TIBCO® Managed File Transfer Internet Server

Utilities Guide

Version 8.5.0
March 2023

Contents

Utilities Overview

TIBCO® Managed File Transfer Internet Server has two command-line utilities that can be invoked from a batch file, a UNIX script, and run-in unattended mode by a job scheduler for ease of use. It also provides a promotion utility that can be invoked from the command line and the GUI using a batch file or a UNIX script.

The installation process creates the `MFTCC_install\distribution` directory that contains Admin Client Utility, the Platform Transfer Client Utility, and the Promotions Utility.

- The Admin Client Utility is designed for the administrator to conduct administrative operations through the command line on Windows and UNIX platforms.
- The Platform Server Command Line Client (or Platform Transfer Client Utility) is designed for the user to perform Platform Server transfers by using the command line on Windows and UNIX platforms.
- The Promotion Utility is designed for the user to copy definitions from one TIBCO MFT system to another TIBCO MFT system using the GUI mode that can run on Windows or UNIX with a GUI interface or the command line mode.

Utility Installation Files

You must use the utility installation files included with the installation setup files to install the utilities.

You must obtain the utilities from the following directories:

- `MFT_install\distribution\AdminClient`
- `MFT_install\distribution\PlatformTransfer`
- `MFT_install\distribution\InternetTransfer`

Admin Client Utility

Admin Client Utility contains two files in the directory, a zip file for Windows and a tar file for UNIX. Distribute the required file for the operating system you will be working on into a new folder.

The following table lists the 2 files in the `MFT_install\distribution\AdminClient` directory:

File Name	Supported Platform
`AdminClient.zip`	Windows
`AdminClient.tar`	UNIX

Platform Transfer Client Utility

Platform Transfer Client Utility contains two files in the directory, a zip file for Windows and a tar file for UNIX.

The following table lists the two files in the `MFT_install\distribution\PlatformTransfer` directory:

File Name	Supported Platform
`PlatformTransferClient.zip`	Windows
`PlatformTransferClient.tar`	UNIX

Internet Server Transfer Client Utility

Internet Server Transfer Client Utility contains two files in the directory, a zip file for Windows and a tar file for UNIX.

The following table lists the two files in the *`MFT_install`*`\distribution\InternetTransfer` directory:

File Name	Supported Platform
`InternetTransferClient.zip`	Windows
`InternetTransferClient.tar`	UNIX

Note: The Internet Server Transfer client is distributed with Command Center and Internet Server, but is executed on MFT Internet Server. For more information about the usage of this CLI utility, see the *TIBCO® Managed File Transfer Internet Server Utilities Guide*.

Promotion Utility

The MFT Promotion Utility contains a single file in the directory for either a Windows or a UNIX operating system.

The following table lists the file in the *`MFT_install`*`\distribution\MFTPromotionUtility` directory:

File Name	Supported Platform
`MFTPromotionUtility.zip`	Windows and UNIX

File Name	Supported Platform
	Note: The following six files are available in this file: `config.bat`, `config.sh`, `promoteGUI.bat`, `promoteGUI.sh`, `promote.bat`, `promote.sh`

Preparing to Install Utilities

Before installing and using the command-line utilities, ensure that you have installed JRE version 1.8 or later on the client.

Procedure

1. Download Java JRE from Oracle's official website.
2. Install Java JRE.
3. Add the Java `bin` directory to the `PATH` environment variable on your computer.

   On Windows, the default Java `bin` directory is `C:\Program Files\Java\`*`jre_version`*`\bin`.
4. Add the `JRE_HOME` environment variable to your system environment variables.

   For example, `JRE_HOME=C:\Program Files\Java\jre1.8`

Command-Line Utilities

The three command-line utilities shipped with TIBCO MFT Internet Server andTIBCO MFT Command Center are Administrator Command Line Client Utility, Platform Transfer Client Utility, and Internet Server Command Line Client Utility.

- Administrator Command Line Client Utility (or Admin Client Utility) is designed for the administrator to conduct administrative operations through the command line on Windows and UNIX platforms.

  > **Note:** Admin Client Utility is not available from the TIBCO MFT Internet Server download website.

- The Platform Transfer Client Utility allows you to configure and execute Platform Server Transfers from Command Center.
- Internet Server Command Line Client Utility (or Internet Transfer Client Utility) is designed to let the end-user perform transfers without the use of a web browser.

Installing and Configuring Command-Line Utilities

To use the utilities, you must configure the utilities as required.

Procedure

1. Download the appropriate utility file to a new folder. You must obtain the command-line utility from the following directories:

   `<MFT_install>\distribution\PlatformTransfer`

   `<MFT_install>\distribution\AdminClient`

   `<MFT_install>\distribution\InternetTransfer`

   If you are installing the Platform Transfer Client Utility on Windows, the following file is required:

- `PlatformTransferClient.zip`

If you are installing Platform Transfer Client Utility on UNIX, the following file is required:

- `PlatformTransferClient.tar`

or
If you are installing Admin Client Utility on a Windows, the following file is required:

- `AdminClient.zip`

If you are installing Admin Client Utility on a UNIX, the following file is required:

- `AdminClient.tar`

or

If you are installing the Internet Transfer Client Utility on a Windows, the following file is required:

- InternetTransferClient.zip

If you are installing Internet Server Client Utility on a UNIX, the following file is required:

- InternetTransferClient.tar

The directory of Platform Transfer Client Utility, Admin Client Utility and Internet Transfer client utility contains two files each: a `.zip` file for Windows and a `.tar` file for UNIX. See Installation Files for more details.

2. Extract the `.zip` file or the `.tar` file into the same directory where you obtained the files from Step 1.

   For example, run the following command on the UNIX platform to extract the `.tar` file:

   ```
   tar -xvf PlatformTransferClient.tar
   ```

   or

   tar –xvf AdminClient.tar

   > **Note:** If you want to use more than one utility on the same machine, ensure that you extract the utility files into their own directories.

3. Open a command line and navigate to the folder where the files are extracted, and then run the following command to set up the class path for the program:
   - On Windows: `setutilcp`
   - On UNIX: `. ./setutilcp.sh`
4. When the setup is complete, run `java cfcc.Config` and respond to the prompts to configure the server and certificate information.

   The following information is required during the configuration:

   - The user ID and password to connect to TIBCO MFT Internet Server .
   - The name of the Java trusted keystore.

     > **Note:** This file can be located in either the Java or directory. If the file does not exist, you will be asked whether you want to create the file.

   - The password for the trusted keystore.
   - The IP name or IP address of the server.
   - The IP port of the server.
   - The REST service version to use if you selected REST web service.
   - The server context.

Result

When the configuration is completed, the program connects to TIBCO MFT Internet Server or TIBCO MFT Command Center and sets up the necessary certificate files. With the provided information, the program performs the following functions during the configuration:

- Encrypts all passwords.
- Updates the `Global.xml` file.
- Validates the certificate and, if necessary, adds the certificate to the Java trusted keystore.
- Tests the connection to the TIBCO MFT Command Center server.

Internet Transfer Client Utility Sample Command

Internet Transfer Client Utility is designed to let the user perform transfers without the use of a web browser.

Below is a sample command using the Internet Transfer Client Utility program. This utility program is run from the same directory where the `.zip` or `.tar` files are extracted.

```
java cfcc.CFInternet a:ListDownloadFiles
```

CFInternet Commands

CFInternet accepts the following commands after the action parameter (`a:`).

Note: All commands should be typed as a single line with parameters separated by a space. They are shown on separate lines for readability.

The following commands are used to list information about files available for the user to transfer.

Command	Description
`ListAllFiles`	Lists all files available to transfer.
`ListUploadFiles`	Lists all files available for upload.
`ListDownloadFiles`	Lists all files available for download.
`ListFile`	Lists the file that matches defined selection criteria.

The following commands are used to perform file transfers.

Command	Description
ProcessAllFiles	Transfers all files currently available.
ProcessUploadFiles	Transfers upload files currently available.
ProcessDownloadFiles	Transfers download files currently available.
ProcessFile	Transfers file that matches defined selection criteria.

The following commands retrieve system information from the TIBCO MFT Internet Server system.

Command	Description
GetCopyrightInfo	Displays copyright information. **Note:** This command is not supported for REST web service.
GetProductNameVersion	Gets version information.
ChangePassword	Changes password in TIBCO MFT Internet Server .

ListAllFiles

The `ListAllFiles` command action is used to display a list of all the files that the user can upload or download.

The CFInternet client communicates with the defined by the `Global.xml` file. It extracts a list of all files that the user can transfer.

Note: This list is the same as the list of files that would be displayed by the TIBCO MFT Internet Server web interface.

Parameter	Description	Default	Required
`SubDir (sd)`	For directory uploads, if required, specifies TIBCO MFT Internet Server to scan subdirectories for files to transfer. For directory downloads, if required, specifies TIBCO MFT Internet Server to process data in TIBCO MFT Internet Server subdirectories. When No is specified, TIBCO MFT Internet Server processes files only in the defined directory. When Yes is defined, TIBCO MFT Internet Server processes files in subdirectories as well as in the defined directory. This parameter is valid only for TIBCO MFT Internet Server files defined with the directory flag. It is ignored for all other requests. This parameter is supported on all List and Process calls.	None	No

Sample ListAllFiles Command

The following sample command shows how to display a list of all file definitions that the user can upload or download from the command line.

```
java cfcc.CFInternet a:ListAllFiles
```

ListUploadFiles

The `ListUploadFiles` command action is used to display a list of all file definitions that the user can upload.

The CFInternet client communicates with the TIBCO MFT Internet Server defined by the `Global.xml` file. It extracts a list of all the files that the user can upload and displays the list of files.

Parameter	Description	Default	Required
`SubDir (sd)`	For directory uploads, if required, specifies TIBCO MFT Internet Server to scan subdirectories for files to transfer. For directory downloads, if required, specifies TIBCO MFT Internet Server to process data in TIBCO MFT Internet Server subdirectories. When No is specified, TIBCO MFT Internet Server processes files only in the defined directory. When Yes is defined, TIBCO MFT Internet Server processes files in subdirectories as well as in the defined directory. This parameter is valid only for TIBCO MFT Internet Server file definitions defined with the directory flag. It is ignored for all other requests. This parameter is supported on all List and Process calls.	None	No

Sample ListUploadFiles Command

The sample command below shows how to list files that are defined for upload.

```
java cfcc.CFInternet a:ListUploadFiles
```

ListDownloadFiles

The `ListDownloadFiles` command action is used to display a list of all files that the user can download.

The CFInternet client communicates with the TIBCO MFT Internet Server defined by the `Global.xml` file. It extracts a list of all files that the user can download and displays the list of files.

Parameter	Description	Default	Required
`SubDir (sd)`	For directory uploads, if required, specifies TIBCO MFT Internet Server to scan subdirectories for files to transfer. For directory downloads, if required, specifies TIBCO MFT Internet Server to process data in TIBCO MFT Internet Server subdirectories. When No is specified, TIBCO MFT Internet Server processes files only in the defined directory. When Yes is defined, TIBCO MFT Internet Server processes files in subdirectories as well as in the defined directory. This parameter is valid only for TIBCO MFT Internet Server file definitions defined with the directory flag. It is ignored for all other requests. This parameter is supported on all List and Process calls.	None	No

Sample ListDownloadFiles Command

The sample command below shows a list of files defined for download.

```
java cfcc.CFInternet a:ListDownloadFiles
```

ListFile

The `ListFile` command action is used to list the files that match the defined selection criteria.

The CFInternet client communicates with the TIBCO MFT Internet Server defined by the `Global.xml` file. It extracts a list of all files that the user can transfer and compares it against the filters that are defined. If multiple filters are defined, all filters must match for the file to be displayed.

Parameter	Description	Default	Required
`ClientFileName (cfn)`	Specifies the 1 to 256 byte client file name to be used as a filter. The `ClientFileName` is compared against the `ClientFileName` of the TIBCO MFT Internet Server file definitions returned to CFInternet. If they match, then the file is compared against any other filters defined. This field is case sensitive. The asterisk (*) may be used as a wildcard character. For example: `ClientFileName :/prod/acct/file1.txt`	None	No
`Description (d)`	Specifies the 1 to 256 byte description to be used as a filter. The `Description` is compared against the `Description` of the TIBCO MFT Internet Server file definitions returned to CFInternet. If they match, then the file is compared against any other filters defined.	None	No

Parameter	Description	Default	Required
	This field is case sensitive. The asterisk (*) may be used as a wildcard character. For example: `Description`:Prod_ ACCT_Y2005		
`FileId (fid)`	Specifies the ID of the transfer file definition.	None	No
`FileName (fn)`	This parameter is used only on directory download requests. It allows the user to define a single server file name to download. It is allowed only on ListFile and ProcessFile calls. The asterisk (*) may be used as a wildcard character.	None	No
`LocalFileName (lfn)`	Specifies the local file or directory name used for upload or download. If no value is provided, the client file name is used.	None	No
`SubDir (sd)`	For directory uploads, if required, specifies TIBCO MFT Internet Server to scan subdirectories for files to transfer. For directory downloads, if required, specifies TIBCO MFT Internet Server process data in TIBCO MFT Internet Server subdirectories. When No is specified, TIBCO MFT Internet Server processes files only in the defined directory.	None	No

Parameter	Description	Default	Required
	When Yes is defined, TIBCO MFT Internet Server processes files in subdirectories as well as in the defined directory. This parameter is valid only for TIBCO MFT Internet Server file definitions defined with the directory flag. It is ignored for all other requests. This parameter is supported on all List and Process calls.		

Sample ListFile Command

The sample command below shows how to list the file that matches the defined selection criteria from the command line.

```
java cfcc.CFInternet a:ListFile Description:Prod_ACCT_Y2005
```

ProcessAllFiles

The `ProcessAllFiles` command action is used to transfer all files that the user can upload or download.

The CFInternet client communicates with the TIBCO MFT Internet Server defined by the `Global.xml` file. It extracts a list of all files that the user can transfer. This command causes all files to be transferred. If one transfer is unsuccessful, TIBCO MFT Internet Server continues to the next transfer. The command terminates with one of the following return codes:

Code	Meaning
0	All files transferred successfully.

Code	Meaning
3	No files selected for processing.
4	Partial success.
8	All files transferred unsuccessfully.

The following table lists parameters supported for this command action.

Parameter	Description	Default	Required
`RetryInterval (ri)`	Specifies the number of seconds to wait before the next retry.	None	No
`RetryTimes (rt)`	Specifies the number of times to retry the transfer.	None	No
`SilentMode (sm)`	Specifies whether to display the byte count during transfer. The values supported for this parameter are as follows: • Y - does not display progress in byte count during transfer. • others - displays progress during transfer.	None	No
`SubDir (sd)`	For directory uploads, if required, specifies TIBCO MFT Internet Server to scan subdirectories for files to transfer. For directory downloads, if required, specifies TIBCO MFT Internet Server to process data	None	No

Parameter	Description	Default	Required
	in TIBCO MFT Internet Server subdirectories. When No is specified, TIBCO MFT Internet Server processes files only in the defined directory. When Yes is defined, TIBCO MFT Internet Server processes files in subdirectories as well as in the defined directory. This parameter is valid only for TIBCO MFT Internet Server file definitions defined with the directory flag. It is ignored for all other requests. This parameter is supported on all List and Process calls.		

Sample ProcessAllFiles Command

The sample command below shows how to transfer all files that the user can upload or download.

```
java cfcc.CFInternet a:ProcessAllFiles
```

ProcessUploadFiles

The `ProcessUploadFiles` command action is used to transfer all files that the user can upload.

The CFInternet client communicates with the TIBCO MFT Internet Server defined by the `Global.xml` file. It extracts a list of all files that the user can upload. This command causes all files to be uploaded. It does not transfer files to be downloaded. If one transfer is unsuccessful, TIBCO MFT Internet Server continues to the next transfer. The command terminates with one of the following return codes:

Code	Meaning
0	All files transferred successfully.
3	No files selected for processing.
4	Partial success.
8	All files transferred unsuccessfully.

The following table lists parameters supported for this command action.

Parameter	Description	Default	Required
`RetryInterval (ri)`	Specifies the number of seconds to wait before the next retry.	None	No
`RetryTimes (rt)`	Specifies the number of times to retry the transfer.	None	No
`SilentMode (sm)`	Specifies whether to display the byte count during transfer. The values supported for this parameter are as follows: • Y - does not display progress in byte count during transfer. • others - displays progress during transfer.	None	No
`SubDir (sd)`	For directory uploads, if required, specifies TIBCO MFT Internet Server to scan subdirectories for files to	None	No

Parameter	Description	Default	Required
	transfer. For directory downloads, if required, specifies TIBCO MFT Internet Server to process data in TIBCO MFT Internet Server subdirectories. When No is specified, TIBCO MFT Internet Server processes files only in the defined directory. When Yes is defined, TIBCO MFT Internet Server processes files in subdirectories as well as in the defined directory. This parameter is valid only for file definitions defined with the directory flag. It is ignored for all other requests. This parameter is supported on all List and Process calls.		

Sample ProcessUploadFiles Command

The sample command below shows how to transfer all files that the user can upload.

```
java cfcc.CFInternet a:ProcessUploadFiles
```

ProcessDownloadFiles

The `ProcessDownloadFiles` command action is used to transfer all files that the user can download.

The CFInternet client communicates with the TIBCO MFT Internet Server defined by the `Global.xml` file. It extracts a list of all files that the user can download. This command causes all files to be downloaded. It does not transfer files to be downloaded. If one transfer is unsuccessful, TIBCO MFT Internet Server continues to the next transfer. The command terminates with one of the following return codes:

Code	Meaning
0	All files transferred successfully.
3	No files selected for processing.
4	Partial success.
8	All files transferred unsuccessfully.

The following table lists parameters supported for this command action.

Parameter	Description	Default	Required
`RetryInterval (ri)`	Specifies the number of seconds to wait before the next retry.	None	No
`RetryTimes (rt)`	Specifies the number of times to retry the transfer.	None	No
`SilentMode (sm)`	Specifies whether to display the byte count during transfer. The values supported for this parameter are as follows: • Y - does not display progress in byte count during transfer. • others - displays progress during transfer.	None	No
`SubDir (sd)`	For directory uploads, if required, specifies TIBCO MFT	None	No

Parameter	Description	Default	Required
	Internet Server to scan subdirectories for files to transfer. For directory downloads, if required, specifies TIBCO MFT Internet Server to process data in TIBCO MFT Internet Server subdirectories. When No is specified, TIBCO MFT Internet Server processes files only in the defined directory. When Yes is defined, TIBCO MFT Internet Server processes files in subdirectories as well as in the defined directory. This parameter is valid only for TIBCO MFT Internet Server file definitions defined with the directory flag. It is ignored for all other requests. This parameter is supported on all List and Process calls.		

