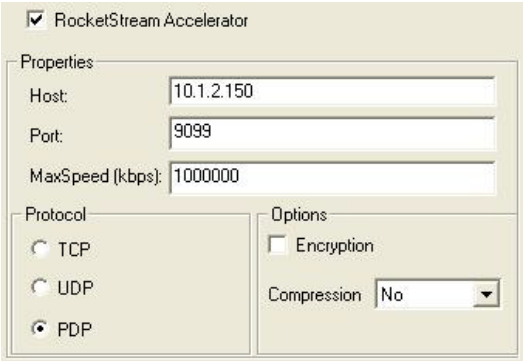
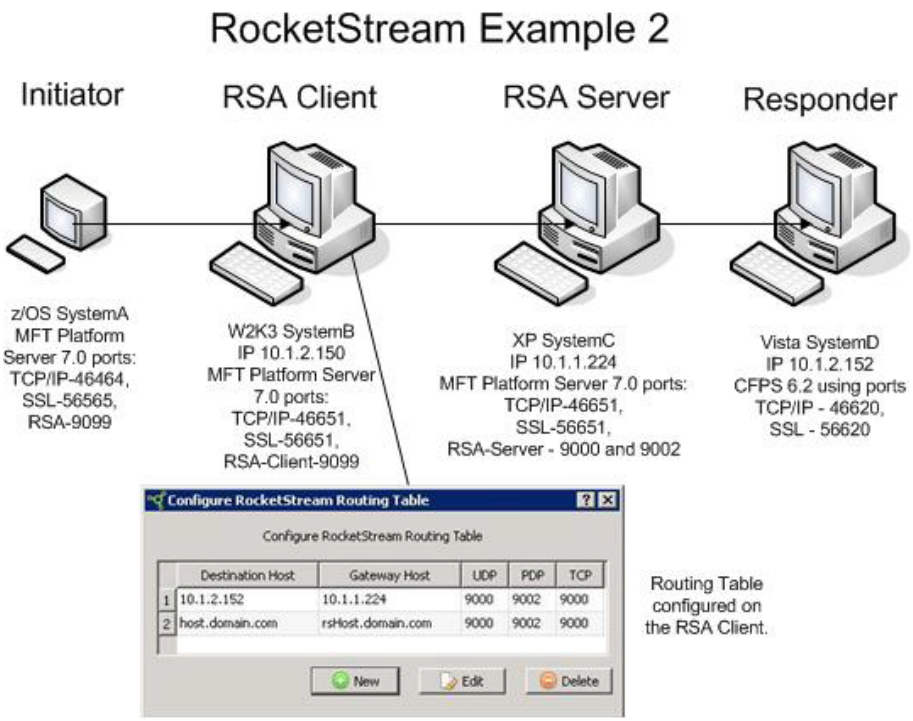


Figure 4.11.a

By default, the RocketStream Accelerator parameters will be grayed out. To have this transfer be sent via RocketStream Accelerator you must set the check mark in the RocketStream Accelerator box. Only then will you be able to configure the RocketStream Accelerator parameters. In our screenshot above we have defined our transfer to go through the local RocketStream Accelerator Client (SystemA). You can read more about the RocketStream Accelerator parameters in Section 2.2.7 of this manual. We have accepted the defaults for all other fields on this screen.

Once your transfer details are complete you would click on the OK button on the bottom of your Transfer Properties window. Your file will now be sent using RocketStream Accelerator.

5.11.2.2 Example 2 – z/OS to UNIX using RocketStream Accelerator for Windows



As you can see in Example 2, there is more going on 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 RocketStream Accelerator technology contained in them and therefore must pass the transfer to a MFT Platform Server for Windows server running the RocketStream Accelerator service.

When conducting RocketStream Accelerator transfers of this kind you must configure a RocketStream Accelerator Routing Table on the RocketStream Accelerator Server that the Platform Server Initiator connects to. This RocketStream Accelerator Server, referred to as the RocketStream Accelerator client, needs the connectivity information for the destination RocketStream Accelerator Server that will connect to the Platform Server Responder. Note if your final destination is the RocketStream Accelerator Server itself, no routing table entry is needed. It is only required when the Platform Server Responder is a different machine than the RocketStream Accelerator Server. Example 1 shows the Platform Server Responder on the same machine as the RocketStream Accelerator Server; therefore no routing table updates are needed. Example 2 shows the Platform Server Responder on a different machine as the RocketStream Accelerator Server; therefore the routing table must be updated. To configure the routing table open Windows Explorer and navigate to the following folder:

<PlatformServer_Install>\TIBCO\MFT Platform Server\RSTunnel\

Double click on the file **RSTunnelConfig.exe**. The following window will open:

Figure 4.11.b

Click on the **Configure the RocketStream Accelerator**, (Set Debug Options should only be used when instructed by TIBCO Technical Support.)

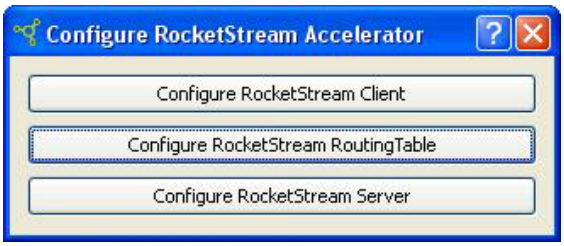


Figure 4.11.c

Click on the **Configure RocketStream Accelerator Routing Table** button. You will see an example setup.

Figure 4.11.d

Click on the first line to highlight it and then click on the  button. The following window will open:

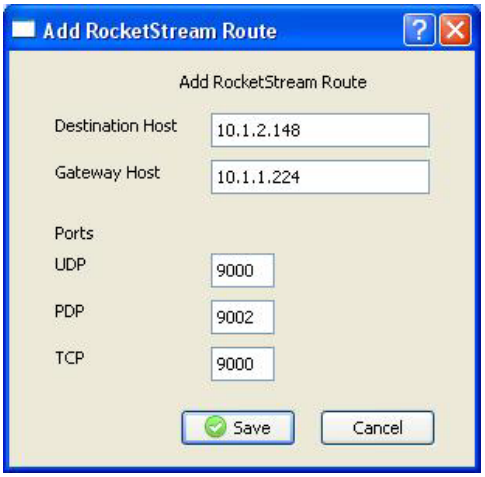


Figure 4.11.e

In this window, you define the Destination Host (the IP of the server that is the final destination for your file being transferred: the Responder.) and the Gateway Host (the remote RocketStream Accelerator Server that will be initially receiving your file transfer before passing it off to the Destination Host). You also have the opportunity to edit the default ports being used for the various protocols RocketStream Accelerator provides. As you can see from the screenshot above we have configured our Routing Table with SystemC's and SystemD's IP addresses. When you are done click on the  button.

You will next be presented with a warning that the RocketStream Accelerator service must be restarted for the changes to take place. Stop and Start RocketStream Accelerator from your Server Properties window or you can open your Services window and restart MFT Platform Server. You can close the **Configure RocketStream Accelerator Routing Table** window.

We just configured our routing table on the RocketStream Accelerator Client defining what ports should be used when sending files with the various protocols RocketStream Accelerator offers. At this time we can also define what port and IP address our Client should bind to if multiple network cards are available. By default the client listens on port 9099. If you need to change the default port number, click the **Configure RocketStream Accelerator Client** button seen in Figure 4.11.c. The following window will open:

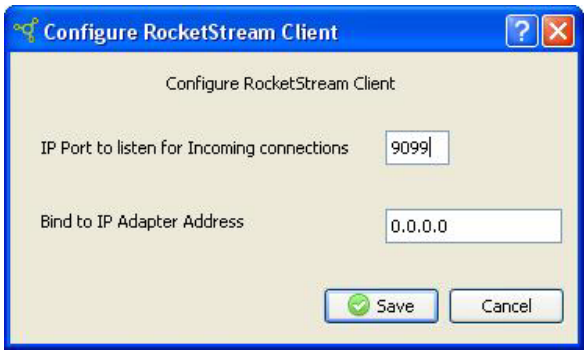


Figure 4.11.f

If your server has multiple network cards you would define the IP Adapter Address you want the RocketStream Accelerator Client to bind to, otherwise you can leave the **Bind to IP Adapter Address** field alone. When you are done click on the 

button and close the window.

The above configurations we have discussed are all you need to define for a RocketStream Accelerator Client. However, if your RocketStream Accelerator Client would ever be switching roles and acting as an RocketStream Accelerator Server in the future, you can take this opportunity to configure the RocketStream Accelerator Server ports and IP Address' to bind to at this time by clicking on the **Configure RocketStream Accelerator Server** button as seen in Figure 4.11.c. The following window will open:

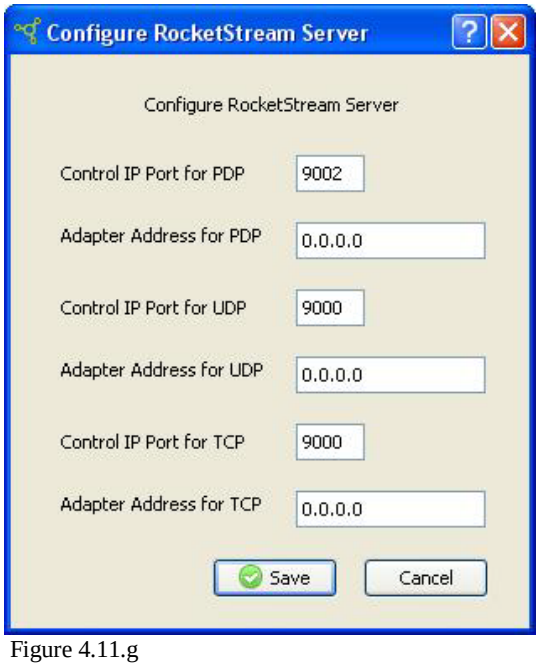


Figure 4.11.g

Unless you want to change the default ports being used by the server or you need to bind to a specific IP Address because multiple network cards are installed on the system, you would leave these settings alone.

This completes configuring our RocketStream Accelerator Client and RocketStream Accelerator Server, but what about the transfer being initiated from the z/OS platform? MFT Platform Server v7.0 for z/OS and above, as well as MFT Platform Server v7.0 for UNIX and above, has the ability to send configured transfer to a RocketStream Accelerator Host (RocketStream Accelerator Client). When configured the transfer will go to the RocketStream Accelerator Client where it will be decided what RocketStream Accelerator Server will be receiving the transfer request by referencing the routing table we configured earlier in order for the transfer request to reach its final destination. For more information on sending files from a z/OS or UNIX MFT Platform Server please refer to the MFT Platform Server documentation for that platform.

A RocketStream Accelerator Server can send a file to any MFT Platform Server Responder v7.0 and below. This includes 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. Should you need further assistance than what this section provides please contact the TIBCO Technical Support. Please refer to the Information Page at the beginning of this manual for contact information.