Sample ProcessDownloadFiles Command

The sample command below shows how to transfer all files that the user can download.

```
java cfcc.CFInternet a:ProcessDownloadFiles
```

ProcessFile

The `ProcessFile` command action is used to transfer a file that matches the defined selection criteria.

The CFInternet client communicates with the TIBCO MFT Internet Server defined by the `Global.xml` file. It extracts a list of all files that the user can transfer and compares it against the filters that are defined. If multiple filters are defined, both filters must match for the file to be displayed. This command causes the file that matches the filters to be transferred (either uploaded or downloaded). If one transfer is unsuccessful, TIBCO MFT Internet Server continues to the next transfer. The command terminates with one of the following return codes:

Code	Meaning
0	All files transferred successfully.
3	No files selected for processing.
4	Partial success.
8	All files transferred unsuccessfully.

The following table lists parameters supported for this command action.

Parameter	Description	Default	Required

`ClientFileName`
`(cfn)`
	Specifies the 1 to 256 byte client file name to be used as a filter.
	The `ClientFileName` is compared against the `ClientFileName` of the TIBCO MFT Internet Server file definitions returned to CFInternet. If they match, then the file is compared against any other filters defined.
	This field is case sensitive. The asterisk (*) may be used as a wildcard character.
	For example: `ClientFileName`: NYACCT_Test_File
		None	No

Parameter	Description	Default	Required
`Description (d)`	Specifies the 1 to 256 byte description to be used as a filter. The Description is compared against the description of the TIBCO MFT Internet Server file definitions returned to CFInternet. If they match, then the file is compared against any other filters defined. This field is case sensitive. The asterisk (*) may be used as a wildcard character. For example: `Description`: NYACCT_Test_File	None	No
`LocalFileName (lfn)`	Specifies the 1 to 256 byte local file name. This file name replaces the `ClientFileName` defined by the TIBCO MFT Internet Server file definition. When a file is uploaded, this field defines the client source file name. For example, the file that is read and sent to the remote system. When a file is downloaded, this field defines the client target file name. For example, the file that is written to the local system. This field is case sensitive on some platforms such as UNIX. For example: `LocalFileName :/prod/cfcc/NY/file1.abc`	None	No

Parameter	Description	Default	Required
`SubDir (sd)`	For directory uploads, if required, specifies TIBCO MFT Internet Server to scan subdirectories for files to transfer. For directory downloads, if required, specifies TIBCO MFT Internet Server process data in TIBCO MFT Internet Server subdirectories. When No is specified, TIBCO MFT Internet Server processes files only in the defined directory. When Yes is defined, TIBCO MFT Internet Server processes files in SubDirectories as well as in the defined directory. This parameter is valid only for TIBCO MFT Internet Server file definitions defined with the directory flag. It is ignored for all other requests. This parameter is supported on all List and Process calls.		
`FileName (fn)`	This parameter is used only on directory download requests. It allows the user to define a single server file name to download. It is allowed only on ListFile and ProcessFile calls. The asterisk (*) may be used as a wildcard character.		
`FileId (fid)`	Specifies the ID of the transfer file definition.	None	No

Parameter	Description	Default	Required
`RetryInterval (ri)`	Specifies the number of seconds to wait before the next retry.	None	No
`RetryTimes (rt)`	Specifies the number of times to retry the transfer.	None	No
`SilentMode (sm)`	Specifies whether to display the byte count during transfer. The values supported for this parameter are as follows: • Y - does not display progress in byte count during transfer. • others - displays progress during transfer.	None	No

Sample ListFile Command

The sample command below shows how to transfer a file that matches the defined selection criteria.

```
java cfcc.CFInternet a:ProcessFile Description:Prod_ACCT_Y2005
ClientFileName:my.cfcc.file LocalFilename:/prod/cfcc/ny/file1.abc
```

GetCopyrightInfo

The `GetCopyrightInfo` command action is used to display copyright information about TIBCO MFT Internet Server.

No parameters are supported for this command action.

Note: This command is not supported for REST web service.

Sample GetCopyrightInfo Command

The command below displays the TIBCO MFT Internet Server copyright information.

```
java cfcc.CFInternet a:GetCopyrightInfo
```

GetProductNameVersion

The `GetProductNameVersion` command action is used to display version information about TIBCO MFT Internet Server.

No parameters are supported for this command action.

Sample GetProductNameVersion Command

The command below displays the version of the TIBCO MFT Internet Server product.

```
java cfcc.CFInternet a:GetProductNameVersion
```

Get Command Action - Change Password

The `ChangePassword` command action is used to change the user password in TIBCO MFT Internet Server.

The following table lists parameters supported for this command action.

Parameter	Description	Default	Required
`NewPassword (np)`	The new password that a user sets.	None	Yes

Sample ChangePassword Command

The following command is used to change and set a new password in TIBCO MFT Internet Server.

```
java cfcc.CFInternet a:ChangePassword
```

Action File

The action Template file is an XML file specified by `T` parameter in the command line. Using an action template file would allow you to define all information in a single file.

Alternatively, you can put multiple actions in one file, specified using the following XML format:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE actions SYSTEM "siftactions.dtd">
<actions>
<action name="action1" output=”action2:file1”>
<arg name="arg1" value="somevalue" sc="a1"/>
……
</action>
……
</actions>
```

The <action> element defines an action. The <arg> element defines a parameter needed for this action. If there are multiple <action> elements in the file, the program executes them one by one.

The ‘name’ attribute for <action> element specifies the action name. This must be a valid action. The XML file names are all valid actions.

The ‘name’ attribute for <arg> element specifies the parameter name for an action. The name is case sensitive and should not be edited. The `sc` attribute for <arg> element specifies a shortcut name for the `name` attribute, and it is case insensitive. You can use shortcut names to specify values in command line to replace default values specified in this file. If the action is specified by `A` parameter in command line, you must specify a parameter name for that action rather than a shortcut name. Shortcut names can be found in each XML file.

Shortcuts Usage in the Action File

One of the advantages of using the Action file template is that you can use shortcuts to define the parameter names.

Below is an example using shortcuts defined:

```
java cfcc.CFInternet U:xyz P:xyz KN:certificate KP:pswd a:ListAllFiles TKN:cacerts
TKP:changeit
```

If the `Global.xml` file is updated to contain the userid, password and keystore information, then run the following command:

```
java cfcc.CFInternet a:ListAllFiles
```

For client certificate authentication, the client must specify the keystore for its certificate via the Java system parameter, or via the command line's `KN` and `KP` parameters. To run the program over an SSL connection, the certificate authority (CA) that signed client certificates must be a trusted CA. This may require you to update your keystore.

Note: The batch file to setup classpath overwrites the default system classpath. Experienced users are encouraged to use other environment variables for classpath, and specify classpath in the Java command.

Name	Description
`U`	The user ID is sent to the web service for authentication to use the web service. May be specified in `Global.xml`.
`P`	The user password is sent to the web service for authentication to use the web service. May be specified in `Global.xml`.
`A`	The action to take. For example, add a file. If the parameter is specified, the program ignores the `T` parameter that specifies the action file name. The program only accepts one action from command line.
`T`	The action Template file name. The file can contain multiple actions in XML format. The program executes all actions specified in the file. If the program specified the `A` parameter, this parameter is ignored.
`TL`	The trace level. This value only affects this utility. This parameter should only be set when instructed to do so by TIBCO Technical Support. The valid value range is 0-10.

Name	Description
`TD`	The trace directory. This value only affects this utility. Sets the directory where the trace file or files are written.
`G`	The global template file name. The default one is `Global.xml` in the current directory.
`S`	The web service address. For example, https://DNS_ HostName:httpsPort/cfcc/…..
`KN`	The Java keystore name for client certificate authentication. Keystore name can be specified as a Java parameter, in which case, it is not necessary to use this parameter again. May be specified in `Global.xml`.
`KP`	The Java keystore password for client certificate authentication. The keystore password can be specified as a Java parameter, in which case, it is not necessary to use this parameter again. May be specified in `Global.xml`.
`TKN`	The trusted Java keystore name for certificate authentication. This file should contain the name of the keystore file that contains the Java Trusted Certificate Authorities. You can leave this parameter blank if you want to use the default trusted keystore. May be specified in `Global.xml`.
`TKP`	The trusted Java keystore password for client certificate authentication. If the default password is used, you can leave this parameter blank. May be specified in `Global.xml`.
`AD`	The audit file directory. This parameter defines the directory where the audit file is written. This should point to an existing directory and should not include a file name. creates the file name in the following format: `MFTIS Audit_YYYYMMDD.xml`.
`help`	The program displays the command line parameter list.

Name	Description
`help:action`	The program displays the parameters needed for the action if the action is a valid action; otherwise, display all currently supported actions.
`[name:value]`	Other name:value pairs. These values are used to assign the parameters' value if the action is specified by `A` parameter, or to replace the default values if `T` parameter is used. The 'name' is case sensitive if 'name' is a parameter name for an action. The 'name' is not case sensitive if 'name' is a shortcut for a real parameter name.

The following entries are defined in the `addFile.xml` file.

```
<arg name="ClientFileName" value="clientFileName" sc="CFN"
description="Client File Name"/>
<arg name="ServerFileName" value="serverFileName" sc="SFN"
description="Server File Name"/>
<arg name="Description" value="fileDesc" sc="D" description="File
Description"/>
<arg name="UserId" value="user id" sc="UID" description="UserID
authorized to transfer this file"/>
```

Note: At the end of each line, there is a parameter that starts with the value sc=. This is the shortcut name defined by the XML file. When executing the TIBCO MFT Internet Server File Transfer Command Line Utility with the Action File parameter `T:` defined, you can use the shortcut name instead of the actual parameter name.

As indicated in the previous example, when you define the client file name, you can use the `CFN` parameter instead of the `ClientFileName` parameter. The following example describe the standard parameters and shortcuts that you can use in TIBCO MFT Internet Server commands:

- Using standard parameter names:

  ```
  java cfcc.CFInternet a:ProcessFile
  ClientFileName:client.file1
  LocalFileName:prod.file.name
  ```

```
Description:”file upload”
```

- Using shortcut parameter names:

```
java cfcc.CFInternet t:ProcessFile.xml
CFN:client.file1
LFN:prod.file.name
D:”file upload”
```

The parameter names are much shorter when using the shortcut parameters. The shortcut parameter names can only be used when the Action File Template `T:` parameter is used in the TIBCO MFT Internet Server File Transfer Command Line Utility. The shortcut values must be defined by the sc= value in the template.

The shortcut names can be changed by the user. The shortcut names defined in the XML template are the default shortcut names. In the above mentioned example of using shortcut parameter names text box, the `CFN` parameter is defined as the shortcut name for the `ClientFileName` parameter. You can change this value to any value that you want, as long as the value does not conflict with an existing parameter name or shortcut value. For example, you could use a text editor to change the value CFN to CN. Therefore, you could use the value CN in the command line to reference the `ClientFileName` parameter whenever you used that XML template file.

CFInternet Audit File

CFInternet creates an audit record for every file transfer request attempted.

The audit file name is in the following format: `CFCCAudit_YYYYMMDD.xml`

The audit file is created in the directory defined by the `Global.xmlauditdirectory` parameter. If this parameter is not defined, the file is created in the current working directory. Note that the user can override the `auditdirectory` defined in the `Global.xml` file by specifying the `AD` parameter on the command line.

If the file defined by the `auditdirectory` parameter does not exist, then TIBCO MFT Internet Server creates the file. If the file does exist, TIBCO MFT Internet Server appends any audit records to the end of this file. One audit file is created for each day that a TIBCO MFT Internet Server file transfer is attempted. This file is created in XML format. There is one XML tag for each audit record created. Within each tag are attributes to define each field written to the audit log.

Below is a sample of the TIBCO MFT Internet Server audit file. Because the audit file is written using XML, it can be opened as a spreadsheet by Microsoft Excel. Likewise, it can be opened by any other editor that supports XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<Audits>
<Audit>
<AuditId value="A60950000035"/>
<FileId value="F50950000001"/>
<LocalUser value="admin"/>
<LocalHost value="DBKN2G01(192.168.100.1)"/>
<Type value="Upload"/>
<CompletionDate value="20050609"/>
<CompletionTime value="16:32:03"/>
<StartDate value="20050609"/>
<StartTime value="16:32:01"/>
<Status value="SUCCESS"/>
<Bytes value="256000"/>
<CompressedBytes value="123765"/>
<StatusMsg value="Transfer Complete. AuditID=A60950000035"/>
<ClientFileName value="c:\temp\testcfcc.txt"/>
<Description value="Test_MFT Internet Server_File"/>
<LocalFileName value=""/>
</Audit>
<Audit>
<AuditId value="A60950000035"/>
<FileId value="F50950000001"/>
<LocalUser value="admin"/>
<LocalHost value="DBKN2G01(192.168.100.1)"/>
<Type value="Upload"/>
<CompletionDate value="20050609"/>
<CompletionTime value="16:32:03"/>
<StartDate value="20050609"/>
<StartTime value="16:32:01"/>
<Status value="SUCCESS"/>
<Bytes value="256000"/>
<CompressedBytes value="123765"/>
<StatusMsg value="Transfer Complete. AuditID=A60950000035"/>
<ClientFileName value="c:\temp\testcfcc.txt"/>
<Description value="Test_MFT Internet Server_File"/>
<LocalFileName value=""/>
</Audit>
</Audits>
```

Admin Client Utility Sample Command

The Admin Client Utility program is designed for the administrator to conduct administrative operations through the command prompt on Windows and UNIX platforms.

Admin Client Utility is run from the same directory where the three `.jar` files are unpacked.

CFAdmin Commands

The commands of Admin Client Utility are used to define, list, update, and delete the definition records in the system.

CFAdmin will accept the following commands after the action parameter (`a:`).

Command Groups	Commands
Audit Commands	GetAudit
	RemoveAudit
	SearchForAudits
Department Commands	AddDepartment
	GetDepartment
	RemoveDepartment
	RetrieveAllDepartments
	UpdateDepartment

Command Groups	Commands
Group Commands	AddGroup
	AddUserToGroup
	GetGroup
	RemoveGroup
	RetrieveAllGroups
	RetrieveAllGroupsForUser
	RetrieveAllUsersInGroup
	RemoveUserFromGroup
PGP Public Keys	AddPGPPublicKey
	DeletePGPPublicKey
	GetPGPPublicKey
	RetrievePGPPublicKeys
	UpdatePGPPublicKey
Protocol Public Keys	AddProtocolPublicKey
	DeleteProtocolPublicKey
	GetProtocolPublicKey
	RetrieveProtocolPublicKeys
	UpdateProtocolPublicKey

Command Groups	Commands
Sync LDAP Authenticators	SyncAll
	SycAuth
	SyncUser
Role Commands	AddUserToRole
	GetRole
	RetrieveAllRoles
	RetrieveAllRolesForUser
	RetrieveAllUsersInRole
	RemoveUserFromRole
Server Commands	AddServer
	GetServer
	RetrieveAllServers
	RemoveServer
	UpdateServer
Session Commands	DeleteSessionId
	DeleteExpiredSessionIds
	GetExpiredSessionIds
	Note: The listed session commands are not supported when using REST web service.

Command Groups	Commands
Transfer Commands	AddTransfer
	DeleteExpiredTransfers
	GetTransfer
	RetrieveAllTransfers
	RetrieveAllTransfersForUser
	RemoveTransfer
	SearchForTransfers
	UpdateTransfer
User Commands	AddAdminUser
	AddTransferUser
	ChangePassword
	GetUser
	RetrieveAllUsers
	RemoveUser
	UpdateUser

Command Groups	Commands
User Profile Commands	AddUserProfile
	GetUserProfiles
	RetrieveAllUserProfiles
	RemoveUserProfile
	UpdateUserProfile
Miscellaneous Commands	GetProductNameVersion
	Help

Audit Commands

The audit commands are used to list and delete audit records in the system.

Action	Description
GetAudit	Displays a specific audit record.
RemoveAudit	Removes an audit record.
SearchForAudits	Searches for audit records.

GetAudit

The `GetAudit` command action is used to display a specific audit record.

To use the `GetAudit` action command, you must have AdministratorRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`AuditId (aid)`	Specifies the 12-character audit ID of the transfer you want to display.	None	Yes

Sample `GetAudit` Command

This command displays the information for the file transfers for the audit ID given.

```
java cfcc.CFAdmin a:GetAudit AuditId:A51450000142
```

RemoveAudit

The `RemoveAudit` command action is used to delete the specific audit records from the system.

The `RemoveAudit` command action deletes audit records in two ways:

- You can specify the number of days to keep audit records. All audit records written before the oldest day will be purged.
- You can specify a purge date. All records written before that date will be purged.

To use the `RemoveAudit` action command, you must have AdministratorRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`Days (day)`	Specifies the number of days of audit records that should be saved.	None	Either the `Days` or `PurgeDate` parameter must be specified.
`PurgeDate (pd)`	Specifies the purge date. Any audit record written before the purge date will be deleted.	None	Either the `Days` or `PurgeDate` parameter must be specified.

Parameter	Description	Default	Required
	The purge date format is *YYYY/MM/DD*.		
`ServerType (st)`	Specifies the server type. The valid values are as follows: - I: Internet server - P: Platform server - B: Both	B: Both	No

Sample `RemoveAudit` Command

This command keeps audit records written within 30 days. Any audit record written before 30 days will be purged.

```
java cfcc.CFAdmin a:RemoveAudit Days:30
```

SearchForAudits

The `SearchForAudits` command action is used to search for all audit records that match the defined selection criteria.

You should use the asterisk (*) as a wildcard character for REST web service in defined parameters to select file records based on a partial key.

> **Note:** Detailed information is displayed for all audit records that match the selection criteria.