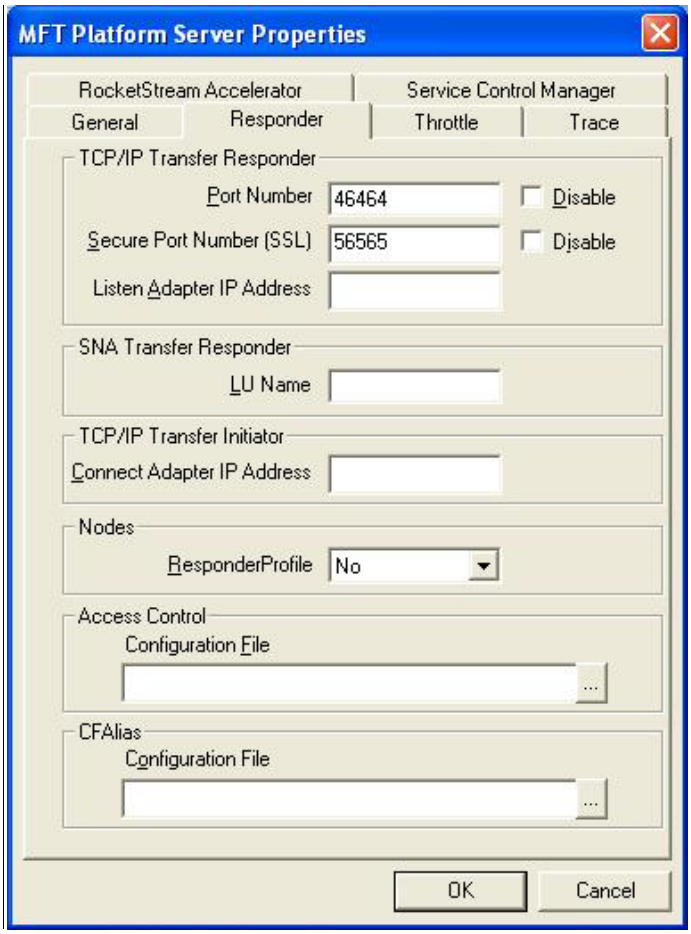
5.12 SSL

An additional layer of security may be configured for MFT 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, Platform Server includes an SSL utility, SSLutility.exe (See section SSL Utility), which generates a private key and Certificate Signing Request (CSR) file. A public key is then obtained by forwarding the CSR to a Certificate Authority (CA) for authorization. Once authorized, the CA will return a public certificate that has been signed by the CA and can be used by the MFT Platform Server. This section will describe installation, configuration and usage of SSL on MFT Platform Server.

5.12.1 SSL Installation

All SSL transactions must be performed on a port specifically identified for this purpose only. It will not be the same port as the TCP/IP port that MFT Platform Server listens on for incoming requests. The SSL Port is optional. Entering "0" as the SSL port number disables SSL.

The SSL port number can be configured at the time of installation by following the Install Shield steps. However, if it is not entered at the time of installation, you can set the port by opening the Server Properties Window and clicking on the Responder tab as seen below:



Choose a port in the range [5000-65535] that you wish to use for SSL in the **Secure Port Number** text box. The suggest port is 56565. Ensure that the **Disable** check box is not checked. Then click on the **OK** button. In order to invoke this change/addition made to the registry, it is necessary to stop and start the MFT Platform Server service.

5.12.2 SSL Utility

If you already have an SSL certificate in base64 format for the machine you have MFT Platform Server installed on, you can use it for SSL transfers. If you need an SSL certificate, you can use our SSL utility to issue a certificate request to the 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 on SSLUtility.exe.

The utility creates certificate requests and private keys, as well as allowing a user to view an existing certificate. Note: The bit strength must meet the requirements of the CA.

5.12.2.1 Creating Certificates

The following screen shot depicts the menu of choices available to the user once SSLUtility is executed.

```
SSL Utilities Menu
1. Generate a Certificate Request
2. View a Certificate
3. Exit

Please enter your choice:
```

Selecting choice "1" to create a certificate request will prompt the user 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 being made
Organization	Group or company to 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 will be required to access the private key; maximum value 20

The utility will then create a certificate request and private key and place them in the files that the user provided. These files can be forwarded to a certificate authority to request a certificate.

***Note: The name of the file and directory for the Certificate Request File and the Private Key File cannot contain any spaces or the files will not be created properly.**

5.12.2.2 Viewing a Certificate

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

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

View Certificate Menu

Please enter the Certificate Filename:
c:\MFT Platform Server\sslcert
```

When prompted to enter the certificate file name, enter the fully qualified file name.

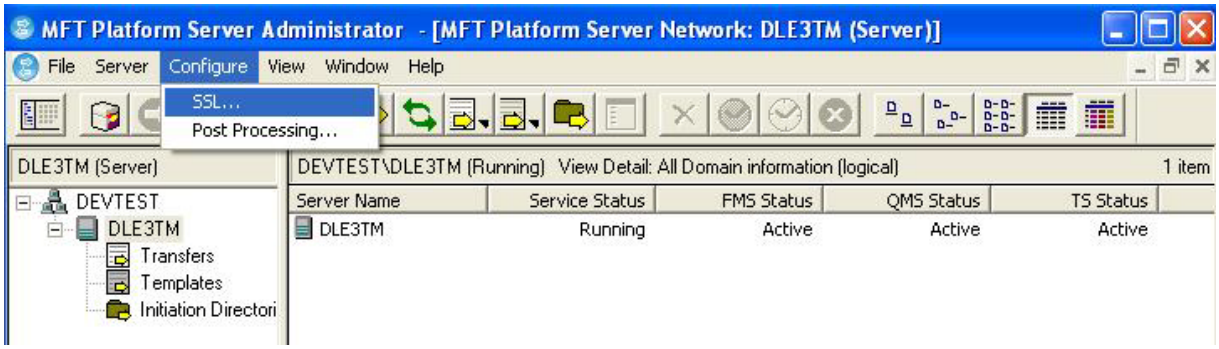
The output will look like the following:

```
Please enter the Certificate Filename:
c:\MFT Platform Server\sslcert
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 7 (0x7)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=TIBCO, OU=TIBCO Local CertAuth
        Validity
            Not Before: Aug 13 00:00:00 2005 GMT
            Not After : Aug 13 23:59:59 2006 GMT
        Subject: C=US, ST=NY, L=Garden City, O=TIBCO Software Inc.,
OU=Technical Support, CN=Joleen/Email=jbarker@tibco.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:ae:6a:25:45:19:e0:ec:d1:13:b7:a6:9c:fc:f4:
                    39:b6:a3:74:b2:98:4c:02:77:74:37:69:2f:08:f1:
                    3f:3e:95:68:1d:e8:93:09:90:8a:ec:16:8e:50:62:
                    82:57:31:8e:a5:6f:db:1c:72:79:c0:d3:de:83:e4:
                    f6:da:e1:ee:e0:d4:2f:26:05:77:f0:94:e9:70:20:
                    75:42:0d:64:eb:8f:36:a2:04:67:a9:e5:e0:ab:a3:
                    f9:a8:22:5d:75:b1:60:6e:82:ea:6f:5a:cf:61:d6:
                    2e:f7:36:b9:76:9e:4e:6d:f5
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            Netscape Comment:
                .<Generated by the SecureWay Security Server for z/OS (RACF)
            X509v3 Subject Key Identifier:
2C:C4:0E:E4:AC:E2:2D:9F:E3:EC:5F:32:67:53:B0:6A:D4:EB:36:F3
            X509v3 Authority Key Identifier:
keyid:42:77:A2:C7:AE:3D:A5:47:5C:30:FF:4F:51:B8:CF:ED:AC:D1:9C:3A
    Signature Algorithm: sha1WithRSAEncryption
        9f:7d:bd:66:f1:d5:2c:cf:5d:c5:cc:aa:16:16:e5:52:ae:04:
        89:51:66:c6:c5:03:0a:19:66:c1:d2:c9:30:4d:a4:85:c9:91:
        79:79:b0:61:bf:88:61:44:3e:21:fa:2d:98:85:b8:df:c5:77:
        ea:ee:c5:8b:7f:c3:27:56:69:3d:42:8b:c2:4a:89:2e:6f:85:
        fe:62:9c:fe:45:a0:3b:07:9b:1f:7b:f8:c0:35:89:af:be:72:
        8a:0c:a2:37:a5:fc:70:58:48:99:4f:40:ae:95:21:1e:4b:90:
        30:36
-----BEGIN CERTIFICATE-----
DXMxCzAJBgNVBAgTAm55QMowCAYDVQQHEwFnMQowCAYDVQQKEwFwMQowCAYDVQQLEwFxMQowCAYDVQQ
DEwFqMRAwDgYJKoZIhvcANQkBFgFqMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCuaiVFGeDs0R
O3ppz89Dm2o3SymEwCd3Q3aS8I8T8+lWgd6JMJkIrsFo5QYoJBGep5eCro/moIl11sWBugupvWs9h1i
73Nrl2nk5t9QIDAQABo4GQMIGNMEsGCWCGSAGG+
9yIE9TLzM5MCAoUkFDRikwHQYDVR0OBBYEFCzEDuSs4i2f4+xfMmdTsGrU6zbzMB8GA1UdIwQYMBaAF
EJ3oseuPaVHXDD/T1G4z+2s0Zw6MA0GCSqGSIb3DQEBBQUAA4GBAJ99vWbxXSzPXcXMqhYW5VKuBIlR
ZsbFAwoZZsHSyTBNpIXJkXl5sGH7iGFEPiH60piFuN/Fd+ruxoMojel/HCFSJlPQK6VIR5LkDA2
-----END CERTIFICATE-----

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

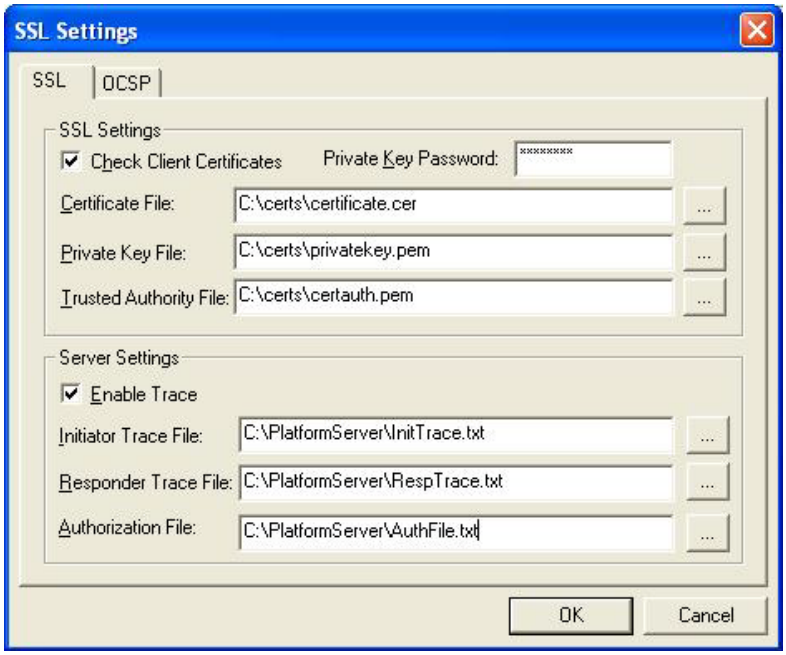
5.12.3 SSL Configuration

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



This will open the SSL Settings panel.

5.12.3.1 SSL tab settings



This panel is divided into two sections: SSL Settings and Server Settings. The SSL Settings are required for all SSL transfers. The Server Settings are optional and are used only when additional tracing or certificate authorization is required.

- **SSL Settings**

  Check Client Certificates
  In the upper left corner of the SSL Settings section, click the **Check Client Certificates** box if you would like to perform Client Authentication in addition to Server Authentication. If this box is not checked, only Server Authentication will be performed. Checking the Check Client Certificates box also enables the **Authorization File** text box in the MFT Platform Server Settings section of this panel. An authorization file may be entered for additional security if **Check Client Certificates** is selected.

Private Key Password
To the right of **Check Client Certificate**, there is a text box for the **Private Key Password**. The password/passphrase must be entered in order for MFT Platform Server to access the private key file for data encryption/decryption. Asterisks will appear in the box as the password is entered to ensure the security of the private key file.

Certificate File
In the **Certificate File** text box, enter the drive, path and file name of the Base 64 encoded certificate to be used by the MFT Platform Server. This certificate will be presented when MFTPS 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 Base 64 encoded private key to be used when MFT Platform Server is decrypting data that is received. 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 Base 64 encoded file containing the trusted authority certificates of the CA that 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.

- **MFT Platform Server Settings**

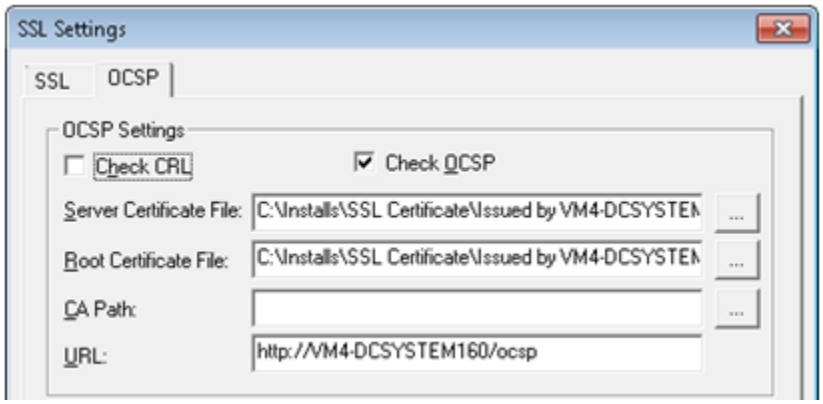
Enable Trace
In the upper left corner of the MFT Platform Server Settings section, click the check box to enable tracing. This will enable other fields in this section. Although SSL tracing is optional, once selected the Initiator Trace File and Responder Trace File fields are required. Tracing should only be turned on at the request of TIBCO Technical Support.

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
In order to enter an authorization file, check the **Check Client Certificates** box in the SSL 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 allows the user to exclude and include certificates based on components of the distinguished name (i.e. user name, company, division, serial number, etc.) as well as by date and time. This is an optional component of SSL transfers, and can only be implemented if client authentication is being performed (i.e. the **Check Client Certificates** box is checked.)

5.12.3.2 OCSP (Online Certificate Status Protocol) tab settings



The Online Certificate Status Protocol (OCSP) is a protocol used to communicate the status of certificates. OCSP offers an advantage over CRLs by querying a service for the status of a particular certificate as opposed to a Certificate Revocation List which may be infrequently updated. Once configured, the Platform Server will query the defined OCSP Responder for the status of both the local and remote certificates used in the transfer.

To enable this optional feature, check the checkbox for **Check OCSP**.

- **OCSP Settings**

Check CRL
Enable the **Check CRL** box if you would like to have the Platform Server check the published Check Revocation List.

Check OCSP
Enable the **Check OCPS** box if you would like to have the Platform Server check the published Check Revocation List that can be found by going to the URL defined in the **URL** field.

Server Certificate File
In the **Server Certificate File** text box, enter the drive, path and file name of the Base 64 encoded (PEM) certificate for the OCSP Responder. A browse button is provided to the right of the text box to facilitate this process.

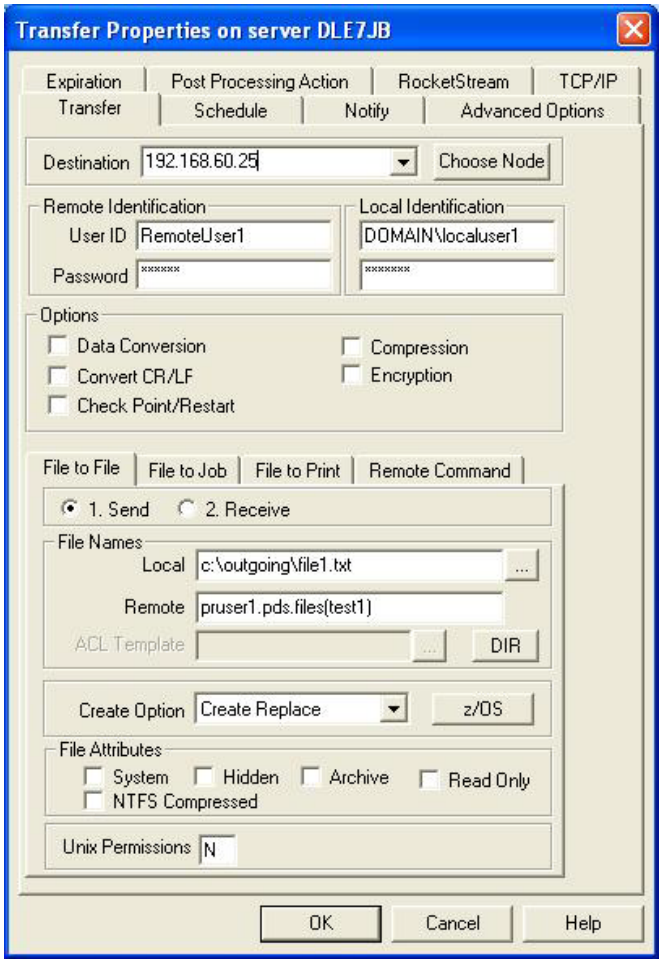
Root Certificate File
In the **Root Certificate File** text box, enter the drive, path and file name of the Base 64 encoded (PEM) certificate of the signing CA of the OCSP Responder. A browse button is provided to the right of the text box to facilitate this process.

CA Path
In the **CA Path** text box, enter the drive, path and file name to allow the Platform Server to trust certificates issued from this certificate authority. A browse button is provided to the right of the text box to facilitate this process.

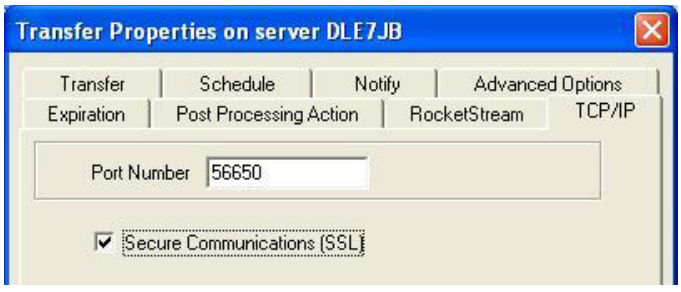
URL
If you have selected to **Check OSCP** you must specify in the **URL** text box, the path to the web proxy service that handles OCSP requests.

5.12.4 SSL Transfer

1) Open your Advanced TCP/IP transfer window.
2) Set up your transfer. See example below:



3) When you have completed your Transfer setup go to the TCP/IP tab.
4) Configure the remote MFT Platform Server servers SSL port being used and enable SSL:



5) Once the transfer configurations are complete click on the **OK** button and the transfer request will run.

5.12.5 SSL Authorization Parameters

MFT Platform Server supports an extension to the standard SSL processing to allow the system administrator to determine which certificates should be accepted and which should be rejected. This is done by the creation of an SSLAUTH file. This is supported on all MFT Platform Servers. The format of the file is the same on all platforms, but the way that the file is defined is dependent on each platform.

The table below gives the name of the SSL authorization file on each platform.

Platform	Default Location	File Name
z/OS	SAMPLIB	SSLAUTH
Windows	C:\Program Files\TIBCO\MFT Platform Server	SslAuth
UNIX	/PlatformServer/samples	SSLAUTH

Note that the authorization file checking is above and beyond the authorization checking performed by SSL. Only if a certificate is accepted by SSL will the authorization file processing be performed.

The authorization file is compared against the Certificate that was received by the MFT Platform Server. The authorization file is not used on the client. The components of the Certificate's Distinguished Name (DN) are compared to the parameter in the authorization file to determine if a certificate should be accepted or rejected. On many of the parameters, a generic character is supported. A generic character is defined in a parameter by an *. 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 will be accepted. If you want to reject all requests unless defined by the authorization file, then you should insert the following statement as the last entry in the authorization file:

REVOKE

There are two request types supported within the authorization file:

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, then 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 will be continued on the next record. All parameter data is case sensitive. Be very careful when entering the values when using mixed case fields.

Parameters allowed in the authorization file. These parameters must be defined in Upper Case.

/CN Define 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: ccyymmdd. Generic entries are NOT supported. The start date is compared against the date that the transfer request is received by MFT Platform Server. If the start date is before the current date, then SSLAUTH processing will check the next parameter. If the start date is after the current date, then the transfer request will

be terminated and an error will be sent to the remote system.

/STIME Defines the Start time for the certificate in the format: hhmm. Generic entries are NOT supported. The start time is only checked if the SDATE parameter exactly matches the current date. The start time is compared against the time that the transfer request is received by MFT Platform Server. If the start time is before the current time, then SSLAUTH processing will check the next parameter. If the start time is after the current time, then the transfer request will be terminated and an error will be sent to the remote system.

/EDATE Defines the End date for the certificate in the format: ccyymmdd. Generic entries are NOT supported. The end date is compared against the date that the transfer request is received by MFT Platform Server. If the end date is after the current date, then SSLAUTH processing will check the next parameter. If the end date is before the current date, then the transfer request will be terminated and an error will be sent to the remote system.

/ETIME Defines the End time for the certificate in the format: hhmm. Generic entries are NOT supported. The end time is only checked if the EDATE parameter exactly matches the current date. The end time is compared against the time that the transfer request is received by MFT Platform Server. If the end time is after the current time, then SSLAUTH processing will check the next parameter. If the end time is before the current time, then the transfer request will be terminated and an error will be sent to the remote system.

/USER This parameter is supported only by the z/OS system. It allows the administrator to define a userid that should be used when an SSL certificate is accepted. This userid overrides the userid associated with the file transfer. Using this option, the remote user does not have to have any knowledge of a userid/password on the z/OS system.