To use the `SearchForAudits` action command, you must have AdministratorRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`AS2MDNStatus (as2s)`	Specifies the AS2 MDN status. - S: Success - F: Failure - P: Pending	None	No
`AuditId (aid)`	Specifies the 12-character audit ID that is assigned when the audit record is added. This parameter supports wildcard characters.	None	No
`ClientFileName (cfn)`	Specifies the 1-to-256-character file name/location on the client machine. If the file name/location contains embedded blanks the entire filename should be enclosed in double quotation marks (" "). This parameter supports wildcard characters.	None	No
`Days (day)`	Specifies the number of days to be searched. The way that the `Days` parameter is used depends on whether the `FromDate` and `ToDate` parameters are defined: - Both `FromDate` and	1	No

Parameter	Description	Default	Required
	ToDate defined – Days are ignored. • Only FromDate defined – The Days parameter defines the number of days after the FromDate that are searched. • Only ToDate defined – The Days parameter defines the number of days before the ToDate that are searched. • Neither FromDate nor ToDate defined – Days defines the number of days before the current date that are searched. • FromDate, ToDate and Days not defined – scans for today's audit records only.		

Parameter	Description	Default	Required
	Note: The `Days` parameter gives the total number of days that are scanned. If you specify `FromDate`:2004/12/01 and `Days`:10 parameters, then scans from 2004/12/01 until 2004/12/10; this searches a total of 10 days.		
`Department (dpt)`	Specifies the department for the audit search.	None	No
`FileId (Transfer Id) (tid)`	Specifies the 12-character transfer ID that is assigned when the file definition is added.	None	No
`FromDate (fd)`	Specifies the start date for your audit search. This can be combined with either the `ToDate` or `Days` parameter to define the dates to be returned. The format of the `FromDate` is YYYY/MM/DD. This parameter does not support wildcard characters.	None	No
`FromTime (ft)`	Specifies the start time for your audit search.	None	No

Parameter	Description	Default	Required
	This time is relative to the starting date only. The search starts from the `FromTime` on the `FromDate` and extends to the `ToTime` on the `ToDate`. The format of the `FromTime` is HHMM and the time is defined using military time (0000-2359). This parameter does not support wildcard characters.		
`LocalTransactionId (ltid)`	Specifies the 10 character MFT local transaction ID that is assigned by when the file transfer started. This parameter supports wildcard characters.	None	No
`Node Name (nn)`	Specifies the name of the node for the audit search.	None	No
`Process Name (pn)`	Specifies the name of the process for the audit search.	None	No
`Server Name (sn)`	Specifies the name of the server for the audit search.	None	No
`ServerFileName (sfn)`	Specifies the 1-to-256-character file name/location of the server machine.	None	No

Parameter	Description	Default	Required
	If the `NodeName` is *LOCAL, the `ServerFileName` would be located on the TIBCO MFT Internet Server. If the file name/location contains embedded blanks, the entire file name must be enclosed in double quotation marks (" "). This parameter supports wildcard characters.		
`ServerType(or Audit Type) (st)`	Specifies the server type: - I: Internet Server - P: Platform Server - B: both	None	No
`ToDate (tod)`	Specifies the end date for the audit search. This can be combined with either the `FromDate` or `Days` parameter to define the dates to be returned. The format of the `ToDate` is YYYY/MM/DD. The `ToDate` must be greater than the `FromDate`.	None	No
`ToTime (tt)`	Specifies the end time for the audit search. This time is relative to the ending date only. The search starts from the `FromTime` on the `FromDate`	None	No

Parameter	Description	Default	Required
	and extends to the `ToTime` on the `ToDate`. The format of the `ToTime` is HHMM and the time is defined using military time (0000-2359). This parameter does not support wildcard characters.		
`TransferStatus (ts)`	Specifies whether you want to extract successful transfers, failed transfers, or both. The valid values are as follows: • S: successful transfers to be returned. • F: failed transfers to be returned. If you want both successful and failed transfers to be returned, you should omit this field. This parameter does not support wildcard characters.	None Returns both successful and failed transfers.	No
`TransferUserId (tu)`	Specifies the 1-to-32-characters MFT user ID that is used to initiate the file transfer request with MFT. MFT user IDs can be defined in the file record,	None	No

Parameter	Description	Default	Required
	node records or by the user profile record. This parameter supports wildcard characters.		
`User Data (ud)`	Specifies information about the user for the audit search.	None	No
`Virtual Alias (va)`	Specifies the virtual alias.	None	No

Sample `SearchForAudits` Command

This command searches for all audit records that match the selection criteria. It will search for all failed transfers with the NYNode1 node within the past 5 days.

```
java cfcc.CFAdmin a:SearchForAudits NodeName:NYNode TransferStatus:F Days:5
```

Department Commands

The department commands are used to define, list, update, and delete department definition records in the system.

Action	Description
AddDepartment	Adds a department definition to .
GetDepartment	Lists a specific department definition.
RemoveDepartment	Deletes a department definition.
RetrieveAllDepartments	Lists all department definitions.
UpdateDepartment	Alters a department definition.

AddDepartment

The `AddDepartment` command action is used to define a department.

The delegated administration offers an administrator the ability to divide the system into smaller units which can be managed independently of one another. The departments can be all users at a specific location, business unit, or whatever grouping you chose.

To use the `AddDepartment` action command, you must be a super administrator. For more information, see "Delegated Administration" of *TIBCO Managed File Transfer Command Center User's Guide*.

Parameter	Description	Default	Required
`Description (d)`	Specifies the 1-to-64-character description of this department. If the description contains embedded spaces, the entire description must be enclosed in double quotation marks ("").	None	No
`Name (dn)`	Specifies the 1-to-64-character department name.	None	Yes

Sample `AddDepartment` Command

This command adds a department.

```
java cfcc.CFAdmin a:AddDepartment Name:Shoes Description:"Womens Shoe Department"
```

GetDepartment

The `GetDepartment` command action is used to display a department in the system.

To use the `GetDepartment` action command, you must be a super administrator. For more information, see "Delegated Administration" of *TIBCO Managed File Transfer Command Center User's Guide*.

Parameter	Description	Default	Required
`Name (dn)`	Specifies the 1-to-64-character department name.	None	Yes

Sample `GetDepartment` Command

This command displays the parameters for the NorthEast department.

```
java cfcc.CFAdmin a:GetDepartmentName:NorthEast
```

RemoveDepartment

The `RemoveDepartment` command action is used to delete a department from the system.

To use the `RemoveDepartment` action command, you must be a super administrator. For more information, see "Delegated Administration" of *TIBCO Managed File Transfer Command Center User's Guide*.

Parameter	Description	Default	Required
`Name (dn)`	Specifies the 1-to-64-character department name.	None	Yes

Sample `RemoveDepartment` Command

This command removes the GM426 department from the database.

```
java cfcc.CFAdmin a:RemoveDepartment DepartmentId:GM426
```

RetrieveAllDepartments

The `RetrieveAllDepartments` command action is used to display all departments defined to the system.

To use the `RetrieveAllDepartments` action command, you must be a super administrator. For more information, see "Delegated Administration" of TIBCO Managed File Transfer Internet Server User's Guide.

No parameters are supported for this command action.

Sample `RetrieveAllDepartments` Command

This command displays all parameters for all departments defined in the database.

```
java cfcc.CFAdmin a:RetrieveAllDepartments
```

UpdateDepartment

The `UpdateDepartment` command action is used to update a department in the system.

To use the `UpdateDepartment` action command, you must be a super administrator. For more information, see "Delegated Administration" of *TIBCO Managed File Transfer Command Center User's Guide*.

Parameter	Description	Default	Required
`Description (d)`	Specifies the 1-to-64-character description of this department. If the description contains embedded spaces, the entire description must be enclosed in double quotation marks ("").	None	No
`Name (dn)`	Specifies the 1-to-64-character department name.	None	Yes

Sample `UpdateDepartment` Command

This command updates the GM426 department in the database.

```
java cfcc.CFAdmin a:UpdateDepartment Name:GA426 Description:"General Administration
– section 426"
```

Group Commands

The group commands are used to define, list, update, delete, and assign membership of group records in the system.

Action	Description
AddGroup	Defines a group.
UpdateGroup	Updates a group.
AddUserToGroup	Adds a user to a group.
GetGroup	Displays a group.
RemoveGroup	Deletes a group.
RetrieveAllGroups	Displays all groups.
RetrieveAllGroupsForUser	Displays groups that the user is a member of.
RetrieveAllUsersInGroup	Displays all users in a group.
RemoveUserFromGroup	Deletes a user from a group.

AddGroup

The `AddGroup` command action is used to define a group.

TIBCO MFT Internet Server has a facility to group user IDs together. These groups can be all users at a specific location, business unit, or whatever grouping you chose. Create a group before the users can be grouped together.

To use the `AddGroup` action command, you must have UpdateGroupRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`Department (dpt)`	Specifies the department of this group. This value is ignored for department administrators.	None	No
`Description (d)`	Specifies the 1-to-64-character description of this group. If the description contains embedded spaces, the entire description must be enclosed in double quotation marks (").	None	No
`GroupId (gid)`	Specifies the 1-to-64-character group ID.	None	Yes
`Visibility (vsb)`	The visibility of this group. The valid values are as follows: • public • private	private	Yes

Sample `AddGroup` Command

This command adds a group.

```
java cfcc.CFAdmin a:AddGroup GroupId:Store68 Description:"68 – Plano, TX"
```

UpdateGroup

The `UpdateGroup` action command is used to update a group.

TIBCO MFT Internet Server has a facility to group user IDs together. Before users can be grouped together, a group has to be created. A group can be all users at a specific location, business unit, or whatever grouping you choose.

To use the `UpdateGroup` action command, a user must have UpdateGroupRight. For more information, see the AddUserToRole command.

Parameter	Description	Default	Required
`Department (dpt)`	Specifies the department of a group. This value is ignored for department administrators.	None	No
`Description (d)`	Specifies the description of this group in 1-to-64-characters. If the description contains embedded blanks the whole description should be enclosed in double quotation marks (" ").	None	No
`GroupId (gid)`	Specifies the group ID in 1-to-64-characters.	None	Yes
`Visibility (vsb)`	Specifies the group’s visibility. The valid values are public and private.	private	Yes

Sample UpdateGroup Command

The command below is a sample of updating a group.

```
java cfcc.CFAdmin a:UpdateGroup GroupId:Store67 Description:"67 – Plano, TX"
```

AddUserToGroup

The `AddUserToGroup` command action is used to add a user to a group.

To use the `AddUserToGroup` action command, you must have UpdateGroupRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`GroupId (gid)`	Specifies the 1-to-64-character group ID.	None	Yes

Parameter	Description	Default	Required
`UserId (uid)`	Specifies the 1-to-64-character user ID of the user to be assigned to this group.	None	Yes

Sample `AddUserToGroup` Command

This command adds the user Marketing008 to the Marketing group.

```
java cfcc.CFAdmin a:AddUserToGroup GroupId:Marketing UserId:Marketing008
```

GetGroup

The `GetGroup` command action is used to display a group defined to the system.

To use the `GetGroup` action command, you must have UpdateGroupRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`GroupId (gid)`	Specifies the 1-to-64-character group ID.	None	Yes

Sample `GetGroup` Command

This command displays the parameters for the TRANSFER01 group.

```
java cfcc.CFAdmin a:GetGroup GroupId:TRANSFER01
```

RemoveGroup

The `RemoveGroup` command action is used to delete a group from the system.

To use the `RemoveGroup` action command, you must have UpdateGroupRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`GroupId (gid)`	Specifies the 1-to-64-character group ID.	None	Yes

Sample `RemoveGroup` Command

This command removes the GM426 group from the database.

```
java cfcc.CFAdmin a:RemoveGroup GroupId:GM426
```

RetrieveAllGroups

The `RetrieveAllGroups` command action is used to display all groups defined to the system.

To use the `RetrieveAllGroups` action command, you must have UpdateGroupRight. For more information, see AddUserToRole.

No parameters are supported for this command action.

Sample `RetrieveAllGroups` Command

This command displays all parameters for all groups defined to the database.

```
java cfcc.CFAdmin a:RetrieveAllGroups
```

RetrieveAllGroupsForUser

The `RetrieveAllGroupsForUser` command action is used to display a list of all the groups that a specific user ID is a member of.

To use the `RetrieveAllGroupsForUser` action command, you must have UpdateGroupRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`UserId (uid)`	Specifies the 1-to-64-character user ID of the user whose group memberships are to be displayed.	None	Yes

Sample `RetrieveAllGroupsForUser` Command

This command displays the parameters for each group where the specified user is defined.

```
java cfcc.CFAdmin a:RetrieveAllGroupsForUser UserId:FT61825
```

RetrieveAllUsersInGroup

The `RetrieveAllUsersInGroup` command action is used to display a list of all the users that are a member of a specific group.

To use the `RetrieveAllUsersInGroup` action command, you must have UpdateGroupRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`GroupId (gid)`	Specifies the 1-to-64-character group ID.	None	Yes

Sample `RetrieveAllUsersInGroup` Command

This command displays all the parameters for each user in the specified group.

```
java cfcc.CFAdmin a:RetrieveAllUsersInGroup GroupId:TeleSales
```

RemoveUserFromGroup

The `RemoveUserFromGroup` command action is used to remove an user from a group.

To use the `RemoveUserFromGroup` action command, you must have UpdateGroupRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`GroupId (gid)`	Specifies the 1-to-64-character group ID.	None	Yes
`UserId (uid)`	Specifies the 1-to-64-character user ID of the user to be removed from the group.	None	Yes

Sample `RemoveUserFromGroup` Command

This command removes the user Investor248 from the Stockholders group.

```
java cfcc.CFAdmin a:RemoveUserFromGroup GroupId:Stockholders UserId:Investor248
```

PGP Public Keys

The PGP Public Key commands are used to add/create, list, retrieve, update, and delete MFT PGP Public Key definitions. PGP public keys are used to verify signatures for incoming requests and encrypt outgoing data.

Action	Description
AddPGPPublicKey	Adds or Creates a PGP public key.
GetPGPPublicKey	Displays a PGP public key.
RetrievePGPPublicKeys	Displays PGP public keys based on selection criteria.
DeletePGPPublicKey	Deletes a PGP public key.
UpdatePGPPublicKey	Updates a PGP public key.

Add PGP Public Key

The `AddPGPPublicKey` command action is used to add a PGP Public Key to . To use the `AddPGPPublicKey` action command, you must have UpdatePGPKeyRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
KeyType (kt)	Key Type: : User - Key is for a user : Server - Key is for a server	N/A	Y
KeyName (n)	Key Name	N/A	Y
Default (df)	Default Key	N	N
Status (st)	Key Status when added: : Enabled : Disabled	Enabled	N
CreateAdd (ca)	Create Add Flag: Create - Error if public key exists for user or server Add - Add even if key exists for user or server	Create	Y
PublicKeyFileName (pkfn)	Name of the file that contains the PGP Public Key	N/A	Y

Delete PGP Public Key

The `DeletePGPPublicKey` command action is used to delete a PGP Public Key. To use the DeletePGPPublicKey action command, you must have UpdatePGPKeyRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
KeyId (id)	Key Type id: Defines the 12-digit KeyID associated with the PGP Public key. Example: K205G0000BC9	N/A	Y

Get PGP Public Key

The `GetPGPPublicKey` command action is used to list the details of one PGP Public Key. To use the `GetPGPPublicKey` action command, you must have UpdatePGPKeyRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
KeyId (id)	Key Type id: Defines the 12-digit KeyID associated with the PGP Public key. Example: K205G0000BC9	N/A	Y

Retrieve PGP Public Keys

The `RetrievePGPPublicKeys` command action is used to retrieve (i.e. List) PGP Public Keys. To use the `RetrievePGPPublicKeys` action command, you must have UpdatePGPKeyRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
KeyType (kt)	Key Type:	Both	N
KeyName (n)	Key Name	N/A	N
Status (st)	Key Status when added:	N/A	N

Parameter	Description	Default	Required
EncryptionKeyId (ekid)	Encryption Key Type id: Defines the Encryption Key Id of the key to be retrieved.	N/A	N

Update PGP Public Key

The `UpdatePGPPublicKey` command action is used to update a PGP Public Key. To use the `UpdatePGPPublicKey` action command, you must have UpdatePGPKeyRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
KeyId (id)	Key Type id: Defines the 12-digit KeyID associated with the PGP Public key. Example: K205G0000BC9	N/A	Y
Default (df)	Default Key	N	N
Status (st)	Key Status when added: : Enabled : Disabled	N/A	N
PublicKeyFileName (pkfn)	Name of the file that contains the PGP Public Key	N/A	Y
KeyType (kt)	Key Type : : User - Key is for a user : Server - Key is for a server	N/A	Y
CreateAdd (ca)	Creates an Add Flag:	N/A	Y

Parameter	Description	Default	Required
	: Create - Error if public key exists for user or server : Add - Add even if key exists for user or server Create		
EncryptionKeyId (ekid)	Encryption Key Type id: Defines the Encryption Key Id of the key to be retrieved.	N/A	N

Protocol Public Key Commands

The Protocol Public Key commands are used to add/create, list, retrieve, update, and delete MFT Protocol Public Key definitions. Protocol public keys are used for validating client certificate/key authentication requests (KeyType=User) or to validate certificates and keys for connections to target servers (KeyType=Server).

Action	Description
AddProtocolPublicKey	Adds or Creates a Protocol public key.
GetProtocolPublicKey	Displays a Protocol public key.
RetrieveProtocolPublicKeys	Displays Protocol public keys based on selection criteria.
DeleteProtocolPublicKey	Deletes a Protocol public key.
UpdateProtocolPublicKey	Updates a Protocol public key.

Add Protocol Public Key

The `AddProtocolPublicKey` command action is used to add a Protocol Public Key to . To use the `AddProtocolPublicKey` action command, you must have UpdatePublicKeyRight.

For more information, see AddUserToRole.

Parameter	Description	Default	Required
CreateAdd (ca)	Create Add Flag: : Create - Error if public key exists for user or server : Add - Add even if key exists for user or server	Create	Y
Description (desc)	Description of the protocol public key to be added	N/A	N
KeyName (n)	Key Name	N/A	Y
KeyType (kt)	Key Type: : User - key is for a user : Server - key is for a server	N/A	Y
Protocol (pkt)	Protocol associated with the key: : FTP : HTTPS : PLATFORM : SSH	N/A	Y
PublicKey (pk)	Public key to be added.	N/A	Y *1*2
PublicKeyFileName (pkfn)	Name of the file that contains the Protocol Public Key	N/A	Y *1*2
Status (st)	Key Status when added:	Enabled	N

Parameter	Description	Default	Required
	: Enabled : Disabled		

- *1 - Either PublicKeyFilename or Public Key is required.
- *2 - PublicKeyFileName and PublicKey parameters are mutually exclusive.