Examples of authorization file processing:

Accept /OU=Marketing/O=TIBCO
revoke
MFT Platform Server will accept all certificates defined with an Organization of TIBCO and an Organization Unit of Marketing. It will reject all other certificates.

REVOKE /SN=987654
REVOKE /SN=12:34:56
ACCEPT
MFT Platform Server will reject any certificates with a serial number of 987654 or 123456. It will accept all other certificates.

Accept /OU=ACCT*/O=ACME
revoke
MFT Platform Server will accept all certificates defined with an Organization of ACME and an Organization Unit starting with ACCT. It will reject all other certificates.

Accept /CN=Joe*,
/L=New York,
/ST=NY,
/C=US,
/OU=Dept1,
/O=ACME,
/SDATE=20051201,
/EDATE=20061130
revoke

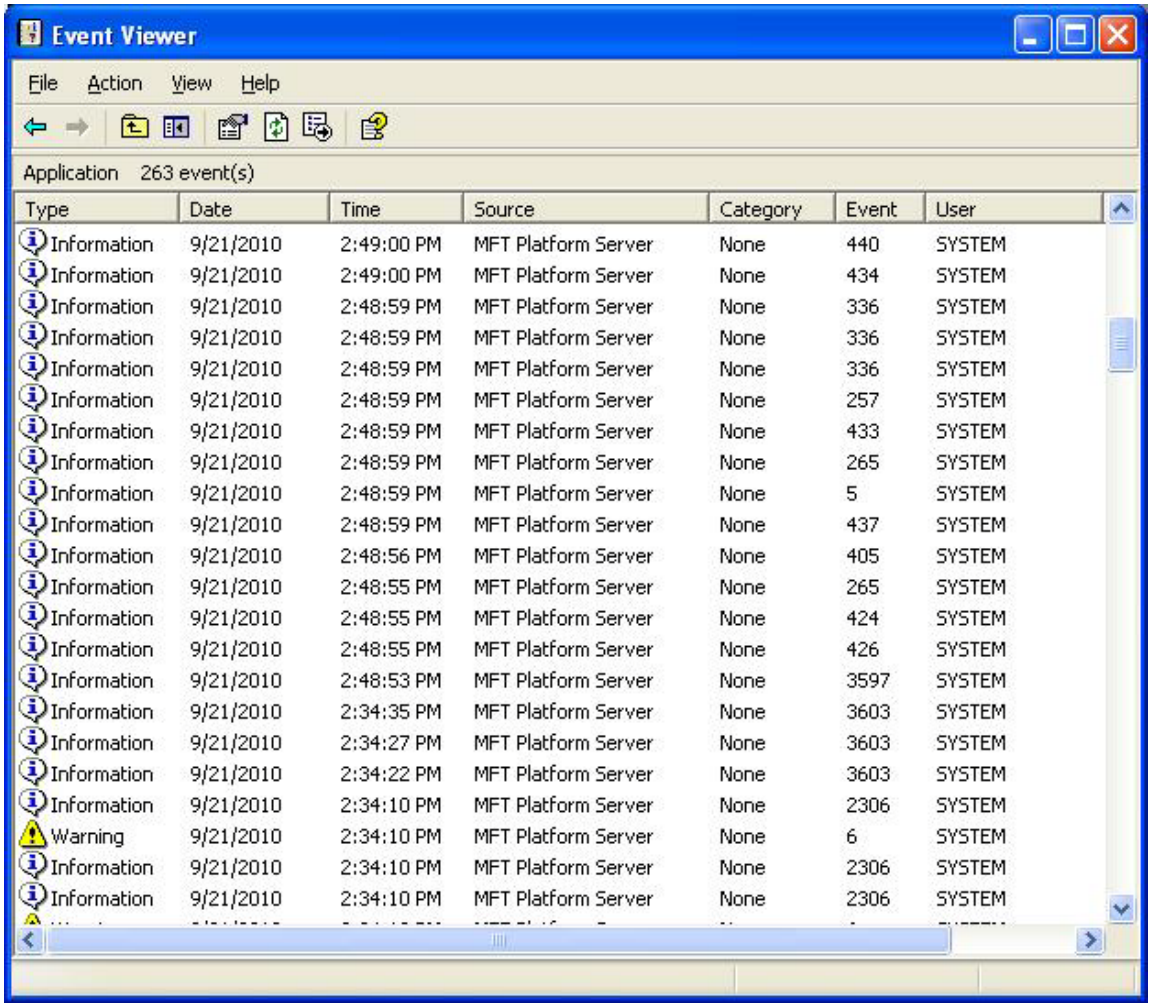
MFT Platform Server will accept all certificates that match the information defined by the /CN, /L, /ST, /C, /OU and /O parameters. The certificate is valid from December 1, 2005 until November 30, 2006. If the certificate is received before December 1, 2005 or after November 30, 2006, the request will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

Appendix A

Appendix A. The Event Log

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

The Event Viewer is located in the Administrative Tools panel in Program Manager. To view the log, double-click the Event Viewer icon. The following is 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

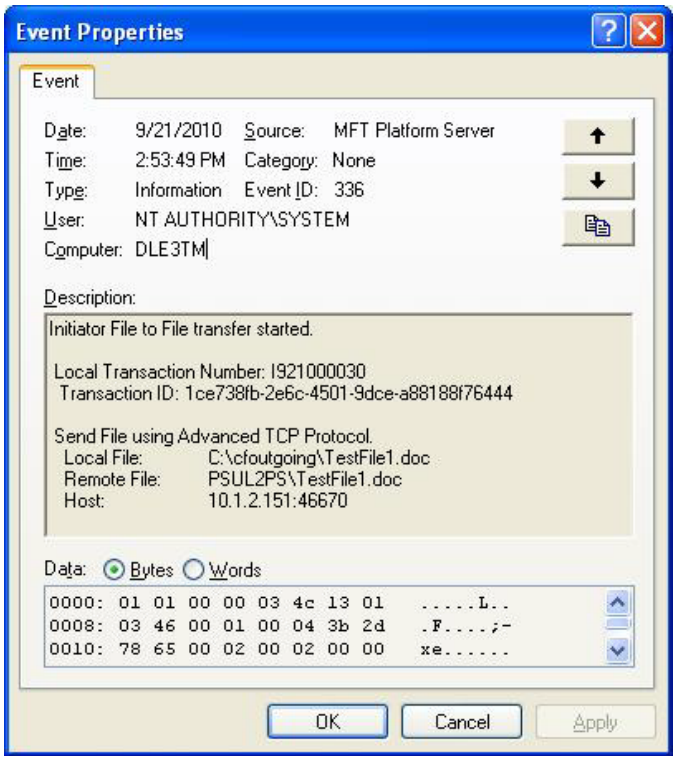
A.1 Using the Event Log

You can view three types of logs: application, system, and security. To select the log to view, select a log type from the Log pull-down menu.

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

MFT Platform Server writes the event to the event log in both successful and unsuccessful cases. MFT Platform Server also writes an informational 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 will be 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 Detail panel will appear.



A.1.1 Event IDs and Transaction IDs

When MFT Platform Server writes to the Windows Event Log, it provides an Event ID.

MFT Platform Server also writes transaction IDs for each of the transfers in the Windows Event log. The transaction IDs are broken down into two categories: local and remote. A transaction is assigned to one of these two categories by the MFT Platform Server initiator at the earliest possible time during the transfer. This transaction ID assigned is unique for all machines.

If a transaction is displayed in the event log before it has been issued a transaction ID, the transaction will not have an ID number in the Event Log. For example, a transaction ID will not be assigned if the failure occurs before a connection to the remote system is established. The transaction would not have been assigned an ID by the remote system because it never actually got to the remote system.

In addition to the transfer ID there are three additional types of information provided on the panel; message specific, error severity and retry information.

Message specific information provides the user with the details of the particular transfer ID that they are viewing at that time. This information includes remote file name, local file name, the transfer direction etc. Below this message information is information on the severity of the error. If the transfer failed with a severe error, this will be indicated in the message. If the error is anything other than a severe error, MFT Platform Server will retry the transfer if the Try Count has been set to a value greater than one. If MFT Platform Server retries the transfer more than once, the retry information will state the number of times that the transfer was attempted before it completed successfully or failed.

A.1.2 Severity 1 Errors

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

The following errors are classified as Severity 1.

1. Could not open the source file.
    - the name is incorrectly formatted
    - the volume name is incorrectly formatted
    - the path is non-existent
2. Could not open the destination ACL Template.
    - the name is incorrectly formatted
    - the volume name is incorrectly formatted
    - the path is non-existent
3. The destination printer name is invalid.
4. Logon failure.
5. File compression failed.
    - not an NTFS formatted drive
6. Destination incorrect.
    - the LU name or IP Address is incorrect
    - entered an LU name for a TCP/IP transfer
    - entered and IP address for an SNA transfer

A.2 Clearing the Event Log

When you receive a message that indicates that the event log is full, you must clear the log. You can use one of two methods:

To clear the event log:

Method 1. Empty the current log.

1. Switch to the log whose events you would like to clear.
2. From the Log menu, choose **Clear All Events**.
   You will be given the option to save the currently logged events.
   - If you choose to archive the events, you must select a filename and choose the directory path on which you want to store the log.
   - If you choose to not save the events, the Event Viewer empties the current log.

Method 2. Each new event replaces the oldest event.

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

Appendix B

Appendix B. 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 then perform MFT Platform Server transfers on Windows without having to specify the password. You can easily manage the cached password from the remote side as needed.

To enable this 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:

1. X: password
2. X:
3. X:DELETE
4. X:DELETEALL.

The tokens are case-sensitive. For example, *x:password* (note the lowercase x) will be interpreted as the user's password and not as the token (with the uppercase x) to set the cached password.

1. X:*password*

Use this token to set the 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 p*assword* 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, the transfer will execute.

2. Given X: without a password

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

3. X:DELETE

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

4. X:DELETEALL

Use this token to instruct MFT Platform Server to retrieve the cached password saved from a prior transaction for your user ID, decrypt it, and log in to Windows to conduct a transaction and then delete all the cached password from the registry.

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

Note: The cached password feature is only supported on Windows. It is not supported on the z/OS side. If you send over X:*password*, z/OS will interpret the full string as the password.

Restrictions:

1. The service must be running with System Authority.
2. 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.
3. Passwords that could otherwise contain X:, X:*text*…, X:DELETE, or X:DELETEALL will be accepted as triggers to the feature and not as legitimate Windows passwords.
4. Since the passwords are saved in a restricted area of the registry, the uninstall program cannot delete them. You must use *X:DELETEALL* to remove the passwords before using uninstall. If you do not, \\HKEY_LOCAL_MACHINE\SOFTWARE\TIBCO registry key will not be removed.

Example:

User MARY wants to create a batch transfer to a Windows remote system. However, Mary doesn't want everyone to know her password.