Delete Protocol Public Key

The `DeleteProtocolPublicKey` command action is used to delete a Protocol Public Key. To use the `DeleteProtocolPublicKey` action command, you must have UpdatePublicKeyRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
KeyId (id)	Key Type id: Defines the 12-digit KeyID associated with the Protocol Public key. Example: K205G0000123	N/A	Y

Get Protocol Public Key

The `GetProtocolPublicKey` command action is used to list the details of one Protocol Public Key. To use the `GetProtofcolPublicKey` action command, you must have UpdatePublicKeyRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
KeyId (id)	Key Type id: Defines the 12-digit KeyID associated with the Protocol Public key. Example: K205G0000123	N/A	Y

Retrieve Protocol Public Keys

The `RetrieveProtocolPublicKeys` command action is used to retrieve Protocol Public Keys. To use the `RetrieveProtocolPublicKeys` action command, you must have UpdatePublicKeyRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
Protocol (pkt)	Protocol associated with the key: : FTP : HTTPS : PLATFORM : SSH	N/A	N
KeyType (kt)	Key Type: : User - key is for a user : Server - key is for a server : Both: retrieve User and Server keys	Both	N
KeyName (n)	Key Name	N/A	N
Status (st)	Key Status when added: : Enabled : Disabled	N/A	N
EncryptionKeyId (ekid)	Encryption Key Type id: Defines the Encryption Key Id of the key to be retrieved.	N/A	N

Update Protocol Public Key

The `UpdateProtocolPublicKey` command action is used to update a protocol public Key. To use the `UpdateProtocolPublicKey` action command, you must have UpdatePublicKeyRight. For more information, see AddUserToRole.

Parameter	**Description**	**Default**	**Required**
CreateAdd (ca)	Creates or adds a key. Valid values are as follows: • Create - Creates a key. Shows an error if a public key exists for a user or a server. • Add - Adds a key even if a key exists for a user or a server.	Create	Y
Description (desc)	Description of the protocol public key to be updated.	N/A	N
KeyId (id)	Key type ID defines the 12-digit key ID associated with the protocol public key. Example: K205G0000BC9	N/A	Y
PublicKey (pk)	Public key to be added. The key should be without line separators and without spaces.	N/A	Y*1
PublicKeyFileName (pkfn)	Name of the file that contains the protocol public key	N/A	Y*1
Status (st)	Key status when added: • Enabled • Disabled	N/A	N

Parameter	Description	Default	Required
EncryptionKeyId (ekid)	Encryption Key Type id defines the Encryption Key Id of the key to be retrieved.	N/A	N
KeyName(n)	Key Name	N/A	N
KeyType(kt)	Key Type: : User - key is for a user : Server - key is for a server : Both - Retrieve User and Server keys	N/A	N
Protocol	Protocol associated with the key: : FTP : HTTPS : PLATFORM : SSH	N/A	Y

- 1 - `PublicKeyFileName` and `PublicKey` parameters are mutually exclusive.

Sync LDAP Authenticator Commands

The Sync LDAP Authenticator commands are used to add/create, list, retrieve, update, and delete MFT Protocol Public Key definitions. Protocol public keys are used for validating client certificate/key authentication requests (KeyType=User) or to validate certificates and keys for connections to target servers (KeyType=Server).

Action	Description
SyncAll	Syncs all enabled Authenticators.
SycAuth	Syncs one enabled LDAP Authenticator.
SyncUser	Syncs one LDAP User.

Sync All

The `SyncAll` command action is used to sync all enabled LDAP Authenticators. To use the `SyncAll` action command, you must have AdministratorRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
No parameters are required			

Sync Auth

The `SyncAuth` command action is used to sync one enabled LDAP Authenticator. To use the `SyncAuth` action command, you must have AdministratorRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
SyncName (sn)	Authenticator Name to Sync	N/A	Y

Sync User

The `SyncUser` command action is used to sync LDAP User. To use the `SyncUser` action command, you must have AdministratorRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
SyncName (sn)	User Id to Sync	N/A	Y

Role Commands

The role commands are used to define, list, delete, and assign rights to users in the system.

Action	Description
AddUserToRole	Adds a right to a user.
GetRole	Displays a right.
RetrieveAllRoles	Displays all rights.
RetrieveAllRolesForUser	Displays the rights assigned to a user.
RetrieveAllUsersInRole	Displays users that have a specific right.
RemoveUserFromRole	Removes a right from a user.

AddUserToRole

The `AddUserToRole` command action is used to assign a user to a role.

The roles define the rights that a user has to perform file transfers and administrative functions.

Note: The word role in this section is referred to as right in the rest of the manual.

To use the `AddUserToRole` action command, you must have UpdateTransferUserRight.

Parameter	Description	Default	Required
`RoleId (rid)`	Specifies the right to be given to the user as defined in the table below.	None	Yes
`UserId (uid)`	Specifies the 1-to-64-character user ID. This is the name of the user to whom you want to assign rights.	None	Yes

The following table lists the roles and their supported functions:

Right	Description	Description using Delegated Administration
AdministratorRight	Allows a user to perform all administrative functions within the system. This right does not include TransferRight or FTTransferRight or any functions that correspond to these rights.	Allows a user to perform all administrative functions within his own department and the departments that the user can manage. This right does not include TransferRight or FTTransferRight or any functions that correspond to these rights. The department administrator cannot update servers or server Credentials unless given UpdateServerRight and UpdateServerCredentialRight.
DBReportRight	Allows a user to login and view and generate the TIBCO MFT Command Center's database reports through the **Reports > Database Reports** option.	Allows a user to login and view and generate TIBCO MFT Command Center's database reports through the **Reports > Database Reports** option.

Right	Description	Description using Delegated Administration
DeleteAuditRight	Allows any user to delete an audit record.	Allows any user to delete audit records. Department checking will not be done.
ExecuteSchedulerJobRight	Allows a user to view and execute a job through the **Execute Now** button and Platform Server command. **Note:** This right does not allow to update a job.	Allows a user to view and execute a job through the **Execute Now** button and Platform Server command. **Note:** This right does not allow to update a job.
HelpDeskRight	Allows a user to change another user's password, turn on and off the disable flag for a user as well as turn on and off the lock flag for a user.	Allows a user to change another user's password, turn on and off the disable flag for a user as well as turn on and off the lock flag for a user.
OnDemandTransferRight	Allows a user the ability to use the desktop client **Site Manager** menu item to set up and conduct on-demand transfers.	Allows a user the ability to use the desktop client **Site Manager** menu item to set up and conduct on-demand transfers.
TransferRight	Allows a user to execute TIBCO MFT Command Center's Internet transfers.	Allows a user to execute TIBCO MFT Command Center's Internet Transfers.
UpdateAlertRight	Allows a user to update alert records and view alerts that have occurred.	Allows a user to update alert records and view alerts that have occurred.

Right	Description	Description using Delegated Administration
UpdateCheckpointRight	Allows a user to access the TIBCO MFT Internet Server checkpoints web page and delete checkpoints taken.	Allows a user to access the TIBCO MFT Internet Server checkpoints web page and delete checkpoints taken.
UpdateFTTransferRight	Allows a user to update platform transfer defined through the **Management > Platform Transfers > Manage Platform Transfers** option. This right will not allow the user to execute platform transfers.	Allows a user to update platform transfer defined through the **Management > Platform Transfers > Manage Platform Transfers** option. This right will not allow the user to execute platform transfers.
UpdateGroupRight	Allows a user to view and update TIBCO MFT Command Center's group records.	Allows a user to view and update TIBCO MFT Command Center's group records.
UpdateOnDemandRight	Allows a user the ability to add or remove the on-demand sites.	Allows a user the ability to add or remove the on-demand sites assigned to other users within their department.
UpdatePGPKeyRight	Allows a user to add and manage the configurations of PGP public keys.	Allows a user to add and manage the configurations of PGP public keys.
UpdatePGPSystemKeyRight	Allows a user to add and manage the configurations of TIBCO MFT Command Center's PGP system keys.	Allows a user to add and manage the configurations of TIBCO MFT Command Center's PGP system keys.

Right	Description	Description using Delegated Administration
UpdatePublicKeyRight	Allows a user to add and manage the configurations of FTPS, SFTP, Platform Server, and HTTPS public keys.	Allows a user to add and manage the configurations of FTPS, SFTP, Platform Server, and HTTPS public keys.
UpdateSchedulerRight	Allows a user to add and manage the Scheduler jobs in TIBCO MFT Command Center.	Allows a user to add and manage the Scheduler jobs in TIBCO MFT Command Center.
UpdateServerCredentialRight	Allows a user to view or update TIBCO MFT Command Center's server credential records.	Allows a user to view or update TIBCO MFT Command Center's server credential records.
UpdateServerRight	Allows a user to view or update TIBCO MFT Command Center's server records.	Allows a user to view or update TIBCO MFT Command Center's server records in his own department. New servers cannot be added.
UpdateSessionRight	Allows a user to view and delete active user sessions.	Allows a user to view and delete active user sessions.
UpdateSystemKeyRight	Allows a user to add and manage the configurations of AS2, FTP, SFTP, Platform SSL, HTTPS and SAML system keys through the **Administration > Protocol Keys > System Keys** option.	Allows a user to add and manage the configurations of AS2, FTP, SFTP, Platform SSL, HTTPS and SAML system keys through the **Administration > Protocol Keys > System Keys** option. Allows a user to add and manage the configurations of

Right	Description	Description using Delegated Administration
	Allows a user to add and manage the configurations of Kerberos KeyTab files through the **Administration > Protocol Keys > Kerberos KeyTabs** option.	Kerberos KeyTab files through the **Administration > Protocol Keys > Kerberos KeyTabs** option.
UpdateTransferDefinitionRight	Allows a user to view and update TIBCO MFT Command Center's Internet transfer definitions.	Allows a user to view and update TIBCO MFT Command Center's Internet transfer definitions.
UpdateTransferUserRight	Allows a user to view and update TIBCO MFT Command Center's user records. Only TransferRight and OnDemandTransferRight can be given to a user unless you are an administrator. The super administrator can assign any right to a user. **Note:** When assigning this right to a user, you must also assign either ViewGroupRight or UpdateGroupRight.	Allows a user to view and update TIBCO MFT Command Center's user records. Only TransferRight and OnDemandTransferRight can be given to a user unless you are an administrator. The department administrator can assign any rights to a user within his own department, except UpdateServerRight and UpdateServerCredentialRight. **Note:** When assigning this right to a user, you must also assign either ViewGroupRight or UpdateGroupRight.

Right	Description	Description using Delegated Administration
ViewAlertRight	Allows a user to view alert records and view alerts that have occurred.	Allows a user to view alert records and view alerts that have occurred.
ViewAuditRight	Allows a user to view audit records and update the audit search filters.	Allows a user to view audit records and update the audit search filters.
ViewCheckpointRight	Allows a user to access the TIBCO MFT Command Center's Internet Checkpoints page and view checkpoints taken.	Allows a user to access the TIBCO MFT Command Center's Internet Checkpoints page and view checkpoints taken.
ViewFTTransferRight	Allows a user to view platform Transfers defined through the **Management > Platform Transfers > Manage Platform Transfers** option. This right will not allow the user to add, update, or execute platform transfers.	Allows a user to view platform Transfers defined through the **Management > Platform Transfers > Manage Platform Transfers** option. This right will not allow the user to add, update, or execute platform transfers.
ViewGroupRight	Allows a user to view TIBCO MFT Command Center's group records.	Allows a user to view TIBCO MFT Command Center's group records.
ViewOnDemandRight	Allows a user to view TIBCO MFT Command Center's on-demand site records.	Allows a user to view TIBCO MFT Command Center's on-demand site records.
ViewPCILogRight	Allows the user to view Admin change reports.	Allows the user to view Admin change reports.

Right	Description	Description using Delegated Administration
ViewPGPKeyRight	Allows a user to view PGP public keys.	Allows a user to view PGP public keys.
ViewPublicKeyRight	Allows a user to view TIBCO MFT Command Center FTP, SSH, HTTPS public keys.	Allows a user to view TIBCO MFT Command Center FTP, SSH, HTTPS public keys.
ViewSchedulerRight	Allows a user to view the scheduled transactions.	Allows a user to view the scheduled transactions.
ViewServerCredentialRight	Allows a user to view TIBCO MFT Command Center's server profile records.	Allows a user to view TIBCO MFT Command Center's server profile records.
ViewServerRight	Allows a user to view TIBCO MFT Command Center's server records.	Allows a user to view TIBCO MFT Command Center's server records.
ViewSessionRight	Allows a user to view active user sessions.	Allows a user to view active user sessions.
ViewTransferDefinitionRight	Allows a user to view TIBCO MFT Command Center's Internet Server transfer records.	Allows a user to view TIBCO MFT Command Center's Internet Server transfer records.
ViewUserRight	Allows a user to view TIBCO MFT Command Center's user records and the rights associated with those users.	Allows a user to view TIBCO MFT Command Center's user records and the rights associated with those users.

Sample `AddUserToRole` Command

This command gives the user mftuser1 the TransferRight role.

```
java cfcc.CFAdmin a:AddUserToRole UserId:mftuser1 RoleId:TransferRight
```

GetRole

The `GetRole` command action is used to display information about a role.

The TIBCO MFT Internet Server roles define the rights that a user has to perform file transfers and administrative functions.

To use the `GetRole` action command, you must have UpdateTransferUserRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`RoleId (rid)`	Specifies the 1-to-64-character role name. This is the name of the role that you want to display.	None	Yes

Sample `GetRole` Command

This command displays the information about the TransferRight role.

```
java cfcc.CFAdmin a:GetRole RoleId:TransferRight
```

RetrieveAllRoles

The `RetrieveAllRoles` command action is used to display a list of all roles that have been defined.

The roles define the rights that an user has to perform file transfers and administrative functions.

To use the `RetrieveAllRoles` action command, you must have UpdateTransferUserRight. For more information, see AddUserToRole.

No parameters are supported for this command action.

Sample RetrieveAllRoles Command

This command displays the information about all defined roles.

```
java cfcc.CFAdmin a:RetrieveAllRoles
```

RetrieveAllRolesForUser

The RetrieveAllRolesForUser command action is used to display a list of all roles that a user has been granted access to.

The roles define the rights that an user has to perform file transfers and administrative functions.

To use the RetrieveAllRolesForUser action command, you must have UpdateTransferUserRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
UserId (uid)	Specifies the 1-to-64-character user ID. This is the name of the user that you want to display roles for.	None	Yes

Sample RetrieveAllRolesForUser Command

This command displays the information about all roles defined for a user.

```
java cfcc.CFAdmin a:RetrieveAllRolesForUser UserId:user1
```

RetrieveAllUsersInRole

The RetrieveAllUsersInRole command action is used to display a list of all users granted rights to a role.

The roles define the rights that an user has to perform file transfers and administrative functions.

To use the RetrieveAllUsersInRole action command, you must have UpdateTransferUserRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
RoleId (rid)	Specifies the 1-to-64-character role name. This is the name of the role that you want to display all users granted access to.	None	Yes

Sample RetrieveAllUsersInRole Command

This command displays the user definition for all users with rights to the TransferRight role.

```
java cfcc.CFAdmin a:RetrieveAllUsersInRole RoleId:TransferRight
```

RemoveUserFromRole

The RemoveUserFromRole command action is used to remove a user from a role.

The roles define the rights that an user has to perform file transfers and administrative functions.

To use the RemoveUserFromRole action command, you must have UpdateTransferUserRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
RoleId (rid)	Specifies the 1-to-64-character role name. This is the name of the role that you want to remove the user rights to.	None	Yes
UserId (uid)	Specifies the 1-to-64-character user ID. This is the name of the	None	Yes

Parameter	Description	Default	Required
	user that you want to remove rights from a role.		

Sample `RemoveUserFromRole` Command

This command removes the user mftuser1 from the UpdateTransferDefinitionRight role.

```
java cfcc.CFAdmin a:RemoveUserFromRole UserId:mftuser1 RoleId:TransferRight
```

Server Commands

The server commands are used to define, list, update, and delete MFT Server definitions in the system.

Action	Description
AddServer	Creates a server
GetServer	Displays a server
RetrieveAllServers	Displays all servers
RemoveServer	Deletes a server
UpdateServer	Updates a server

AddServer

The `AddServer` command action is used to add a node definition to TIBCO MFT Internet Server.

The node definition contains information about the remote system. You only have to define node definitions when you are connecting to a remote system. If you are storing files locally, you do not have to define node definitions.

To use the `AddServer` action command, you must have UpdateServerRight. For more information, see AddUserToRole.

In the following table, parameters for this command are provided in alphabetical order.

> **Note:** The parameters provided in this table are also used for the `UpdateServer` command.

Parameter	Description	Default	Required
`AmazonAWSAuthentication (s3awsa)`	Specifies how to authenticate to Amazon AWS when the server type is Amazon S3. The values supported for this parameter are as follows: • SK: secret key • EC2: EC2 metadata • SAML: SAML IDP form	None	No
`AmazonS3NumberOfUploadBuffers (ubc)`	Specifies the Amazon S3 number of upload buffers (1-10). It should be less than or equal to the number of upload threads.	None	No
`AmazonS3AssumeRole (sar)`	Specifies the ARN of the role that is assumed when accessing an S3 Bucket. This option is only supported when the *AmazonAWSAuthentication* parameter is set to *secret key* or *EC2 metadata.* When Assume Role is defined, you must set the Amazon S3 Region parameter to the region of the S3 Bucket.	None	No

Parameter	Description	Default	Requir ed
	Note: Amazon IAM definitions must be configured to allow the user to assume the defined role.		
`AmazonS3Vendor (s3v)`	Specifies the vendor associated with the S3 Storage. The following values are supported for this parameter: • Amazon AWS - Defines that the S3 storage is Amazon S3 Storage. • S3 Compatible - Defines that the S3 storage is for a 3rd party Amazon compatible server. **Note:** When **S3 Compatible** option is selected, you must define the VPC Endpoint Interface DNS to point to the S3 Compatible storage server DNS name.	Default	Yes
`AmazonS3NumberOfUploadThreads (utc)`	Specifies the Amazon S3 number of upload threads (1-10).	None	No
`AmazonS3Region (reg)`	Specifies the Amazon S3 region. The following values are supported for this parameter: • GovCloud • US_EAST_1 • US_EAST_2	None	No

Parameter	Description	Default	Required
	• US_WEST_1 • US_WEST_2 • EU_WEST_1 • EU_WEST_2 • EU_CENTRAL_1 • AP_SOUTH_1 • AP_SOUTHEAST_1 • AP_SOUTHEAST_2 • AP_NORTHEAST_1 • AP_NORTHEAST_2 • SA_EAST_1 • CN_NORTH_1 • CA_CENTRAL_1		
`AmazonS3UploadChunkSize (ucs)`	Specifies the Amazon S3 upload chunk size.	None	No
`AmazonSAMLIDPFormJSONFile (s3sifj)`	The JSON file that is used to connect to the SAML IDP server to extract credentials.	None	No
`AmazonServerSideEncryption (s3encry)`	Specifies the Amazon server side encryption. The values supported for this parameter are as follows: • S: Amazon S3-Managed Keys • K: AWS KMS-Managed Keys	None	No
`AS2EncryptPublicKeyFile (as2epkfile)`	Specifies the AS2 encrypt key file.	None	No

Parameter	Description	Default	Required
`AS2EncryptSystemKey (as2eskey)`	Specifies the AS2 encrypt system key.	None	No
`AS2HTTPSPublicKeyFile (as2httpspkfile)`	Specifies the AS2 HTTPS public file path.	None	No
`AS2HTTPSSystemKey (as2httpskey)`	Specifies the AS2 HTTPS system key.	None	No
`AS2IncomingEncrAlg (as2iealg)`	Specifies the AS2 incoming encrypt algorithm. The values supported for this parameter are as follows: • ALL • 3DES • AES • NONE	None	No
`AS2IncomingSignAlg (as2isalg)`	Specifies the AS2 incoming signing algorithm. The values supported for this parameter are as follows: • ALL • SHA1 • SHA-256 • SHA-384 • SHA-512 • MDS • NONE	None	No
`AS2IncomingUserID (as2iuid)`	Specifies the AS2 incoming user ID.	None	No

Parameter	Description	Default	Required
`AS2LocalId (as2lid)`	Specifies the AS2 local server ID.	None	No
`AS2OutgoingCompAlg (as2ocalg)`	Specifies the AS2 outgoing compress algorithm. The values supported for this parameter are as follows: • ZLIB • NONE	None	No
`AS2OutgoingDataType (as2odt)`	Specifies the AS2 outgoing data type. The values supported for this parameter are as follows: • Application/EDI-X12 • Application/EDIFACT • Application/EDI-consent • Application/octet-stream	None	No
`AS2OutgoingEncrAlg (as2oealg)`	Specifies the AS2 outgoing encrypt algorithm. The values supported for this parameter are as follows: • 3DES • AES • NONE	None	No
`AS2OutgoingMDNReceipt (as2omdnr)`	Specifies the AS2 outgoing MDN receipt. The values supported for this parameter are as follows: • S: sync • A: async • N: no receipt	None	No

Parameter	Description	Default	Requir ed
`AS2OutgoingMDNSignatureAlg (as2omdnsalg)`	Specifies the AS2 outgoing MDN signature algorithm. The values supported for this parameter are as follows: • SHA1 • SHA-256 • SHA-384 • SHA-512 • MDS • NONE	None	No
`AS2OutgoingSignAlg (as2osalg)`	Specifies the AS2 outgoing signing algorithm. The values supported for this parameter are as follows: • SHA1 • SHA-256 • SHA-384 • SHA-512 • MDS • NONE	None	No
`AS2OutgoingStreamingMode (as2osm)`	Specifies the AS2 outgoing streaming mode. The values supported for this parameter are as follows: • Y: yes • N: no	None	No
`AS2OutgoingTimeout (as2oto)`	Specifies the AS2 outgoing timeout.	None	No

Parameter	Description	Default	Required
	The values supported for this parameter are as follows: • 0-99999 (seconds)		
`AS2PartnerId (as2pid)`	Specifies the AS2 partner ID.	None	No
`AS2SignPublicKeyFile (as2spkfile)`	Specifies the AS2 signing public key file.	None	No
`AS2SignSystemKey (as2sskey)`	Specifies the AS2 sign system key.	None	No
`AVDownloadScanFileRegex (avdsfr)`	Specifies the antivirus download scan file REGEX.	None	
`AVMode (avmode)`	Specifies the different antivirus modes. The values supported for this parameter is as follows: • S: Streaming • F: Store and Forward • D: Default	None	
`AVTransferScanDirection (avtsd)`	Specifies the antivirus transfer scan direction. The values supported for this parameter is as follows: • U: Upload • D: Download • B: Both	None	
`AVUploadScanFileRegex (avusfr)`	Specifies the antivirus upload scan file REGEX.	None	
`AzureAccountName (aan)`	Specifies the Azure account name.	None	

Parameter	Description	Default	Required
	This parameter is only applicable when the server type `U=Microsoft Azure` is defined.		
`AzureNumberOfUploadBuffer (nub)`	Specifies the Azure number of upload buffers (1-10). It should be less than or equal to the number of upload threads.	None	No
`AzureNumberOfUploadThreads (nut)`	Specifies the number of upload threads. The values supported for this parameter are as follows: • 1-10	None	No
`AzureRetrieveModified (arm)`	Specifies whether to modify retrieve. The values supported for this parameter are as follows: • Y: yes • N: no	None	No
`AzureStorageType (ast)`	Specifies the type of storage. The values supported for this parameter are as follows: • B: Microsoft Azure Storage Blob • F: Microsoft Azure Storage File • G: Microsoft Azure Data Lake Storage Gen2	None	No
`AzureTenantID (atid)`	Specifies the Azure tenant ID. This is only applicable when server type is defined as either of the	None	

Parameter	Description	Default	Required
	following: - U: Microsoft Azure - P: SharePoint		
`AzureUploadChunkSize (aucs)`	Specifies the upload chunk size.	None	No
`CheckServerStatus (cstat)`	Specifies whether to check the server status. The values supported for this parameter are as follows: - Y: yes - N: no. The default is N.	N	No
`CheckServerStatusOn (cstaton)`	Specifies whether to check if the server status is on. The values supported for this parameter are as follows: - Command Center - Internet Server host name	None	No
`CollectHistory (ch)`	Specifies the collection history.	None	No
`CollectInterval (ci)`	Specifies the collection interval in minutes when collection is done. For TIBCO® Managed File Transfer Platform Servers only.	None	No
`CollectType (ctt)`	Specifies the type of collection to be done. The values supported for this parameter are as follows: - I: initiator - R: responder	None	No

Parameter	Description	Default	Required
	- B: both		
`CompressType (ct)`	Specifies the default compression that is performed between the web client and the server. The values supported for this parameter are as follows: - N: no compression. - Y: use compression. **Note:** This field defines the compression between the web client and the server; not between the server and TIBCO MFT Platform Server. Compression between the server and TIBCO MFT Platform Server is not supported. If this parameter is undefined, the compression flag defined in the configuration is used.	None	No
`ConnectionSecurityType (ftpcst)`	This indicates the security for using a connection type of FTP. The values supported for this parameter are as follows: - None: the FTP connection is unsecure. - Explicit SSL: an unsecure connection is made to the remote FTP node, followed by a negotiation for SSL security. The remote server must be listening on an unsecure port.	None	No

Parameter	Description	Default	Required
	- Implicit SSL: an SSL connection is made to the remote FTP node. The remote server must be listening on an SSL port.		
`Context (ctxt)`	Specifies the Internet Server options context.	None	No
`CRCChecking (crcc)`	Specifies whether to use CRC checking. The values supported for this parameter are as follows: - Yes - No	None	Yes
`CustomServerConfigurationDataFile (cscd)`	Enter any data that should be passed to the custom interface code. You can pass any text data in this box (such as JSON, XML, or CSV). Validation is not performed on the data in this box. You must, therefore, format the data correctly. Up to 65535 bytes of data is allowed in the box. The custom server interface allows you to enter three tokens to pass credentials associated with this server to the custom code. In this way, you do not need to enter clear text passwords into the configuration box. - $(UserId) - passes the clear text user ID to the customer interface code.	None	No

Parameter	Description	Default	Required
	- $(Password) - passes the clear text password to the customer interface code. - $(Domain) - passes the clear text domain name to the customer interface code.		
`CustomServerJavaClassName (csjcn)`	Defines an implementation of the "com.tibco.mft.transfers.custom.CustomTransfer" interface for the custom server. The Java Class Name for the example provided is: `com.example.transfer.CustomXferImpl`	None	No
`DataConnectionType (dct)`	Specifies the connection type for FTP transfers. The values supported for this parameter are as follows: - PORT - PASV - EPRT - EPSV	PORT	No
`DefaultEncryptType (et)`	Specifies the default encryption that is to be performed between the server and the target TIBCO MFT Platform Server node. The values supported for this parameter are as follows: - N: no encryption. - D: for the DES encryption (56	Default	No

Parameter	Description	Default	Required
	bit key). - R: for the Rijndael encryption (256 bit key). **Note:** This encryption is for the TIBCO MFT Platform Server target node only. All communication between the web client is encrypted using SSL encryption. If you want to encrypt data with TIBCO MFT Platform Server, we suggest using rijndael encryption since it is a stronger encryption and is far more efficient.		
`DefaultLTTable (lt)`	Specifies the 1-to-256-byte default local translate table that is used when performing data translation. This parameter must point to the fully qualified translation table file name. This is typically used for ASCII to EBCDIC translation when communicating with TIBCO MFT Platform Server for z/OS and TIBCO MFT Platform Server for IBMi. If the file record has the `LocalTranslationTable` parameter defined, it is used instead.	None	No
`DefaultPass (dp)`	Specifies the 1-to-32-byte default password for communicating with the target TIBCO MFT Platform Server node. This parameter is not used if there	None	No

Parameter	Description	Default	Required
	is a user profile defined for the server definition/user that performs the file transfer. Likewise, it is overridden by the `DefaultUser` parameter on the transfer record. When this parameter is defined, the `DefaultUser` parameter should be defined as well.		
`DefaultRTTable (rt)`	Specifies the 1-to-256-byte default remote translate table that the target TIBCO MFT Platform Server system uses when performing data translation. This parameter must point to the name of the translation table on the remote TIBCO MFT Platform Server system. This parameter is not used if the file record has the `RemoteTranslationTable` parameter defined. When communicating with z/OS this table can be from 1-to-8-characters long and must be enabled at the time the transfer runs.	None	No
`DefaultUser (du)`	Specifies the 1-to-32-byte default user for communicating with the target TIBCO MFT Platform Server node. This parameter is not used if a user profile is defined for the Server definition/user that performs the file transfer. Likewise, it is overridden by the	None	No

Parameter	Description	Default	Required
	DefaultServerUserID parameter on the file record. When this parameter is defined, the DefaultPass parameter should be defined as well.		
DefaultWinDomain (dwd)	Specifies the 1-to-256-byte default NT domain for communicating with the target TIBCO MFT Platform Server Windows node. This parameter is not used if there is a user profile defined for the server definition/user that performs the file transfer. Likewise, it is overridden by the DefaultWinDomain parameter on the transfer record. When this parameter is defined, the DefaultUser and DefaultPass parameters should be defined as well. This parameter is only used when communicating with a Windows environment and defines the domain where the user is defined.	None	No
Department (dpt)	Specifies the department of a node.	None	No
Description (d)	Specifies the description of a node.	None	No
DisableFlag (dis)	Specifies whether the server definition should be disabled. When a server is disabled, it is not available for use by TIBCO MFT Command Center or TIBCO MFT	N	No

Parameter	Description	Default	Required
	Internet Server. The values supported for this parameter are as follows: • N: the server is not disabled. • Y: the server is disabled.		
`DNIManagementPassword (dnimpass)`	Specifies the password for DNI management.	None	No
`DNIManagementPort (dnimp)`	Specifies the DNI management port.	None	No
`DNIManagementUserId (dnimuid)`	Specifies the DNI management user ID.	None	No
`EmailMaximumAttachmentSize (emas)`	Specifies the maximum size of a file that can be attached to the email. The valid value for this parameter is 1-100MB.	10MB	
`EmailSenderEmailAddress (esea)`	Defines the sender's email address.	None	
`EmailSendOnlyToDefinedUsers (esotdu)`	Specifies if email is to be sent to defined users. The values supported for this parameter are as follows: • N: No • Y: Yes	None	
`EmailTrustSMTPTLSCertificates (etsmtptlsc)`	Defines if the SMTPTLS certificates are to be used or trusted. The values supported for this	None	

Parameter	Description	Default	Required
	parameter are as follows: - N: No - Y: Yes		
`EmailUseTLS (eutls)`	Defines whether the email uses TLS or not. The values supported for this parameter are as follows: - N: No - Y: Yes	None	
`FTPCaseSensitive (ftpcs)`	Defines whether access to directories or files on this server are case-sensitive or case-insensitive. The values supported for this parameter are as follows: - Y: Yes - N: No	None	No
`FTPClearCommandChannel (ftpccc)`	Specifies whether to clear the FTP command channel. The values supported for this parameter are as follows: - Y: Yes - N: No	None	No
`FTPExternalIPAddress (ftpeipa)`	Specifies the external IP address.	None	No
`FTPExternalIPAddressFlag (ftpeipf)`	Specifies the FTP external IP address flag. The values supported	None	No

Parameter	Description	Default	Required
	for this parameter are as follows: - Y: Yes - N: No		
`FTPKeepAliveInterval (ftpkai)`	Specifies the keep alive interval. The values supported for this parameter are as follows: - 0-1440 minutes - 0: No keep alive	None	No
`FTPPooling (ftpp)`	Specifies the FTP pooling. The values supported for this parameter are as follows: - Y: Yes - N: No	None	No
`FTPPoolingIdleTimeout (ftppit)`	Specifies the FTP pooling idle time out. The values supported for this parameter are as follows: - 1-60 minutes	None	No
`FTPSystemKey (ftpsk)`	Specifies the FTP system key. The values supported for this parameter are as follows: - D: Default - None - UserId of Key	None	No
`GoogleCloudJsonServiceAccountFile (gcjsafc)`	Defines the JSON service account key associated with the service account. This parameter accepts	None	No

Parameter	Description	Default	Required
	the file path that has the content.		
`GoogleCloudNumberOfUploadBuffers (gcnoub)`	Defines the number of upload buffers. The values supported for this parameter are 1 to 10.	2	No
`GoogleCloudProductType (gcpt)`	Defines the Google Cloud product type. The values supported for this parameter are as follows: • C: cloud storage • B: :big query	None	No
`GoogleCloudUploadChunkSize (gcucs)`	Defines the Google Storage Chunk size in megabytes. The values supported for this parameter are 1 to 64.	5MB	No
`HDFSAuth (hdfsa)`	Specifies the HDFS authentication. The values supported for this parameter are as follows: • 1: Kerberos • 0: simple	None	No
`HDFSPrivKey (hdfspk)`	Specifies the HDFS Kerberos private key.	None	No
`HDFSUserName (hdfsun)`	Specifies the HDFS user name.	None	Np
`HTTPSystemKey (httpsk)`	Specifies the HTTP system key. The values supported for this parameter are as follows: • D: default • None: subnet	None	No

Parameter	Description	Default	Required
	- UserId of key		
`INETServerType (nt)`	Defines the Internet Server type. The values supported for this parameter are as follows: - C: Platform Server - I: Internet Server - J: JMS - F: FTP - L: Local - S: SSH - 2: AS2 - D: HDFS - V: FileShare - H: HTTP - 3: Amazon S3 - U: Microsoft Azure - G: Google Cloud - K: Custom Server - E: Email - M: Mailbox - O: OFTP2 - P: SharePoint	C	No
`IPName (ip)`	The 1-to-64-character IP name. This can be either a machine name or an IP address.	None	No

Parameter	Description	Default	Required
	This defines the TCP information necessary to establish communication with the remote TIBCO MFT Platform Server node. If this parameter is defined incorrectly, is unable to connect to the remote TIBCO MFT Platform Server node.		
`IPPort (pt)`	Specifies the TCP port number that the target server is listening on for incoming connections. This can be any number between 1025 and 65535. This parameter must match the IP port that the remote server (SFTP, FTP, Platform Server) is listening to for incoming connections. If this parameter is defined incorrectly, is unable to connect to the remote TIBCO MFT Platform Server node.	None	No
`KerberosServerIPAddresses (ksipa)`	Specifies the Kerberos server IP addresses using the semicolon (;) as a delimiter among multiple servers.	None	No
`KerberosServerProtocol (ksp)`	Specifies the Kerberos server protocol. The values supported for this parameter are as follows: • 1: TCP • 0: UDP	None	
`MailboxExpirationDays (med)`	Specifies the mailbox expiration	10	

Parameter	Description	Default	Required
	days.		
`MailboxMaximumAttachmentSize (mmas)`	Specifies the maximum size of a file that can be attached to the mailbox. The valid values for this parameter are 0-9999MB.	10MB	
`MailboxSenderEmailAddress (msea)`	Specifies the mailbox sender's email address.	None	
`MailboxSendOnlyToDefinedUsers (msotdu)`	Defines if email is to be sent to defined Mailbox users. The valid values for this parameter are as follows: • Y: Yes • N: No	None	
`ManageCFServerFlag (mcf)`	Specifies whether TIBCO MFT Platform Server is managed. The values supported for this parameter are as follows: • Y: Yes • N: No	N	No
`ManagedKeyId (s3mkid)`	Specifies the managed key ID. The values supported for this parameter are as follows: • AWS KMS-Managed Key Id	None	No
`OFTP2AuthenticationPublicCertificateFile (oapcf)`	Specifies the OFTP2 authentication public certificate file path.	None	

Parameter	Description	Default	Required
`OFTP2AuthenticationSystemKey (oask)`	Specifies the OFTP2 authentication system key.	D	
`OFTP2CompressFiles (ocf)`	Specifies the OFTP2 file transfers that are to be compressed. The values supported for this parameter are as follows: • Y: Yes • N: No	None	
`OFTP2EERPPublicCertificateFile (oeerppcf)`	Specifies the OFTP2 EERP public certificate file path.	None	
`OFTP2EERPSystemKey (oeerpsk)`	Specifies the OFTP2 EERP system key.	D	
`OFTP2EncryptFiles (oef)`	Specifies encryption to be used for OFTP2 transfers. The values supported for this parameter are as follows: • no • 3des • aes	None	
`OFTP2EncryptionPublicCertificateFile (oepcf)`	Specifies the OFTP2 encryption public certificate file path.	None	
`OFTP2EncryptionSystemKey (oesk)`	Specifies the OFTP2 encryption system key.	D	
`OFTP2LocalOdetteID (oloid)`	Specifies the OFTP2 local Odette ID.	None	