Mary should use the X:*password* token to set the cached password. The following batch program will invoke 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
```

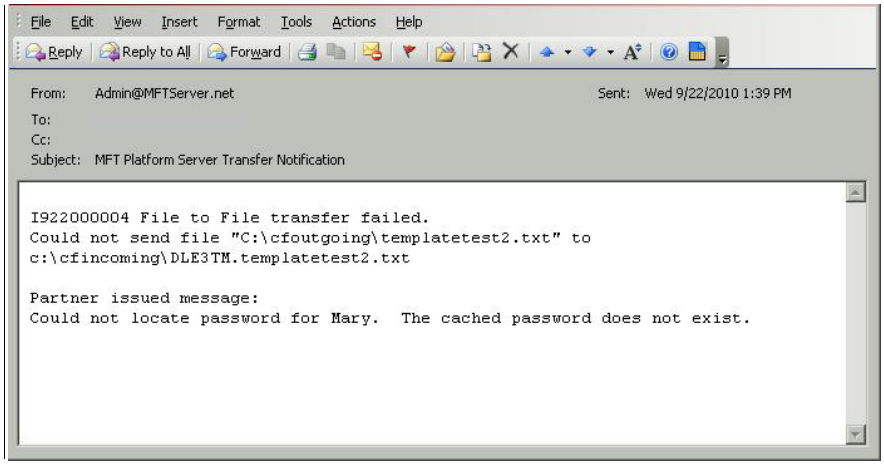
For all future transfers, Mary does not need to specify a password. Instead, she can use the X: token.

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

She can use the following batch program for future transfers.

Note: The password field and the tokens are case-sensitive. If the password was lowercase, then Mary should type "X:pswdmary".

If the password is not yet cached, you may also receive the following:



Appendix C

Appendix C. File Name Tokens

File Name Tokens are a feature of the MFT Platform Server. Given a string of tokens—characters containing a mixture of literal and substitution parameters—the MFT Platform Server generates a formatted file name that you can use to create or read file names based upon 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.

- Section C.1 lists the available tokens.
- Section C.2 provides examples that demonstrate how to use File Name Tokens.
- Section C.3 lists the rules that you must follow when using the File Name Tokens.

C.1 File Name Tokens—List

The following table lists the File Name Tokens, their respective definitions, and their generated values.

Token	Definition	Generated Value (Examples)
SYYYY	Year	0000–9999
SYYY	Year	000–999 (last 3 digits of year)
SYY	Year	00–99 (last 2 digits of year)
SY	Year	0–9 (last 1 digit of year)
SMON	Month of Year	JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
SMon	Month of Year	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
Smon	Month of Year	jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec
SMONTH	Month of Year	JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER
SMonth	Month of Year	January, February, March, April, May, June, July, August, September, October, November, December
Smonth	Month of Year	january, february, march, april, may, june, july, august, september, october, november, december
SMM	Month of Year	01–12
SM	Month of Year	1–C
Sm	Month of Year	1–c
SDD	Day of Month	01–31
SD	Day of Month	1–9, A–V
Sd	Day of Month	1–9, a–v
SJ	Julian Day of Year	001–366
SHH24	24 Hour	00–23
SH24	24 Hour	0–9, A–N
Sh24	24 Hour	0–9, a–n
SHH12	12 Hour	01–12
SH12	12 Hour	1–C
Sh12	12 Hour	1–c
SMI	Minute of Hour	00–59
SSS	Second of Minute	00–59
SMS	Milliseconds of Second	000–999
SAP	AM/PM	AM, PM
SAp	AM/PM	Am, Pm
Sap	AM/PM	am, pm
SWWW	Weekday	SUN, MON, TUE, WED, THU, FRI, SAT
SWww	Weekday	Sun, Mon, Tue, Wed, Thu, Fri, Sat
Swww	Weekday	sun, mon, tue, wed, thu, fri, sat
SWEEKDAY	Weekday	SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY
SWeekday	Weekday	Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
SW1	Weekday 1 based	1–7
SW0	Weekday 0 based	0–6

Token	Definition	Generated Value (Examples)
AllocationPrimary	Primary Allocation size used in the file transfer.	Local File: c:\source\testfile1.txt Remote File: CFUSR.F$(AllocationPrimary).TEST Token resolved to: CFUSR.F800.TEST
AllocationSecondary	Secondary Allocation size used in the file transfer.	Local File: c:\source\testfile1.txt Remote File: CFUSR.F$(AllocationSecondary).TEST Token resolved to: CFUSR.F500.TEST
AllocationType	Allocation type used in the file transfer.	Resolves to: Tracks, Blocks, Cylinders, Megabytes, Kilobytes
BlockSize	Block size used in file the transfer.	Local File: c:\source\testfile1.txt Remote File: CFUSR.F$(BlockSize).TEST Token resolved to: CFUSR.F6,160.TEST
CheckPointInterval	Check point used in the file transfer.	Local File: c:\source\testfile1.txt Remote File: d:\target\test$(CheckPointInterval).txt Token resolved to: d:\target\test5 minutes.txt
Compression	Compression used in the file transfer.	LZ, RLE, or NO
ComputerName	Initiator Computer name.	Local File: c:\source\testfile1.txt Remote File: d:\target\$(ComputerName).txt Token resolved to: d:\target\SYSTEM3032.txt
CrLf	If CRLF was used in the file transfer.	TRUE, FALSE
DataClass	Data Class used in file transfer to z/OS.	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.$(DataClass).FILE1 Token resolves to: PRJOE.DTCLS3.FILE1
DataType	Data Type used in the file transfer.	BINARY, EBCDIC
Date1	The days date formatted as YYYYMMDD.	Local File: c:\source\test.txt Remote File: d:\target\$(Date1)\test.txt Token resolved to: d:\target\20110809\test.txt
Date2	The days date formatted as MMDDYYYY.	Local File: c:\source\test.txt Remote File: d:\target\$(Date2)\test.txt Token resolved to: d:\target\08092011\test.txt
Date3	The days date formatted as DDMMYYYY.	Local File: c:\source\test.txt Remote File: d:\target\$(Date3)\test.txt Token resolved to: d:\target\09082011\test.txt
Destination	IP or Hostname of the final destination for the file being transferred.	Local File: c:\source\testfile1.txt Remote File: d:\target\file1.$(Destination).txt Token resolved to: d:\target\file1.192.168.10.1.txt
FileAvailability	File Availability used in file transfer.	IMMEDIATE, DEFERRED
LocalDomain	Local Domain	Remote file name contains the local domain name in it.
LocalFile	Complete local file path.	Local File: c:\source\testfile1.txt Remote File: $(LocalFile) Token resolved to: c:\source\testfile1.txt
LocalFileBase	The local file name only.	Local file: c:\source\directory\testfile1.txt Remote File: $(LocalFileBase) Token resolves to: testfile1 (File transferred to the MFT Platform Server Windows Directory

Token	Definition	Generated Value (Examples)
		unless a path is configured.)
LocalFileExt	Only the extension of the local file is used.	Local file: c:\source\directory\testfile1.txt Remote File: $(LocalFileExt) Token resolves to: txt (File transferred to the MFT Platform Server Windows Directory unless a path is configured.)
LocalFileBase	The local file name only.	Local file: c:\source\directory\testfile1.txt Remote File: $(LocalFileBase) 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 Remote File: $(LocalFileExt) Token resolves to: txt (File transferred to the MFT Platform Server Windows Directory unless a path is configured.)
LocalFileName	The local file name including the extension will be used.	Local file: c:\source\directory\testfile1.txt Remote File: $(LocalFileName) 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 will be used.	Local file: c:\source\directory\testfile1.txt Remote File: $(LocalFilePath) Token resolves to: c:\source\directory