Parameter	Description	Default	Required
`OFTP2LocalPassword (olp)`	Specifies the OFTP2 local password.	None	
`OFTP2PartnerOdetteID (opoid)`	Specifies the OFTP2 partner Odette ID.	None	
`OFTP2PartnerPassword (opp)`	Specifies the OFTP2 partner password.	None	
`OFTP2RequestEERP (oreerp)`	Specifies the OFTP2 request for EERP. The valid values for this parameter is as follows: • Y: Yes • N: No	None	
`OFTP2RequireSessionAuthentication (orsa)`	Specifies if the OFTP2 server requires session authentication. The valid values for this parameter is as follows: • Y: Yes • N: No	None	
`OFTP2RequireEncryptedFiles (oref)`	Specifies the OFTP2 file transfers that are to be encrypted. The valid values for this parameter is as follows: • Y: Yes • N: No	None	
`OFTP2RequireSignedFiles`	Specifies the OFTP2 file transfers	None	

Parameter	Description	Default	Requir ed
`(orsf)`	that are to be signed. The valid values for this parameter is as follows: - Y: Yes - N: No		
`OFTP2SessionAuthentication (osa)`	Specifies the OFTP2 session authentication. The valid values for this parameter is as follows: - Y: Yes - N: No	None	
`OFTP2SignFiles (osf)`	Specifies if the OFTP2 server requires sign files. The valid values for this parameter is as follows: - Y: Yes - N: No	None	
`OFTP2SigningPublicCertificateFile (ospcf)`	Specifies the OFTP2 signing public certificate file path.	None	
`OFTP2SigningSystemKey (ossk)`	Specifies the OFTP2 signing system key.	D	
`OFTP2TLSPublicCertificateFile (otlspcf)`	Specifies the OFPT2 TLS public certificate file path.	None	
`OFTP2TLSSystemKey (otlssk)`	Specifies the OFTP2 TLS system key.	D	

Parameter	Description	Default	Required
`OFTP2UserIDForIncomingRequests (ouidfir)`	Specifies the OFTP2 user ID for incoming requests.	None	
`OFTP2UseTLS (outls)`	Specifies if the OFTP2 server is using TLS or not. The valid values for this parameter is as follows: • Y: Yes • N: No	None	
`OverrideJMSServiceConfiguration (ojmssc)`	When Yes, the URL defined in the `IP Address or fully qualified IP Name` parameter overrides the URL defined in the `Configure JMS Server` parameter. When No, the URL defined in the **Configure JMS Server** parameter is used, and the URL defined in the `IP Address or fully qualified IP Name` parameter is ignored. The values supported for this parameter are as follows: • Y: yes • N: no	None	No
`PASVChecking (pascchk)`	Specifies the PASV checking. The values supported for this parameter are as follows: • 0: none • S: subnet • I: IP Address	None	No

Parameter	Description	Default	Requir ed
`PGPASCII (pascii)`	Specifies whether the PGP ASCII armored format is used. The values supported for this parameter are as follows: • Y: yes • N: no	N	No
`PGPCompression (pcomp)`	Specifies the PGP compression algorithm. The values supported for this parameter are as follows: • DEFAULT • NO • ZIP • ZLIB	Default	No
`PGPEnabled (pena)`	Specifies whether to enable PGP for a server. The values supported for this parameter are as follows: • Y: yes • N: no.	N	Yes
`PGPEncrypt(pencr)`	Specifies PGP encryption. The values supported for this parameter are as follows: • Y: yes • N: no	N	No
`PGPEncryptAlgorithm (pea)`	Specifies which algorithm is used to encrypt the PGP file with. The values supported for this parameter are as follows:	Default	Yes

Parameter	Description	Default	Required
	- 3des - default - aes128 - aes192 - aes256		
`PGPHashAlgorithm (phash)`	Specifies the PGP hash algorithm. The values supported for this parameter are as follows: - DEFAULT - SHA1 - SHA256 - SHA384 - SHA512	None	No
`PGPPrivateKey (pkey)`	Specifies the 1–64-character private key.	None	No
`PGPSign (psign)`	Specifies whether the PGP file transfer is signed. The values supported for this parameter are as follows: - Y: yes - N: no	N	No
`PGPVerifyServerSignature (puver)`	Specifies whether the server's signature in the defined file definition is verified.	N	No
`PGPVerifySignature (pver)`	Specifies whether the signature of the PGP key is verified. The values	N	No

Parameter	Description	Default	Required
	supported for this parameter are as follows: - Y: yes - N: no		
`PlatformServerSystemKey (pssk)`	Specifies the system key of the Platform Server. The values supported for this parameter are as follows: - D: Default - None - UserId of Key	None	No
`PortChecking (portchk)`	Specifies the port to be checked. The values supported for this parameter are as follows: - 0: none - S: subnet - I: IP Address	None	No
`PrincipalName (pn)`	Specifies the HDFS Kerberos principal name.	None	No
`ProxyIPAddress (pipa)`	Specifies the IP address or fully qualified IP name of the proxy server.	None	No
`ProxyIPPort (pipp)`	Specifies the IP port of the proxy.	None	No
`ProxyPassword (ppass)`	Specifies the password of the proxy.	None	No

Parameter	Description	Default	Requir ed
`ProxyType (prxyt)`	Specifies the proxy type. The values supported for this parameter are as follows: • H: HTTP • N: none.	None	No
`ProxyUserName (pun)`	Specifies the user name of the proxy.	None	No
`PSConnectionSecurityType (pscst)`	Specifies the Platform Server connection security type. The values supported for this parameter are as follows: • 0: none • 1: implicit SSL • 2: TLS Tunnel	None	No
`SecurePort (sprt)`	Specifies the Internet Server options context secure port flag. The values supported for this parameter are as follows: • Y: yes • N: no	None	No
`SeparateThread (septh)`	Specifies whether to run in separate threads. The values supported for this parameter are as follows: • Y: yes • N: no	None	No

Parameter	Description	Default	Required
`S3InterfaceEndpointDNSName (ied)`	Specifies the VPC Interface Endpoint DNS Name. Define this parameter when an S3 Access Point with a VPC Interface Endpoint is used to access the bucket. When using S3 Compatible storage, this field points to the VPC DNS name of the S3 compatible storage server. **Note:** - When this parameter is defined for Amazon S3 storage, the **Amazon S3 Options > Amazon S3 Region** must be set to the S3 Bucket Region and the **Amazon S3 Bucket Name** must be set to the S3 Access Point Alias. - When this parameter is defined for S3 Compatible storage, the **Amazon S3 Options > Amazon S3 Region** can be set to any value and the **Amazon S3 Bucket Name** should be set to the S3 Compatible Storage Bucket Name.		
`ServerFileNamePrefix (sfnp)`	Specifies the prefix of the server file name. This is only valid for L node type.	None	No
`ServerName (nn)`	The 1-to-32-character node name. This is the name that TIBCO MFT	None	Yes

Parameter	Description	Default	Required
	Platform Server is known as within the system. If the `ServerName` contains embedded blanks, the entire `ServerName` should be enclosed in double quotation marks (" "). **Note:** This value must point to an existing server definition, and the server name cannot be changed.		
`ServerPlatform (st)`	Specifies the server platform. If the server type is TIBCO MFT Platform Server, the server platform is the operating system of the defined node. If the server type is FTP or SSH, the server platform is the preferred file system emulation of the node. Most SSH (SFTP) and FTP servers should be defined as UNIX, even when executing on Windows. The values supported for this parameter are as follows: • IBMi • zOS • UNIX • Unspecified • WINDOWS • UNISYS2200	Unspecified	No
`SharePointNumberOfUploadBuff`	Specifies the SharePoint number of	None	

Parameter	Description	Default	Requir ed
`ers (spnoub)`	upload buffers.The values supported for this parameter ranges from 1-10.		
`SharePointUploadChunkSize (spucs)`	Specifies the SharePoint upload chunk size.	None	
`SSHBlockSize (sshbs)`	Specifies the SSH block size. The values supported for this parameter are as follows: • 4: yes • 4096-250000	None	No
`SSHKeyFlag (sshkf)`	Specifies the SSH key flag. The values supported for this parameter are as follows: • K: key • C: certificate	None	No
`SSHPooling (sshp)`	Specifies the SSH pooling. The values supported for this parameter are as follows: • Y: yes • N: no	None	No
`SSHPoolingIdleTimeout (sshpit)`	Specifies the SSH pooling idle time out. The values supported for this parameter are as follows: • 1-60 minutes	None	No
`SSHSystemKey (sshsk)`	Specifies the SSH system key. The values supported for this	None	No

Parameter	Description	Default	Required
	parameter are as follows: - SSH System Key - SSH Private Key		
`TraceLevelFlag (tf)`	Specifies the trace level of the flag. The values supported for this parameter are as follows: - 0: no tracing - 1-5 and 10: different levels of tracing This flag should only be set under instruction from TIBCO Technical Support.	0	Yes
`UseAmazonAcceleration (accl)`	Specifies whether to use Amazon acceleration. The values supported for this parameter are as follows: - Y: yes - N: no	None	No
`Visibility (vsb)`	Specifies the visibility of a node. The values supported for this parameter are as follows: - PUB: public - PRI: private	None	No

Sample `AddServer` Command

This command adds a Platform Server node called NYNode1, assigns an IP address and IP port to the NYNode1 node, and sets some default values for the server. By specifying the `DisableFlag` parameter as N, the server definition is immediately available after it is successfully added.

```
java cfcc.CFAdmin a:AddServer ServerName:NYNode1 IPName:192.192.100.1 IPPort:46464
DefaultEncryptType:N CompressType:Y DisableFlag:N
```

GetServer

The `GetServer` command action is used to display configuration parameters from a single node definition in the node definition table.

To use the `GetServer` action command, you must have UpdateServerRight. For more information, see AddUserToRole.

When this command is executed successfully, the defined ServerName is displayed along with the configuration parameters for the defined server. If the node that you want to display is not defined, an error occurs.

Parameter	Description	Default	Required
`ServerName (nn)`	Specifies the 1-to-32-character node name. This parameter is the name that TIBCO MFT Platform Server is known as within the TIBCO MFT Internet Server . If the server name contains embedded spaces, the entire server name must be enclosed in double quotation marks (“). **Note:** This value must point to an existing server definition. If the node that you want to update is not defined, an error occurs.	None	Yes

Sample `GetServer` Command

This command displays parameters defined for the NYNode1 server. The server name is required for the `GetServer` command action.

```
java cfcc.CFAdmin a:GetServer ServerName:NYNode1
```

RetrieveAllServers

The `RetrieveAllServers` command action is used to display configuration parameters from all node definitions from the node definition table.

To use the RetrieveAllServers action command, you must have UpdateServerRight. For more information, see AddUserToRole.

When this command is executed successfully, each node that is in the server table will be displayed along with the configuration parameters defined for each server definition.

No parameters are supported for this command action.

Sample `RetrieveAllServers` Command

This command displays parameters defined for all server definitions.

```
java cfcc.CFAdmin a:RetrieveAllServers
```

RemoveServer

The `RemoveServer` command action is used to delete a node definition from the node definition table.

To use the `RemoveServer` action command, you must have UpdateServerRight. For more information, see AddUserToRole.

When this command is executed successfully, the server will be removed from the server definition table.

Parameter	Description	Default	Required
`ServerName (nn)`	Specifies the 1-to-32-character node name. This is the name that TIBCO MFT	None	Yes

Parameter	Description	Default	Required
	Platform Server is known as within . If the server name contains embedded spaces, the entire server name must be enclosed in double quotation marks (“). **Note:** This parameter must point to an existing server definition. If the node that you want to update is not defined, you will receive an error.		

Sample `RemoveServer` Command

This command deletes the NYNode1 server. The server name is required for the `RemoveServer` command action.

```
java cfcc.CFAdmin a:RemoveServer ServerName:NYNode1
```

UpdateServer

The `UpdateServer` command action is used to update an existing TIBCO MFT Internet Server node definition.

The node definition contains information about the remote Platform Server system. You must define node definitions when you are connecting to a remote TIBCO MFT Platform Server. If you are storing files locally, you do not have to define node definitions.

To use the `UpdateServer` action command, you must have UpdateServerRight. For more information, see AddUserToRole.

Note: `UpdateServer` and `AddServer` commands have common parameters. For `UpdateServer` command parameters, see AddServer.

Sample `UpdateServer` Command

This command updates the NYNode1 server. The server name is required for the `UpdateServer` command action. This command updates the `DefaultEncryptType`, `CompressType`, and `DisableFlag` parameters.

```
java cfcc.CFAdmin a:UpdateServer ServerName:NYNode1 DefaultEncryptType:R
CompressType:Y DisableFlag:N
```

Session Commands

The session commands are used to list and delete sessions.

Note: These commands are not supported when using REST web service.

Action	Description
DeleteSessionId	Deletes a session ID.
DeleteExpiredSessionIds	Deletes all expired session IDs.
GetExpiredSessionIds	Lists expired session IDs.

DeleteSessionId

The `DeleteSessionId` command action is used to delete a session ID.

The session IDs are used to regulate the amount of time that a user can remain inactive when processing the requests. This command can only be used when requested by TIBCO technical support.

To use the `DeleteSessionId` action command, you must have UpdateSessionRight. For more information, see AddUserToRole.

If the session ID is not found, the action will fail and an error message will be displayed.

Parameter	Description	Default	Required
`SessionID (sid)`	Specifies the 1-to-64-character session ID. This information is typically extracted from the ListActiveSessionIds or GetExpiredSessionIds action command.	None	Yes

Sample `DeleteSessionId` Command

This command deletes the sessions with the defined session ID.

```
java cfcc.CFAdmin a:DeleteSessionId SessionID:583def%6abdeef%7b30
```

Note: This command is not supported when using REST web service.

DeleteExpiredSessionIds

The `DeleteExpiredSessionIds` command action is used to delete all session IDs that are on the session database but have expired.

The session IDs are used to regulate the amount of time that a user can remain inactive when processing the requests. This command can only be used when requested by TIBCO technical support.

To use the `DeleteExpiredSessionIds` action command, you must have UpdateSessionRight. For more information, see AddUserToRole.

No parameters are supported for this command action.

Sample `DeleteExpiredSessionIds` Command

This command deletes all expired sessions.

```
java cfcc.CFAdmin a:DeleteExpiredSessionIds
```

Note: This command is not supported when using REST web service.

GetExpiredSessionIds

The `GetExpiredSessionIds` command action is used to display a list of all session IDs that are on the session database but have expired.

The session IDs are used to regulate the amount of time that a user can remain inactive when processing the requests. This command can only be used when requested by TIBCO technical support.

To use the `GetExpiredSessionIds` action command, you must have UpdateSessionRight. For more information, see AddUserToRole.

No parameters are supported for this command action.

Sample `GetExpiredSessionIds` Command

This command lists all expired sessions.

```
java cfcc.CFAdmin a:GetExpiredSessionIds
```

Note: This command is not supported when using REST web service.

Transfer Commands

The transfer commands are used to define, list, update, and delete transfer definition records in the system.

Action	Description
AddTransfer	Adds a transfer definition.
DeleteExpiredTransfers	Deletes expired transfer records.

Action	Description
GetTransfer	Lists a specific transfer definition.
RetrieveAllTransfers	Lists all transfer definitions.
RetrieveAllTransfersForUser	Lists all transfer definitions for a user.
RemoveTransfer	Deletes a transfer definition.
SearchForTransfers	Searches for transfer records.
UpdateTransfer	Alters a transfer definition.

AddTransfer

The `AddTransfer` command action is used to add a file definition to the TIBCO MFT Internet Server .

The file definition contains information about where the file is located, who can access the file, and the characteristics of the file.

To use the `AddTransfer` action command, you must have `UpdateTransferDefinitionRight`. For more information, see AddUserToRole.

In the following table, parameters for this command are provided in alphabetical order.

> **Note:** The parameters provided in this table are also used for the `UpdateTransfer` command.

Parameter	Description	Default	Required
`AllowableProtocol (apl)`	Specifies the protocol to be used for this transfer. The values supported for this parameter are as follows:	All	Yes

Parameter	Description	Default	Required
	- FTP - Secure FTP (referred as SECUREFTP) - HTTPS - Secure - CF(for Platform Server) - AS2 - All(includes all listed protocols)		
`AllowClientTransferMode (actm)`	Specifies whether to allow client transfer mode. The values supported for this parameter are as follows: - Y: yes - N: no	None	No
`AllowDelete (adel)`	Defines whether MFT allows the FTP client to issue the `Delete` command for a file defined by this transfer definition.	No	No
`AllowFTPSiteCommandPassThrough (afscpt)`	Specifies whether to allow FTP site command pass through. The values supported for this parameter are	None	No

Parameter	Description	Default	Required
	as follows: - Y: yes - N: no		
`AllowMakeDirectory (amkd)`	Defines whether MFT allows the FTP client to create a directory within the directory structure defined by this transfer definition.	No	No
`AllowRemoveDirectory (armd)`	Defines whether MFT allows the FTP client to remove a directory within the directory structure defined by this transfer definition.	No	No
`AllowRename (aren)`	Defines whether MFT allows the FTP client to issue the `Rename` command for a file defined by this transfer definition.	No	No
`AuthGroupId (gid)`	Specifies the 1-to-64-character group ID that is authorized to transfer this file. A transfer can be authorized to a user ID or a group. See also `UserId`.	None	Either `UserId` or `AuthGroupId` must be specified.
`AvailableDate (avd)`	Specifies the date this	Today's date	No

Parameter	Description	Default	Required
	file is available for transfer. The format for this parameter is YYYY/MM/DD. The date range is 2000/01/01 to 2099/12/31.		
`AVMode (avmode)`	Defines the ICAP AV Transfer mode. The valid values for this parameter are as follows: S: Streaming F: Store and Forward D: Default	D	
`AVScanFileRegex (avsfr)`	Defines a regex (regular expression) that defines files to be scanned when doing a transfer.	None	
`AVTransferScan (avts)`	Defines whether this transfer definition is scanned for viruses. The valid values for this parameter are as follows: • Y: Yes • N: No • D: Server Default	D	
`ChkptInterval (cki)`	Specifies how many minutes checkpoint	5	No

Parameter	Description	Default	Required
	interval in. The max value is 59.		
`ChkptRestartFlag (ckf)`	Specifies whether checkpoint restart is supported. The valid values are Y and N.	Yes	No
`ClientCompressFlag (cc)`	Specifies whether to use compression when transferring this file. The valid values are Y and N.	Yes	No
`ClientFileName (cfn)`	The 1-to-256-character file name/location on the client machine. If the file name/location contains embedded blanks the entire filename should be enclosed in double quotation marks (" ").	None	No
`CredPassThruFlag (cpt)`	Specifies whether credentials are passed from the client to the server. This capability is only used when the initiating client is FTP, SSH or Platform Server and when the initiating client enters a user ID and password. Here is how it works. The values supported for this parameter are as	None	No

Parameter	Description	Default	Required
	follows: - N: none - Y: yes **Note:** This parameter is infrequently used and must typically be set to N.		
`CRLF (crlf)`	Specifies how the records are delimited. The values supported for this parameter are as follows: - Y: delimited by carriage return line feed (`CRLF`). - L: delimited by line feed (`LF`). - N: there are no delimiters.	Yes - if `DataType` is Text, No - for any other `DataType`	No
`DataType (dt)`	Specifies the type of data being transferred. Valid data types are: - B: binary - T: text	Binary	No
`DefaultNodePwd (dnp)`	Specifies the password to be used with `DefaultNodeUserid`.	None	No

Parameter	Description	Default	Required
	Note: Certain target nodes may have case sensitive passwords.		
`DefaultNodeUserId (dnu)`	Specifies the 1 to 20 character user ID to be used to authenticate the file transfer. This authentication takes place at the server specified in `ServerName`.	None	No
`DefaultWinDomain (dnt)`	Specifies the Windows domain to be used with `DefaultNodeUserid` and `DefaultNodePwd` parameters. Only applies for Windows based target systems.	None	No
`Department (dpt)`	Specifies the file definition’s department.	None	No
`Description (d)`	Specifies the 1-to-256-character description of this file, this description is presented to the client user to describe the contents of the file. The entire description must be enclosed in double quotation marks (" ").	None	No

Parameter	Description	Default	Required
`DirectoryTransfer (dir)`	Specifies whether this transfer is a directory transfer or a single file transfer. The valid values are Y and N.	No	No
`DisableFlag (dis)`	Specifies whether this transfer definition should be disabled. The valid values are Y and N.	None	Yes
`DownloadRestriction (dr)`	Specifies restrict download REGEX.	None	No
`DownloadRestrictionFlag (drf)`	Specifies whether to set restrictions for download. The values supported for this parameter are as follows: • Y: enforce rules • N: no rules	None	No
`DownloadUploadFlag (duf)`	Specifies the direction of the transfer. This direction is from the user perspective. The values supported for this parameter are as follows: • U: user uploads a file. • D: user downloads a file.	None	No

Parameter	Description	Default	Required
`EmailFailureTemplate (eft)`	Specifies the email template on the server to use for a failed transfer email. This email template must reside on the server.	None	No
`EmailMaximumAttachmentSize (emas)`	Specifies the maximum size of a file that can be attached to the email.	None	
`EmailMessageTextFilePath (emtfp)`	Specifies the path to the file which contains the email message text.	None	
`EmailRecipients (erec)`	Specifies the list of email recipients, separated by a comma (,) or a semicolon(;). Tokens are accepted as well.	None	
`EmailSenderEmailAddress (esea)`	Specifies a valid email address. This field must be filled in when the `EmailUseClientAddress` parameter is set to '0'.	None	
`EmailSubject (esub)`	Specifies the email subject.	None	
`EmailSuccessTemplate (est)`	Specifies the email template on the server to use for a successful transfer email.	None	No

Parameter	Description	Default	Required
`EmailUseClientAddress (euca)`	Specifies the client address used in the email. The supported values for this parameter are as follows: - U: User - S: Server - O: Other	None	
`EncryptFlag (e)`	Specifies the level of encryption to be used with this transfer. The values supported for this parameter are as follows: - N: none - D: DES encryption - R: Rijndael encryption - DEF: default setting	Uses the encryption from the server definition.	No
`ExpirationDate (epd)`	Specifies the date when this transfer expires. The values supported for this parameter are as follows: - never: the transfer does not expire. - +n: n days after	Never	No

Parameter	Description	Default	Required
	the `AvailableDate`. - date: actual date in YYYY/MM/DD format between 2000/01/01 and 2099/12/31.		
`FormPostParameters (fpp)`	Specifies the form post parameters.	None	No
`FTPAlias (fa)`	Specifies the file name or directory that is displayed when an FTP client accesses this file record. Valid length is up to 256 characters. When the file record is defined as a directory, the `FTPAlias` is displayed to the user as a directory. When the file record is defined as a file, the `FTPAlias` is displayed to the user as a file. If an FTP client accesses this file record and this parameter is not defined, the `TransferID` is used as the `FTPAlias`.	TransferID associated with the file record	No - but strongly suggested for FTP/Secure FTP transfers
`HTTPHeaders (httph)`	Specifies the HTTP headers with header name:header value. For multiple HTTP headers,	None	No

Parameter	Description	Default	Required
	separate headers with a semicolon (;).		
`JMSEOFMessage (jmseom)`	Specifies whether MFT writes an empty JMS message at the end of file. The values supported for this parameter are as follows: - Y: yes - N: no	None	No
`JMSInputSelector (jmsis)`	Specifies the selector that is used to filter JMS messages when reading from a JMS queue.	None	No
`JMSMaxMessageSize (jmsmms)`	Specifies the maximum size of any individual message written to the JMS queue.	None	No
`JMSTypeProperty (jmstp)`	Specifies the JMS type output property that is set when writing data to a JMS queue.	None	No
`KeyFlag (kf)`	Specifies the key flag. The values supported for this parameter are as follows: - K: key - C: certificate	None	No

Parameter	Description	Default	Required
`LocalTranslationTable (lt)`	Specifies the location of the character translation table.	None	No
`MailboxRecipients (mrec)`	Specifies the list of recipients that are separated by commas (,) and semicolons (;). Accepts tokens as well.	None	
`MailboxUseClientAddress (muca)`	Specifies the use of the mailbox client address. The values supported for this parameter are as follows: • U: User • S: Server • O: Other	None	
`MailboxSenderEmailAddress (msea)`	Specifies a valid email address. This field must be filled in when `MailboxUseClientAddress` is set to '`O`'.	None	
`MailboxSubject (msub)`	Specifies the subject field of the mailbox.	None	
`MailboxMaximumAttachmentSize (mmas)`	Specifies the maximum file size that can be attached in the mailbox.	None	
`MailboxExpirationDays (mexpd)`	Specifies the number of days when this mailbox	None	

Parameter	Description	Default	Required
	expires.		
`MailboxExpireWhenAllFilesDownloaded (mexpwfd)`	Specifies if the mailbox attachment expires after all the files are downloaded. The valid values for this parameter are as follows: - Y: Yes - N: No	None	
`MailboxMessageTextFilePath (mmtfp)`	Specifies the path to the file which contains the mailbox message text.	None	
`NotifyEmailTemplate (net)`	Specifies the email template on the server to use to notify the user that a file is added.	None	No
`NotifyFileAvailable (nf)`	Specifies whether to send an email to the user when a file is available. If the file being added is for a group, all the members of that group are notified. The email address used for this notification is specified during the AddUser. The valid values are Y and N.	None	No
`OFTP2RecordFormat (oftp2rf)`	Specifies the OFTP2	Unstructure	

Parameter	Description	Default	Required
	record format The values supported for this parameter are as follows: • U: Unstructured • T: Text • F: Fixed • V: Variable	d	
`OFTP2MaximumRecordSize (oftp2mrs)`	Defines the record size (Fixed) or maximum record size (Variable) of the records in the file	None	
`OFTP2DestinationOdetteID (oftp2doid)`	When transferring files to an OFTP2 clearinghouse, this field defines the destination Odette ID.	None	
`OFTP2VirtualFileDescription (oftp2vfd)`	Specifies the virtual file description for the OFTP2 transfers.	None	
`OneTimeFlag (ot)`	Specifies what should happen to the file record after the transfer has completed successfully. The values supported for this parameter are as follows: • Y: after the transfer delete	Yes	No

Parameter	Description	Default	Required
	the record. - N: after the transfer keep the record. - K: after the transfer keep the record, but hide it from the user or group. The default value is Y.		
`PGPASCII (pascii)`	Specifies whether the ASCII armored format is used. The valid values are Y and N.	N	No
`PGPCompression (pcomp)`	Specifies what type of compression is used. The values supported for this parameter are as follows: - default - none - zip - zlib	Default	No
`PGPDecrypt (pde)`	Specifies whether the file is decrypted when it arrives at the remote location. The valid values are Y and N.	N	No
`PGPEncryptAlgorithm (pea)`	Specifies which algorithm is used to	Default	Yes

Parameter	Description	Default	Required
	encrypt the PGP file with. The values supported for this parameter are as follows: - 3des - default - cast5 - blowfish - aes128 - aes192 - aes256		
`PGPEncypt (pen)`	Specifies whether the file is encrypted when it arrives at the remote location. The valid values are Y and N.	N	No
`PGPHashAlgorithm (phash)`	Specifies which hash algorithm is used when encrypting the PGP file. The values supported for this parameter are as follows: - default - sha1 - sha256 - sha384 - sha512	Default	Yes

Parameter	Description	Default	Required
`PGPPrivateKey (pkey)`	Specifies the 1 – 64 character private key.	None	No
`PGPSign (psign)`	Specifies whether the PGP file transfer is signed. The valid values are Y and N.	N	No
`PGPVerifySignature (pver)`	Specifies whether the signature of the PGP key is verified. The valid values are Y and N.	N	No
`PGPVerifyUserSignature (puver)`	Specifies whether the user’s signature in the defined file definition is verified. The valid values are Y and N.	N	No
`PostActionData1-4 (AD1-4)`	Specifies the data passed to the `PostActionType` when the conditions specified in `PostActionFlag` met. Data with embedded blanks should be enclosed in double quotation marks (" ").	None	No
`PostActionFlag1-4 (AF1-4)`	Specifies the conditions when a post processing action should occur. The post processing action is performed at the server defined in `ServerName`. Used in	None	No

Parameter	Description	Default	Required
	conjunction with PostActionType and PostActionData. The values supported for this parameter are as follows: • S: transfer Successful • F: transfer Failed		
PostActionType1-4 (at1-4)	Specifies the type of post processing action to be performed when the PostActionFlag conditions met. The values supported for this parameter are as follows: • CALLPGM: call a z/OS program with program to program parameter linkage. • CALLJCL: call a z/OS program with JCL to program parameter linkage. • COMMAND: issue a command at the node specified in NodeName.	None	No

Parameter	Description	Default	Required
	- SUBMIT: submit a job at the node specified in `NodeName`. - None: delete the PPA data of the transfer.		
`ProcessName (pn)`	Specifies the name of a process.	None	No
`RemoteTranslationTable (rt)`	Specifies the location of the character translation table on the client machine.	None	No
`RemoveTrailingBlanks (fo)`	Used only with text type transfers. Specifies whether to remove any trailing spaces. This option is only valid when z/OS is sending the file. The valid values are Y and N.	None	No
`ServerFileName (sfn)`	Specified the 1-to-256-character file name/location of the server machine. If the `NodeName` is *LOCAL, the `ServerFileName` would be located on the . If the file name/location contains embedded	None	No

Parameter	Description	Default	Required
	blanks the entire filename must be enclosed in double quotation marks (" ").		
`ServerName (sn)`	Specifies the 1-to-64-character name of the MFT Platform Server within your network. A node is a target destination that is running MFT Platform Server that can send or receive files. The `ServerName` may also be specified as *LOCAL, this refers to the which does not have to be running MFT Platform Server.	None	No
`SharePointDocumentLibraryUrl (sdlu)`	Specifies the URL of the SharePoint Document Library.	None	
`SSHSystemKey (sshsk)`	Specifies the name of the SSH system key.	None	No
`ToEmailAddrFailure (eaf)`	Specifies the email address to be used when a transfer fails. You must configure your email server details on the System Configuration page to use this function.	None	No

Parameter	Description	Default	Required
`ToEmailAddrSuccess (eas)`	Specifies the email address to be used when a transfer is successful. You must configure your email server details on the System Configuration page to use this function.	None	No
`TraceLevelFlag (tlf)`	Specifies whether to use trace level. The values supported for this parameter are as follows: • 0: no tracing • 1-5 and 10: different levels of tracing This flag should only be set under instruction from TIBCO Technical Support.	0	No
`TransferType (tt)`	Specifies the type of transfer. The values supported for this parameter are as follows: • S: stream • F: form/post	None	No
`TruncateFlag (tf)`	Specifies whether to truncate. The values	None	no

Parameter	Description	Default	Required
	supported for this parameter are as follows: • NONE • TRUNCATE • WRAP		
`UnixPermissions (uxp)`	Specifies the Unix file permissions. Values are 000 - 777.	None	No
`UploadRestrictionFlag (urf)`	Specifies whether to set restrictions for upload. The values supported for this parameter are as follows: • Y: enforce rules • N: no rules	None	No
`UploadRestriction (ur)`	Specifies restrict upload REGEX.	None	No
`UploadMaximumSize (ums)`	Specifies maximum upload size	None	No
`UserData (ud)`	Specifies user data.	None	No
`UserId (uid)`	Specifies the 1-to-64-character user ID to transfer this file. A transfer can be authorized to a user ID or a group. See also `AuthGroupId`.	None	Either `UserId` or `AuthGroupId`

Parameter	Description	Default	Required
`ValidDays (vd)`	Specifies the 7 character day of week pattern when this file can be accessed, Sunday being the first character, Monday the second, and so on. where each character can be Y or N.	YYYYYYY	No
`ValidEndTime (vet)`	Specifies the end time in military format HHMM when this file can be accessed.	2359	No
`ValidStartTime (vst)`	Specifies the start time in military format HHMM when this file can be accessed.	0000	No
`ViewFilesDirectories (vfd)`	Specifies the files or directories to view. Does not allow downloads. The values supported for this parameter are as follows: • Y: yes • N: no	None	No
`WriteMode (wm)`	Specifies the options used when opening the output file on the target system. The values supported for this parameter are as follows:	CRN	No

Parameter	Description	Default	Required
	• C: create the file, if it already exists, the transfer fails. • CR: create/replace, if the file does not exist, it is created, if the file already exists it is replaced. • R: replace the file. If it does not exist, the transfer fails. • A: append to the file. If it does not exist, the transfer fails. • CA: create/append, if the file does not exist it will be created. If the file already exists, it is appended to. • CRN: create/replace/new. The same as CR (create/replace), but also creates the directory structure if it does not already exist.		

Parameter	Description	Default	Required
`zOSAllocPri (ap)`	Specifies the primary allocation value in units of zOSAllocType. Only for transfers to z/OS.	None	No
`zOSAllocSec (as)`	Specifies the secondary allocation value in units of zOSAllocType. Only for transfers to z/OS.	None	No
`zOSAllocType (at)`	Specifies the allocation type to be used when transferring files to a z/OS system. The values supported for this parameter are as follows: • T: tracks • B: blocks • C: cylinders • K: kilobytes • M: megabytes	None	No
`zOSBlockSize (bs)`	Specifies the block size to be used for file being transferred to z/OS.	None	No
`zOSDataClass (dtc)`	Specifies a valid data class used when transferring files to a z/OS system.	None	

Parameter	Description	Default	Required
	A valid value is a 1-to-8-character data class name defined by your storage administrator.		
`zOSLRECL (cl)`	Specifies the logical record length for files being transferred to z/OS.	None	No
`zOSMgtClass (mgt)`	Specifies a valid management class used when transferring files to a z/OS system. A valid value is a 1-to-8-character management class name defined by your storage administrator.	None	No
`zOSRECFM (fm)`	Specifies the record format for files being transferred to z/OS. The values supported for this parameter are as follows: • F: Fixed. • FA: Fixed ASA. • FB: Fixed Block. • FBA: Fixed Blocked ASA. • FBM: Fixed Blocked Machine. • FBS: Fixed Block	None	No

Parameter	Description	Default	Required
	Standard. - FM: Fixed Machine. - FS: Fixed Standard. - V: Variable. - VA: Variable ASA. - VB: Variable Blocked. - VBA: Variable Blocked. ASA - VBM: Variable Blocked. Machine - VBS: Variable Blocked. Spanned - VM: Variable Machine. - VS: Variable Spanned. - U: Undefined.		
`zOSStorClass (sc)`	Specifies a valid storage class used when transferring files to a z/OS system. A valid value is a 1-to-8-character storage class name defined by your storage administrator.	None	No
`zOSUnit (ut)`	Specifies the device	None	No

Parameter	Description	Default	Required
	type for a file being transferred to z/OS. Valid values are any device type defined to your z/OS system.		
`zOSVolume (v)`	Specifies the volume serial number for transferring files to z/OS. The valid values are any 1-to-6-character volume serial number on your z/OS system.	None	No

Sample `AddTransfer` Command

This command adds a file to the database.

```
java cfcc.CFAdmin a:AddTransfer ClientFileName:"C:\TEMP 001\24.jpg"
ServerFileName:"C:\24.jpg" ServerName:ARTDEPT DisableFlag:N ValidStartTime:0000
ValidEndTime:2359 ValidDays:YYYYYYY OneTimeFlag:K EncryptFlag:D WriteMode:C CRLF:N
Description:"Corporate Logo JPG format" NotifyFileAvailable:Y ExpirationDate:+1
AuthGroupId:PRINTERS DataType:B DownloadUploadFlag:D
```

DeleteExpiredTransfers

The `DeleteExpiredTransfers` command action is used to delete all file definitions that have expired.

A file definition expires when the current date is greater than the date defined by the `ExpirationDate` parameter.

To use the `DeleteExpiredTransfers` action command, you must have UpdateTransferDefinitionRight. For more information, see AddUserToRole.

No parameters are supported for this command action.

Sample `DeleteExpiredTransfers` Command

This command deletes all expired file definitions.

```
java cfcc.CFAdmin a:DeleteExpiredTransfers
```

GetTransfer

The `GetTransfer` command action is used to display detailed information about one specific file definition in the system.