LocalPathWODrive	Local file path without the drive letter or file name used.	Local file: c:\source\directory\testfile1.txt Remote File: $(LocalPathWODrive) Token resolves to: source\directory (File transferred to the MFT Platform Server Windows Directory unless a drive letter is configured.)
LocalUserId	Local User Id used in the file transfer.	Local User Id: TESTLAB\cfuser1 Local file: c:\source\directory\testfile1.log Remote File: d:\target\file1$(LocalUserId).txt Token resolves to: d:\target\file1cfuser1.txt
LuName	LU Name used when transferring files using SNA protocol.	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.$(LuName).FILE1 Token resolves to: PRJOE.SPAPPL.FILE1
MgmtClass	The management class to be used when transferring to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.$(MgmtClass).FILE1 Token resolves to: PRJOE.MGCLS12.FILE1
ModeName	Mode Name used when transferring files using SNA protocol.	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.$(ModeName).FILE1 Token resolves to: PRJOE.FUSN8K.FILE1
NoLocalFileBase	The base name of the local file is not used in the file name on a send.	Local file: c:\source\directory\a.b.c.txt Remote File: c:\target\$(NoLocalFileBase) Token resolves to: b.c.txt
NoLocalFileExt	The extension name of the local file is not used in the file name on a send.	Local file: c:\source\directory\a.b.c.txt Remote File: c:\target\$(NoLocalFileExt) Token resolves to: a.b.c
NoRemoteFileBase	The base name of the	Local file: c:\target\$(NoRemoteFileBase)

Token	Definition	Generated Value (Examples)
	remote file is not used in the file name on a receive.	Remote File: c:\source\directory\a.b.c.txt Token resolves to: b.c.txt
NoRemoteFileExt	The extension name of the remote file is not used in the file name on a receive.	Local file: c:\target\$(NoRemoteFileExt) Remote File: c:\source\directory\a.b.c.txt Token resolves to: b.c.txt
NotifyUser	The **remote** user name configured to be notified in the file transfer.	Local file: c:\source\directory\testfile1.txt Remote File d:\target\file1$(NotifyUser).txt Token resolves to: d:\target\file1JohnD.txt
NotifyUserType	The type of notification used for the **remote** user notification in the file transfer.	Local file: c:\source\directory\testfile1.txt Remote File d:\target\file1$(NotifyUserType).txt Token resolves to: d:\target\file1Windows.txt (Windows, None, TSO, ROSCOE, Email)
PortNumber	The port number used in the file transfer.	Local file: c:\source\directory\testfile1.txt Remote File d:\target\file1$(PortNumber).txt Token resolves to: d:\target\file146,464.txt
PrinterName	The printer name used in File to Print.	<text>
Priority	The priority set in the file transfer.	Local file: c:\source\directory\testfile1.txt Remote File d:\target\file1$(Priority).txt Token resolves to: d:\target\file1Normal.txt
ProcessName	The Process Name configured in the file transfer.	Local file: c:\source\directory\testfile1.txt Remote File d:\target\file1$(ProcessName).txt Token resolves to: d:\target\file1CyberFus.txt
RecordFormat	Record Format used in the file transfer.	FIXED, BLOCKED, FIXED BLOCKED, VARIABLE, VARIABLE BLOCKED, UNDEFINED
RecordLength	Record length used in the file transfer.	Local File: c:\source\testfile1.txt Remote File: CFUSR.F$(RecordLength).TEST Token resolved to: CFUSR.F80.TEST
RemoteDomain	Remote Domain used in the file transfer.	Remote file name contains the remote domain name within it.
RemoteFile (Token used when doing a receive)	Complete remote file path.	Local File: $(RemoteFile) Remote File: c:\source\testfile1.txt Token resolved to: c:\source\testfile1.txt
RemoteFileBase (Token used when doing a receive)	The remote file name only.	Local file: $(RemoteFileBase) Remote File: c:\source\directory\testfile1.txt Token resolves to: testfile1 (File transferred to the MFT Platform Server Windows Directory unless a path is configured.)
RemoteFileExt (Token used when doing a receive)	Only the extension of the remote file is used.	Local file: $(RemoteFileExt) Remote File: c:\source\directory\testfile1.txt Token resolves to: txt (File transferred to the MFT Platform Server Windows Directory unless a path is configured.)
RemoteFileName (Token used when doing a receive)	The remote file name including the extension will be used.	Local file: $(RemoteFileName) Remote File: c:\source\directory\testfile1.txt 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 when doing a receive)	The remote file path without the file name will be used.	Local file: $(RemoteFilePath) Remote File: c:\source\directory\testfile1.txt Token resolves to: c:\source\directory
RemotePathWODrive (Token used when doing a receive)	Remote file path without the drive letter or file name used.	Local file: $(RemotePathWODrive) Remote File: c:\source\directory\testfile1.txt Token resolves to: source\directory (File transferred to the MFT Platform Server Windows Directory unless a drive letter is configured.)
RemoteTransactionNumber	Remote Transaction Number used in the file transfer.	Local file: d:\fn\$(RemoteTransactionNumber).txt Remote File: c:\source\directory\testfile1.txt Token resolves to: d:\fn\
RemoteUserId	Remote User Id used in the file transfer.	Remote User Id: TEST\cfuser1 Local file: c:\fn\file1.$(RemoteUserId).txt Remote File: c:\source\directory\testfile.txt Token resolves to: c:\fn\file1.cfuser1.txt
StorageClass	Storage Class used when doing a file transfer to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.$(StorageClass).FILE1 Token resolves to: PRJOE.STANDARD.FILE1
SysoutClass	The SYSOUT class used when doing a File to Print to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.$(SysoutClass).FILE1 Token resolves to: PRJOE.A.FILE1
SysoutCopies	The amount of SYSOUT copies used when doing a File to Print to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.TS$(SysoutCopies).FILE1 Token resolves to: PRJOE.TS2.FILE1
SysoutDestination	The SYSOUT destination used when doing a File to Print to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote File: HST.$(SysoutDestination).FILE1 Token resolves to: HST.NYPRINTER.FILE1
SysoutFcb	The SYSOUT FCB used when doing a File to Print to a z/OS system	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.$(SysoutFcb).FILE1 Token resolves to: PRJOE.STD2.FILE1
SysoutForms	The SYSOUT forms used when doing a File to Print to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.$(SysoutForms).FILE1 Token resolves to: PRJOE.INVC.FILE1
SysoutUserId	The SYSOUT User Name used when doing a File to Print to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.$(SysoutUserId).FILE1 Token resolves to: PRJOE.MVSUSER1.FILE1
SysoutWriter	The SYSOUT Writer used when doing a File to Print to a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.$(SysoutWriter).FILE1 Token resolves to: PRJOE.WRITER1.FILE1
TransactionNumber	Local Transaction Number used in the file transfer	Local file: c:\source\directory\testfile1.txt Remote File d:\target\f$(TransactionNumber).txt Token resolves to: d:\target\f331600053.txt
TransferFunction	The transfer function used in the file transfer.	SEND, RECEIVE

Token	Definition	Generated Value (Examples)
TransferId	The transfer Id assigned to the file transfer.	Local file: c:\source\directory\testfile1.txt Remote File d:\target\file1.$(TransferId).txt Token resolves to: d:\target\file1.d1544fd2-5fb7-4ce6-a717-ac8907697e4f.txt
TransferWork	The type of transfer being done. Ex. File to File, File to Job, etc.	F-FILE, J-JOB, P-PRINT
TryCount	Try Count used in transfer.	Local file: c:\source\directory\testfile1.txt Remote File d:\target\file1$(TryCount).txt Token resolves to: d:\target\file13 Times.txt
Unit	Unit used for transfer to and from a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.$(Unit).FILE1 Token resolves to: PRJOE.SYSDA.FILE1
UserData	The User Data name used in the file transfer.	Local file: c:\source\directory\testfile1.txt Remote File d:\target\file1$(UserData).txt Token resolves to: d:\target\file1MyUserData.txt
VolSer	Volume used for transfer to and from a z/OS system.	Local file: c:\source\directory\testfile1.txt Remote File: PRJOE.$(VolSer).FILE1 Token resolves to: PRJOE.CFP101.FILE1
WriteMode	The Write Mode used in the file transfer.	C, R, A, CR, CA, CN

C.2 Using the File Name Tokens

When transferring a file, type the file's name using File Name Tokens instead of a regular file name. Consider the following examples.

Examples

These examples use the following sample system date/time:

Wednesday, April 25, 1996 5:03:45.061 PM

- Here, instead of entering a standard file name, the user has entered a string of File Name Tokens. The MFT Platform Server or Responder will resolve the string into the directory name and file name.

  File Name: C:\directory\$(SDD)$(SMON)$(SYYYY)\$(SHH24)(SMI)(SSS).dat
  Resolved File Name: C:\directory\25APR1996\170345.dat

- In this example, the user has used the File Name Tokens to generate a resolved file name that has dashes between the date and time fields.

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

- Here, the MFT Platform Server or Responder will resolve the tokens in the File Name into a long file name using upper- and lowercase letters.

  File Name: \\Server\Volume\$(SMonth)\projectX\$(SWeekday)\products.xls
  Resolved Name: \\Server\Volume\April\projectX\Wednesday\products.xls

- Here, the template is used to create a DOS 8.3 formatted file name whose 3-character extension contains an encoded representation of the date. The number of the day of the week is also used as part of the file name. In this case, the 0-based version is used. 1 based day of week is also provided.

  File Name: C:\DOS\SHORTNM$(SW0).$(SM)(SD)(SY)
  Resolved Name: C:\DOS\SHORTNM4.4P6

- Here, the user has used 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.

- Here, the user is using DNI *and* File Name Tokens. She places *sample.txt* in her DNI directory and uses 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)

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

Note: The various time tokens available are resolved at the beginning of a file transfer from the Initiating Platform Server. As a result, if a file transfer should fail and go into retries the initial file name that was set will not change even though the transfer may be done at a later time due to retries.

C.3 Rules for Use

When you create a file name that uses File Name Tokens, you must follow these rules:

- Substitution parameters are enclosed in $(…) . A dollar sign, $, followed by and open parenthesis, (, followed by the token, followed by a close parenthesis.
- Each $(…) may contain one token only.
- Any text in the remote file name which is not a substitution parameter is embedded as is into the generated name.
- Codes may appear anywhere within the remote file name (as the file name or directory name, share name, or server name).
- Space permitting, there may be any number of substitution parameters embedded within the file name.
- If the resolved remote file name length is greater than the maximum file name allowed by MFT Platform Server on Windows (255 characters), it will be truncated.
- If the transfer type is initiator send, the remote file name will resolve to the destination file for the transfer.
- If the transfer type is initiator receive, the remote file name will resolve to the source file for the transfer.
- Capitalization of the substitution parameters effects the capitalization of the output. See the File Name Token List for details.
- If a formatted name is given which contains an invalid substitution code, the transfer will fail with an error stating that a substitution code is bad.
- The feature was 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 /, \, or : 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 the generated file name, however MFT Platform Server for z/OS currently does not support embedded spaces in remote file names.

C.4 PPA Tokens

The PPA Substitutable fields use the percent character (%) as the escape character instead of the $ that file tokens use. Below is a list of the substitutable parameters that are supported for PPA.

For our example, assume that we have a file called: C:\a\b\c\d\config.txt

Substitutable Parameter	Description	Resolved Name Example
%DIR	Directory without the file name or drive	a\b\c\d sharename\a\b\c\
%DRIVE	Drive Name	C \\server\
%NODRIVE	File name without Drive	a\b\c\d\config.txt \sharename\a\b\config.txt
%SDIR	The lowest level directory	d
%HDIR	The high level directory	a
%NOSDIR	Directory name without 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 \\server\sharename\a\test.txt
%LLQ	Low Level Qualifier of file (data after last period(.))	txt
%HLQ	High level qualifier of file	config
%TRN	Transaction Number	I824500001
%PROC	Process Name	ABC123
%UDATA	User Data	USRDATAABC123
%JDATE	Julian Date (YYDDD)	05236
%JDATEC	Julian Date with Century (CCYYDDD)	2005236
%TIME	Time (hhmmss)	165030
%GDATE	Gregorian Date (yymmdd)	050824
%GDATEC	Gregorian Date with Century (ccyymmdd)	20050824

Note that 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. Note that 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 % sign. If UNIX detects the %DRIVE parameter, then the value DRIVE should be used as substitution. Similarly, %PROC becomes PROC and %UDATA becomes UDATA if not interacting with a z/OS system.

C.5 Directory Tokens

There are two special tokens that may be used for directory transfers. They are discussed below:

$(SDIR)

This case sensitive token may be used with a Receive as part of the LocalFileName path, and with a Send as part of the RemoteFileName path.

Example:
For a Receive, you would set:

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

RemoteFileName:
C:\MFT Platform Server\data*

The text before this token is assumed to be a base directory.
If **ScanSubDir** is checked on and there are files in both the remote directory (**C:\MFT Platform Server\data**) and in the remote subdirectories, then the same subdirectories will be created in the local directory (**C:\johndoe\data**) and local file names will be given as **$(RemoteFileName)** token.

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

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

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

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

$(MEMBER)

This token should be 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 than directory names.

So, this token allows you 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, this token will not be used. For example, if the path was **C:\MFT Platform Server\$(MEMBER)\whatever**, it will become **C:\MFT Platform Server\whatever**.

Appendix D

Appendix D. Configuring HIS for MFT Platform Server

MFT Platform Server for Windows can be configured to use SNA. Microsoft's HIS Product provides SNA protocol capabilities for Windows. To assist users to configure MFT Platform Server with an HIS server, we have included in this section an example step by step setup for an HIS 2004 server. Depending on the version of your HIS server you are running, the windows displayed may vary from the screenshots provided in this appendix.

First, you must have an installed and operating Microsoft HIS Server running once this is complete, open the SNA Manager. Below is an example of the main window:

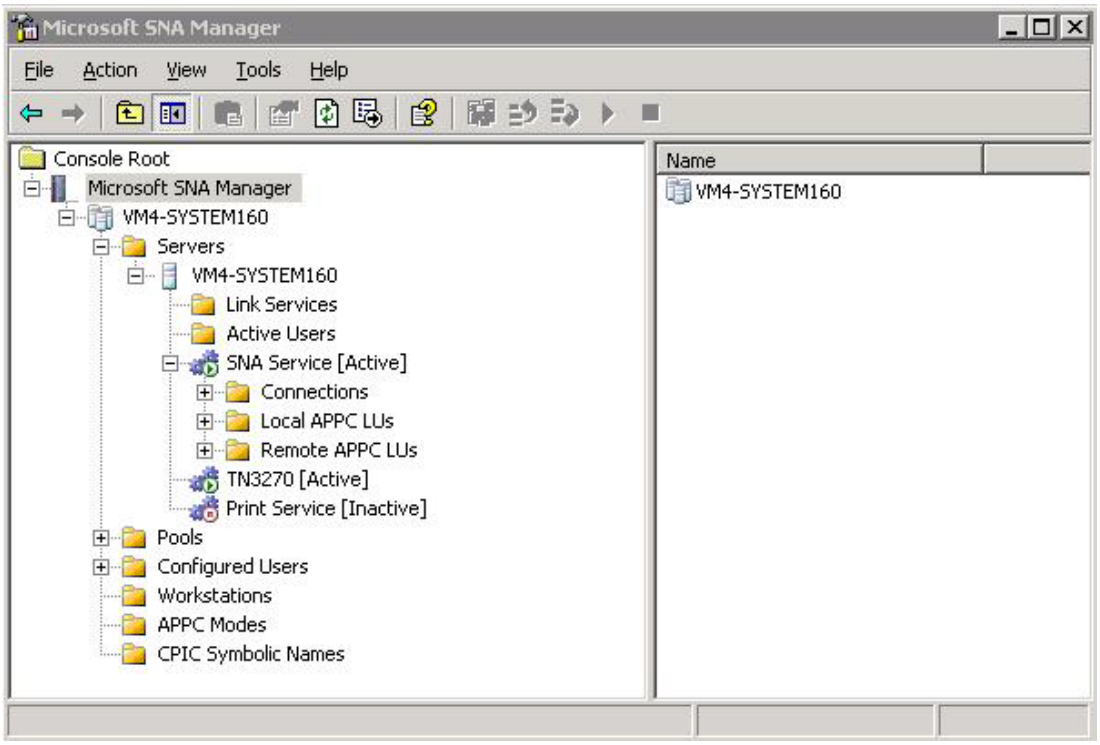


Figure D.1

Step 1: Create a DLC Link Service

You must create a DLC 802.2 Link Service. To do this, right click on the Link Services folder as seen in Figure D.1 above and then click on New, Link Service and you will be given the following list:

Figure D.2

Highlight DLC 802.2 Link Service by clicking on this menu item and then click on the **Add** button. The properties panel will open. Make sure your Local service access point (SAP) is set to 0x4. Accept all other default settings as seen in the screenshot below and click the **OK** button:

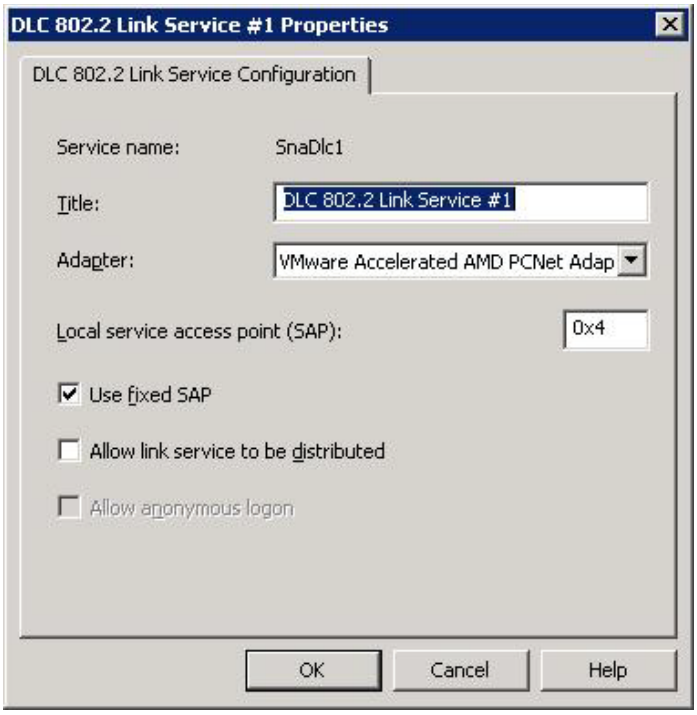


Figure D.3

Step 2: Configure an SNA Service Connection

To setup the connection using the new Link Service you created in Step 1 you must right click the folder named, Connections, and then click New, 802.2 as shown in Figure D.4 below:

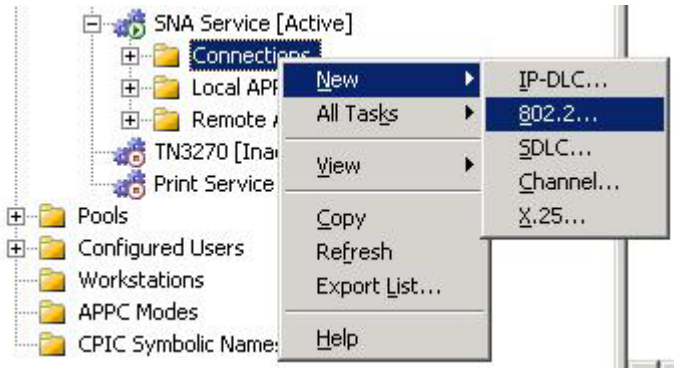


Figure D.4

The Connection Properties configuration window will open as seen in Figure D.5.

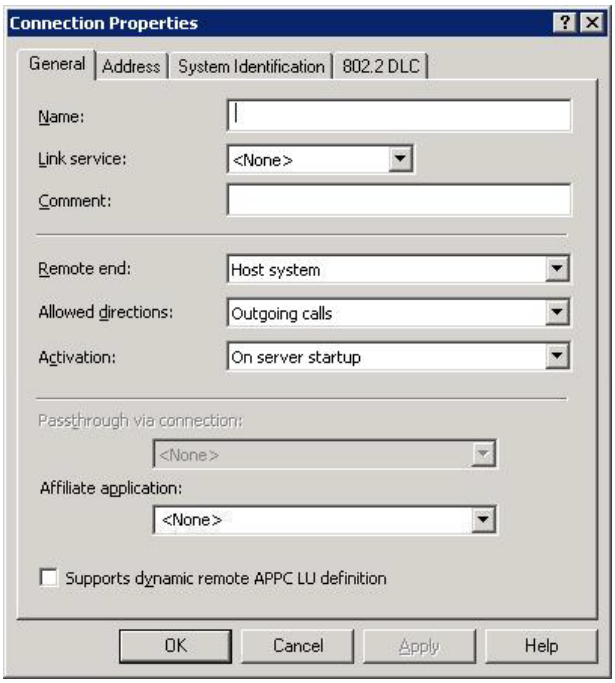


Figure D.5

On the General tab you would define the **Name** you will be using for the connection and the **Link Service** you setup in Step 1 by clicking on the Link service drop down menu. You can accept the default settings for all other options on this tab. Now click on the Address tab.

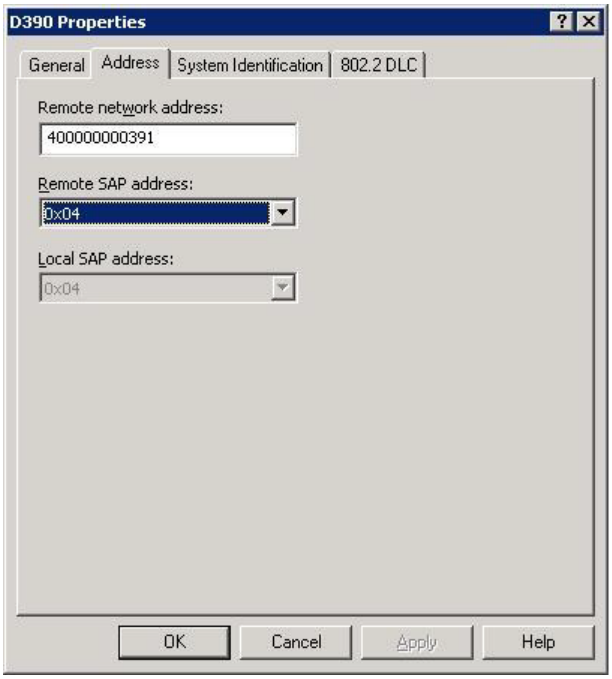


Figure D.6

On this tab you would configure the Remote network address being used by your mainframe. You can accept the defaults for all other options on this tab. Now click on the System Identification tab.

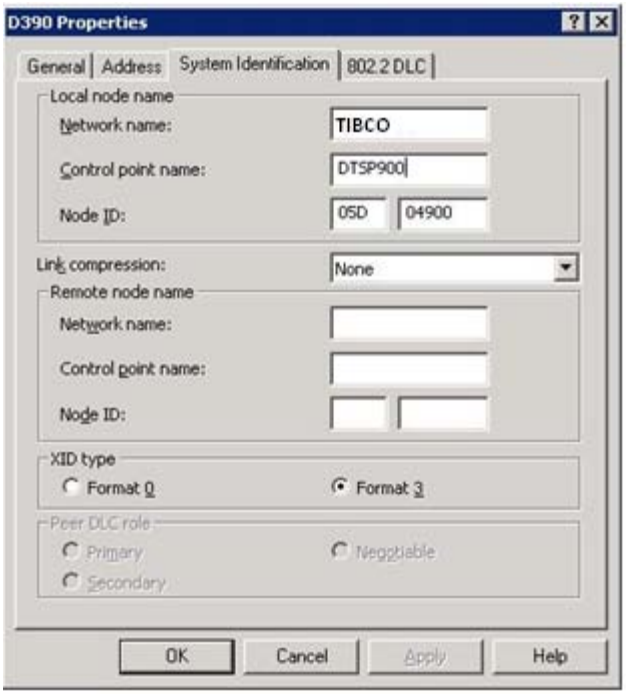


Figure D.7

On this tab you need to define your **Network name**, **Control point name** and your **Node ID** as they are defined on your mainframe. Format 3 should be configured. You can accept the defaults for all other options on this tab. Now click on the 802.2 DLC tab.

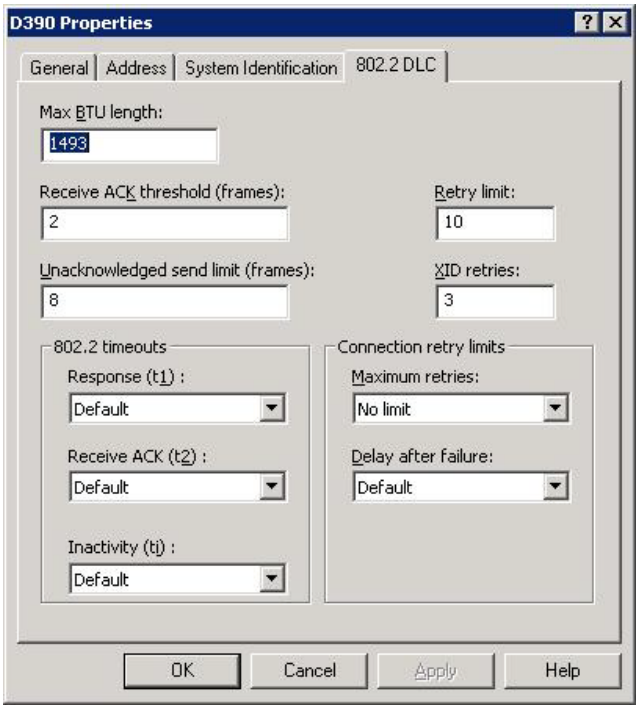


Figure D.8

For your 802.2 connection you want to set the Max BTU length to 1493. You can accept the defaults for all other options on this tab. Now click the **OK** button to save your new connection.

Step 3: Configure the Local LU

To setup the Local LU you must right click the folder named, Local APPC LUs, and then click New, Local LU as shown in Figure D.9 below:

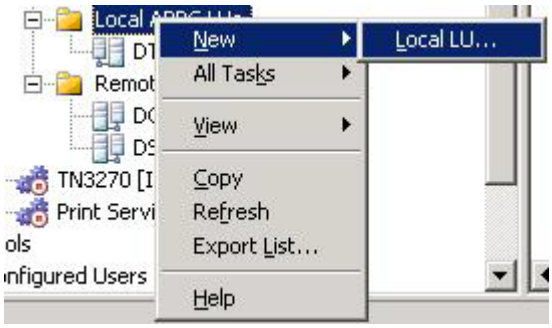


Figure D.9

The Local LU Properties configuration window will open. Fill in your **LU alias**, **Network name**, and **LU name** (this is generally the same as the LU alias). Figure D.10 shows an example Local LU setup:

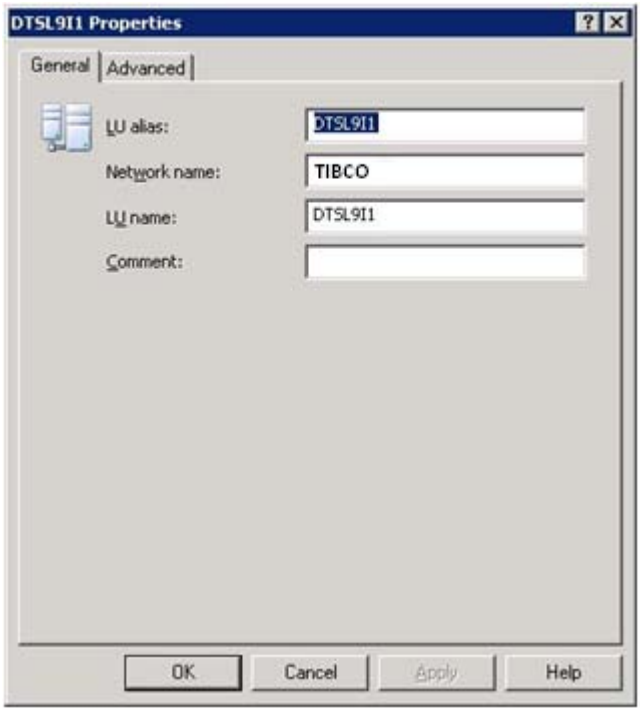


Figure D.10

You can accept the default settings for all options on the Advanced tab. Click **OK** when you are done.

Step 4: Configure the Remote LU

To setup the Remote LU you must right click the folder named, Remote APPC LUs, and then click New, Remote LU as shown in Figure D.12 below:

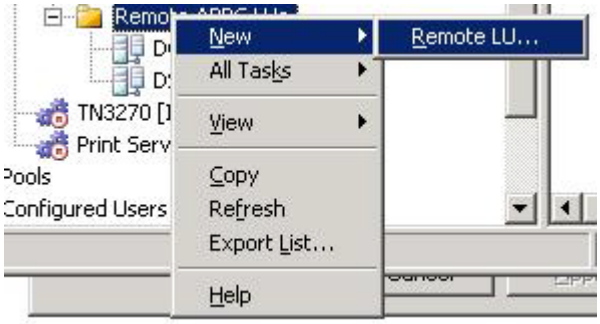


Figure D.12

The Remote APPC LU Properties window will open. From the Connection drop down menu choose your connection you created for earlier in Step 2. Define your partners (mainframe) **LU alias**, **Network name**, **LU name**, and the **Uninterpreted** name. Figure D.13 shows an example Remote LU setup:

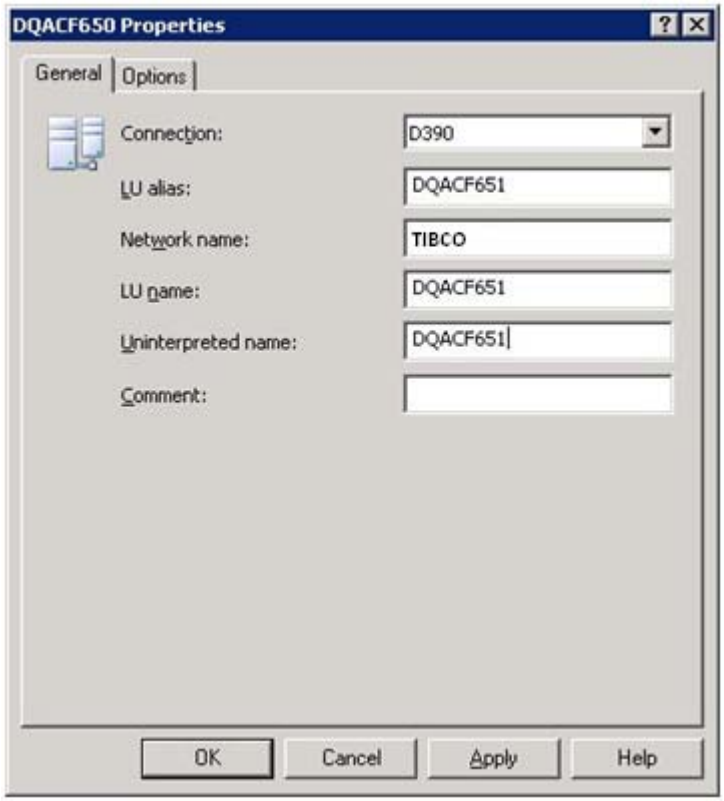


Figure D.13

You can accept the default settings for all options on the Options tab. Click **OK** when you are done.

Step 5: Configure the CPIC Name

The last step is to setup the CPIC Symbolic Name to use for MFT Platform Server. To configure this you would right click the folder named CPIC Symbolic Names, and then click New, CPIC Symbolic Name as shown in Figure D.14 below:

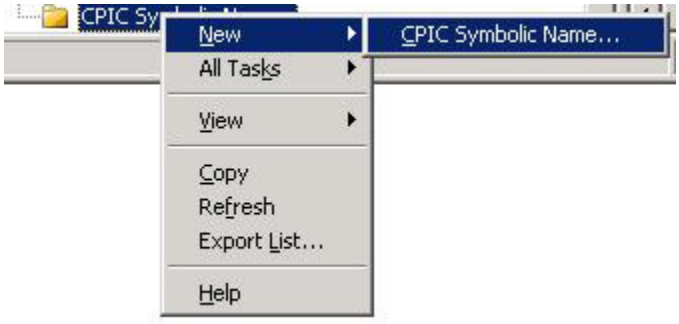


Figure D.14

The CPIC Name Properties window will open. On the General tab configure the **Name** as **ASNAS** and set the Mode name to #Batch as seen in Figure D.15 below:

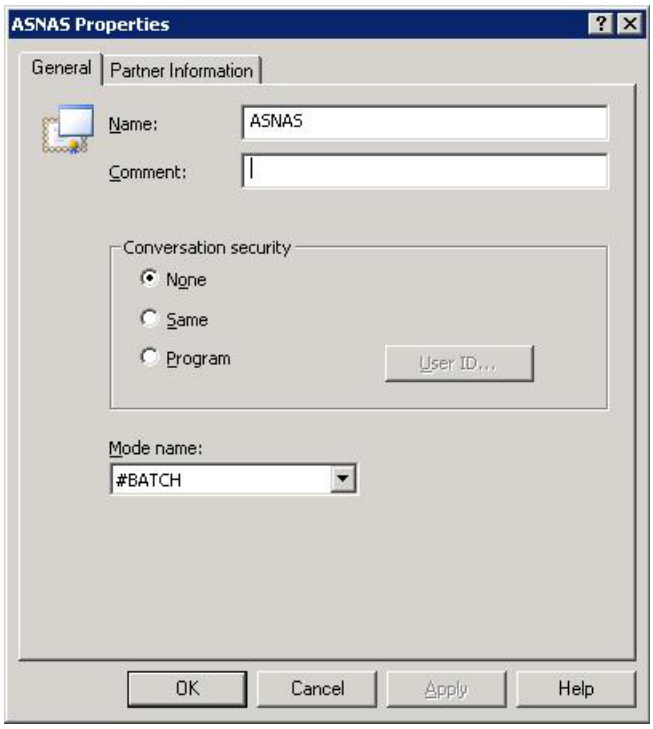


Figure D.15

Now click on the Partner Information tab. Set the **Application TP** to **ASNAS** and define the partners LU Alias name as seen in our example in Figure D.16. When you are done click on the **OK** button.

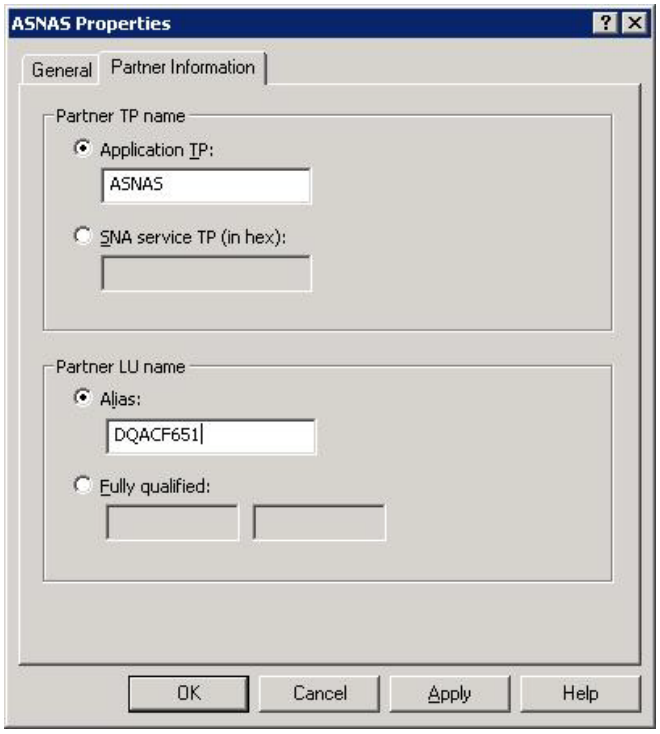


Figure D.16

This step completes our example configuration of a Microsoft HIS 2004 Server to be used to communicate with a mainframe using SNA with MFT Platform Server.

INDEX