To use the `GetTransfer` action command, you must have UpdateTransferDefinitionRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`TransferId (tid)`	Specifies the 12-character transfer ID that was assigned when the file definition was added.	None	Yes

Sample `GetTransfer` Command

This command displays all the parameters defined for the transfer ID specified.

```
java cfcc.CFAdmin a:GetTransfer TransferId:F60930000127
```

RetrieveAllTransfers

The `RetrieveAllTransfers` command action is used to list all file definitions within the system.

To use the `RetrieveAllTransfers` action command, you must have UpdateTransferDefinitionRight. For more information, see AddUserToRole.

No parameters are supported for this command action.

Sample `RetrieveAllTransfers` Command

This command displays the parameters for all the files defined to the database.

```
java cfcc.CFAdmin a:RetrieveAllTransfers
```

RetrieveAllTransfersForUser

The `RetrieveAllTransfersForUser` command action is used to display a list of all file definitions that have been defined for a user ID.

To use the `RetrieveAllTransfersForUser` action command, you must have ViewTransferDefinitionRight and ViewGroupRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`UserId`	Specifies the 1-to-64-character user ID of the user you want to inquire on.	None	Yes

Sample `RetrieveAllTransfersForUser` Command

This command displays all the information for each file definition defined for this user.

```
java cfcc.CFAdmin a:RetrieveAllTransfersForUser UserId:Accounting001
```

RemoveTransfer

The `RemoveTransfer` command action is used to delete a file definition from the system.

To use the `RemoveTransfer` action command, you must have UpdateTransferDefinitionRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`TransferId (tid)`	Specifies the 12-character transfer ID that was assigned when the file definition was added.	None	Yes

Sample `RemoveTransfer` Command

This command removes a file definition from the database.

```
java cfcc.CFAdmin a:RemoveTransfer TransferId:F21530000818
```

SearchForTransfers

The `SearchForTransfers` command action is used to search for all file definitions that match the defined selection criteria.

Use the asterisk (*) as a wildcard character for REST web service in all parameters to select file definitions based on a partial key.

To use the `SearchForTransfers` action command, you must have UpdateTransferDefinitionRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`AuthGroupId (gid)`	Specifies the 1-to-64-character group ID that is authorized to transfer this file. A transfer can be authorized to a user or a group. See also the `UserId` parameter.	All	No
`Department (dpt)`	Specifies the department associated with the file. The value is ignored for	None	No

Parameter	Description	Default	Required
	department admin.		
`Description (d)`	Specifies the 1-to-256-character description of this file. This description will be presented to the client user to describe the contents of the file. If the description contains embedded spaces, the entire description must be enclosed in double quotation marks (").	None	No
`ExpiredTransfers (expt)`	Specifies the valid values for this parameter are as follows: Y:yes, N:no	None	No
`FTPAlias (fa)`	Specifies the transfer virtual alias.	None	No
`ServerFileName (sfn)`	Specifies the 1-to-256-character file name or location of the server machine. If the server name is *LOCAL, the server file name will be located on the server. If the file name or location contains embedded spaces, the entire file name must be enclosed in double quotation marks (“).	None	No
`ServerName (sn)`	Specifies the 1-to-64-	None	No

Parameter	Description	Default	Required
	character name of the node within your network. A node is a target destination that is running TIBCO MFT Platform Server that can send or receive files. The server name might also be specified as *LOCAL, which refers to the server. The server does not have to be running TIBCO MFT Platform Server.		
`TransferId (tid)`	Specifies the 12-character transfer ID that was assigned when the file definition was added.	None	No
`UserId (uid)`	Specifies the 1-to-64-character user ID of the user who is authorized to transfer this file. A transfer can be authorized to a user or a group. See also the `AuthGroupId` parameter.	None	No

Sample `SearchForTransfers` Command

This command searches for all file definitions that match the selection criteria.

```
java cfcc.CFAdmin a:SearchForTransfers ServerFileName:/tmp/% ServerName:NYNode1
```

Note: The ServerFileName parameter uses the wildcard character to match based on a partial key, while the ServerName parameter must exactly match the value in the file record.

UpdateTransfer

The UpdateTransfer command action is used to update a file definition to the system.

The file definition contains information about where the file is located, who has access to the file, and the characteristics of the file.

To use the UpdateTransfer action command, you must have UpdateTransferDefinitionRight. For more information, see AddUserToRole.

Note: UpdateTransfer and AddTransfer commands have common parameters. For UpdateTransfer command parameters, see AddTransfer.

Sample UpdateTransfer Command

This command updates a file definition in the database.

```
java cfcc.CFAdmin a:UpdateTransfer TransferId:F51150000008 ValidDays:YYYYYYY
ValidStartTime:0000 ValidEndTime:2359 ExpirationDate:never
```

User Commands

The user commands are used to define, list, update, and delete users in the system.

Action	Description
AddAdminUser	Adds an administrative user with administrator rights.
AddTransferUser	Adds a user with transfer rights.

Action	Description
ChangePassword	Changes a user password.
GetUser	Displays a specific user.
RetrieveAllUsers	Displays all users.
RemoveUser	Deletes a user.
UpdateUser	Updates a user.

AddAdminUser

The `AddAdminUser` command action is used to define an administrative user to the system.

This user is automatically assigned the administrator right.

In the following table, parameters for this command are provided in alphabetical order.

> ℹ **Note:** The parameters provided in this table are also used for the `UpdateUser` command.

Parameter	Description	Default	Required
`AddPGPKey (paddk)`	Specifies whether to allow a user to add a PGP key. The valid values are as follows: • Y • N • D: default	D	No
`AllowableProtocol (apl)`	Specifies the protocol that the user will be allowed to use for a file transfer.	All	No

Parameter	Description	Default	Required
	The valid values are as follows: - FTP - Secure FTP: referred as SECUREFTP. - HTTPS - CF: for TIBCO MFT Platform Server. - Secure - AS2 - SSL/TLS (for MFT Platform Server) - All: includes all listed protocols except AS2.		
`AssignedGroups (ag)`	Specifies the groups that a user has to be added to. Add groups between double quotation marks ("") and use a semicolon (;) as delimiter.	None	No
`AssignedRights (ar)`	Specifies the rights that have to be assigned to a user. A transfer right is assigned by default. Add rights between double quotation marks ("") and use a semicolon (;) as delimiter.	None	No
`CanChangePassword (ccp)`	Specifies whether to allow this user to change	Y	No

Parameter	Description	Default	Required
	password. The valid values are as follows: - Y - N		
`CertificateDN (dn)`	Specifies the 1-to-1024-character certificate, to distinguish the name of the user.	None	No
`CFAuthType (cfat)`	Specifies the type of authentication for CF transfers. The values supported for this parameter are as follows: - 1: password only - 2: certificate only - 3: certificate or password - 4: certificate and password	None	No
`ChangePasswordNextLogin (cpnl)`	Specifies whether this user has to change password at the next logon. The valid values are as follows: - Y - N	Y	No
`Company Name (cname)`	Specifies the 1-to-64-	None	No

Parameter	Description	Default	Required
	character company name.		
`DefaultRole (dr)`	Specifies the default role of the user.	None	No
`Department (dpt)`	Specifies the department the user will be placed in.	None	No
`Description (d)`	Specifies the 1-to-256-character description for this user. If the description contains embedded spaces, the entire description must be enclosed in double quotation marks (").	None	No
`DisableFlag (dis)`	Specifies whether this user is initially disabled from the system. The valid values are as follows: - Y - N	N	No
`EmailAddr (ea)`	Specifies the 1-to-64-character email address of the user.	None	No
`EndDate (ed)`	Specifies the date when the account of this user will become inactive in the system. The format is *YYYY/MM/DD*.	None	Yes

Parameter	Description	Default	Required
	The date range is 2000/01/01 to 2099/12/31.		
`ExpirationDate (epd)`	Specifies the date when the account of this user is deleted from the system. The format is *YYYY/MM/DD*. The date range is 2000/01/01 to 2099/12/31.	None	Yes
`FTPAuthType (ftpat)`	Specifies the type of authentication for FTP transfers. The values supported for this parameter are as follows: • 1: password only • 2: certificate only • 3: certificate or password • 4: certificate and password	None	No
`FullName (fn)`	Specifies the 1-to-256-character name for this user. If the full name contains embedded spaces, the entire full name must be enclosed in double quotation marks (").	None	Yes
`HTTPSAuthType (httpsat)`	Specifies the type of authentication for HTTPS	None	No

Parameter	Description	Default	Required
	transfers. The values supported for this parameter are as follows: • 1: password only • 2: certificate only • 3: certificate or password • 4: certificate and password		
`IPName (ipn)`	Specifies the 1-to-64-character machine name or IP address. If the `RestrictUser` parameter is configured as Y, this parameter is required.	None	No
`LockFlag (l)`	Specifies whether to lock the user out of the system. The valid values are as follows: • Y • N	None	No
`ManageDepartments (md)`	Specifies the departments to be managed separated by a semicolon (;).	None	No
`MaxFileSize (mfs)`	Specifies the maximum size of a file.	None	No
`Netmask (netm)`	Specifies the 1 to 64 byte	None	No

Parameter	Description	Default	Required
	netmask.		
`Password (pw)`	Specifies the 1-to-30-character password assigned to this user. The password cannot contain any embedded spaces. It is case sensitive.	None	Yes
`PasswordNeverExpires (pne)`	Specifies whether this password ever expires. This parameter overrides the global password rules. The valid values are as follows: • Y • N	N	No
`PhoneNumber (phone)`	Specifies the 1-to-64-character telephone number.	None	No
`RestrictUser (rus)`	Specifies whether to restrict this user. The valid values are as follows: • Y • N	N	No
`SSHAuthType (sshat)`	Specifies the type of authentication for SSH transfers. The values supported for this	None	No

Parameter	Description	Default	Required
	parameter are as follows: • 1: password only • 2: certificate only • 3: certificate or password • 4: certificate and password		
`StartDate (sd)`	Specifies the date when this user will be active in the system. The format is *YYYY/MM/DD*. The date range is 2000/01/01 to 2099/12/31.	None	Yes
`TraceLevelFlag (tf)`	Specifies the trace level of the flag. The values supported for this parameter are as follows: • 0: no tracing • 1-5 and 10: different levels of tracing This flag should only be set under instruction from TIBCO Technical Support.	0	No
`Usage (usg)`	Specifies the type of usage. The values supported for this parameter are as follows: • 0: non-file share user	None	No

Parameter	Description	Default	Required
	- 1: file share user - 2: mailbox user		
`UserId (uid)`	Specifies the 1-to-64-character ID to be assigned to this user. The user ID cannot contain embedded spaces. **Note:** The user ID can be defined in both uppercase and lowercase, but it will be stored in uppercase in the database.	None	Yes
`UserType (usrt)`	Specifies the type of file share user only. This parameter is only applicable if the user is a file share user. The values supported for this parameter are as follows: - 1: guest - 2: full user - 3: power user	None	No
`ValidDays (vd)`	Specifies a 7-character day-of-week pattern when the user can access the system. For example, the first character represents Sunday, and the second one represents Monday.	None	Yes

Parameter	Description	Default	Required
	Each character can be Y or N.		
`ValidEndTime (vet)`	Specifies a time in military format *HHMM* when it will no longer allow this user access.	None	Yes
`ValidStartTime (vst)`	Specifies a time in military format *HHMM* when this user can start using .	None	Yes
`Visibility (vsb)`	Specifies the visibility of the user. The valid values are as follows: • PUB: public • PRI: private	PRI	Yes

Sample `AddAdminUser` Command

This command adds a user to the user database.

```
java cfcc.CFAdmin a:AddAdminUser UserId:CenterAdmin101 FullName:"MFT Command Center
Admin" Password:101 LockFlag:N ExpirationDate:2009/12/31 Description:"MFT Command
Center Admin 101" StartDate:2005/01/03 EndDate:2006/07/01 ValidDays:NYYYYYN
ValidStartTime:1700 ValidEndTime:2100 AllowableProtocol:All
```

AddTransferUser

The `AddTransferUser` command action is used to define a user to the system.

This user will automatically be assigned the transfer right.

Parameter	Description	Default	Required
`AddPGPKey (paddk)`	Specifies whether to allow a user to add a PGP key. The valid values are as follows: - Y - N - D: default. It indicates using the default value of the **Allow users to add PGP keys** parameter in the Global PGP Settings section on the System Configuration page.	D	No
`AllowableProtocol (apl)`	Specifies the protocol that the user will be allowed to use for a file transfer. The valid values are as follows: - FTP - Secure FTP: referred as SECUREFTP. - HTTPS - CF: for TIBCO MFT Platform Server. - Secure - AS2 - SSL/TLS (for MFT Platform Server)	All	No

Parameter	Description	Default	Required
	- All: includes all listed protocols except AS2.		
`AssignedGroups (ag)`	Specifies the groups that a user has to be added to. Add groups between double quotation marks ("") and use a semicolon (;) as delimiter.	None	No
`AssignedRights (ar)`	Specifies the rights that have to be assigned to a user. A transfer right is assigned by default. Add rights between double quotation marks ("") and use a semicolon (;) as delimiter.	None	No
`CFAuthType (cfat)`	Specifies the type of authentication for CF transfers. The values supported for this parameter are as follows: - 1: password only - 2: certificate only - 3: certificate or password - 4: certificate and password	None	No
`CanChangePassword (ccp)`	Specifies whether to allow this user to change password. The valid values are as	Y	No

Parameter	Description	Default	Required
	follows: • Y • N		
`CertificateDN (dn)`	Specifies the 1-to-1024-character certificate to distinguish the name of the user.	None	No
`ChangePasswordNextLogin (cpnl)`	Specifies whether this user has to change password at the next logon. The valid values are as follows: • Y • N	Y	No
`Company Name (cname)`	Specifies the 1-to-64-character company name.	None	No
`DefaultGroup (dg)`	Specifies the group in which a user is added.	None	No
`Department (dpt)`	Specifies the department the user will be placed in.	None	No
`Description (d)`	Specifies the 1-to-256-character description for this user. If the description contains embedded spaces, the entire description must be enclosed in double quotation marks (").	None	No

Parameter	Description	Default	Required
`DisableFlag (dis)`	Specifies whether this user initially is disabled from the system. The valid values are as follows: • Y • N	N	No
`EmailAddr (ea)`	Specifies the 1-to-64-character email address of the user.	None	No
`EndDate (ed)`	Specifies the date when the account of this user will become inactive in the system. The format is *YYYY/MM/DD*. The date range is 2000/01/01 to 2099/12/31.	None	Yes
`ExpirationDate (epd)`	Specifies the date when the account of this user will expire from the system. The format is *YYYY/MM/DD*. The date range is 2000/01/01 to 2099/12/31.	None	Yes
`FTPAuthType (ftpat)`	Specifies the type of authentication for FTP transfers. The values supported for this parameter are as follows: • 1: password only	None	No

Parameter	Description	Default	Required
	• 2: certificate only • 3: certificate or password • 4: certificate and password		
`FullName (fn)`	Specifies the 1-to-256-character name for this user. If the full name contains embedded spaces, the entire full name must be enclosed in double quotation marks (").	None	Yes
`HTTPSAuthType (hhtpsat)`	Specifies the type of authentication for HTTPS transfers. The values supported for this parameter are as follows: • 1: password only • 2: certificate only • 3: certificate or password • 4: certificate and password	None	No
`IPName (ipn)`	Specifies the 1-to-64-character machine name or IP address. If the `RestrictUser` parameter is configured as Y, this parameter is	None	No

Parameter	Description	Default	Required
	required.		
`LockFlag (l)`	Specifies whether to lock the user out of the system. The valid values are as follows: • Y • N	N	No
`ManageDepartments (md)`	Specifies the departments to be managed. Each department is separated by a semicolon (;).	None	No
`MaxFileSize (mfs)`	Specifies the maximum size of a file.	None	No
`Netmask (netm)`	Specifies the 1 to 64 byte netmask.	None	No
`Password (pw)`	Specifies the 1-to-30-character password assigned to this user. The password cannot contain any embedded spaces. It is case sensitive.	None	Yes
`PasswordNeverExpires (pne)`	Specifies whether this password ever expire. This parameter overrides the global password rules. The valid values are as follows:	N	No

Parameter	Description	Default	Required
	- Y - N		
`PhoneNumber (phone)`	Specifies the 1-to-64-character telephone number.	None	No
`RestrictUser (rus)`	Specifies whether to restrict this user. The valid values are as follows: - Y - N	N	No
`SSHAuthType (sshat)`	Specifies the type of authentication for SSH transfers. The values supported for this parameter are as follows: - 1: password only - 2: certificate only - 3: certificate or password - 4: certificate and password	None	No
`StartDate (sd)`	Specifies the date when this user will be active in the system. The format is *YYYY/MM/DD*. The date range is 2000/01/01 to 2099/12/31.	None	Yes

Parameter	Description	Default	Required
`TraceLevelFlag (tf)`	Specifies the trace level of the flag. The values supported for this parameter are as follows: • 0: no tracing • 1-5 and 10: different levels of tracing This flag should only be set under instruction from TIBCO Technical Support.	0	No
`Usage (usg)`	Specifies the type of usage. The values supported for this parameter are as follows: • 0: non-file share user • 1: file share user • 2: mailbox user	None	No
`UserId (uid)`	Specifies the 1-to-64-character ID to be assigned to this user. The user ID cannot contain embedded spaces. **Note:** The user ID can be defined in both uppercase and lowercase, but it will be stored in uppercase in the database.	None	Yes
`UserType (usrt)`	Specifies the type of file	None	No

Parameter	Description	Default	Required
	share user only. This parameter is only applicable if usage is a file share user. The values supported for this parameter are as follows: • 1: guest • 2: full user • 3: power user		
`ValidDays (vd)`	Specifies a 7-character day of week pattern when the user can access the system. For example, the first character represents Sunday, the second one represents Monday. Each character can be Y or N. For example, NYYYYYN	None	Yes
`ValidEndTime (vet)`	Specifies the time in military format *HHMM* when it will no longer allow this user access.	None	Yes
`ValidStartTime (vst)`	Specifies the time in military format *HHMM* when this user can start using .	None	Yes
`Visibility (vsb)`	Specifies the visibility of the user. The valid values are as	PRI	Yes

Parameter	Description	Default	Required
	follows: • PUB: public • PRI: private		

Sample `AddTransferUser` Command

This command adds a user to the user database with the transfer right.

```
java cfcc.CFAdmin a:AddTransferUser UserId: CenterUser001 FullName:"Brian Smith -
Accounting" Password: CenterUser001 LockFlag:N ExpirationDate:2009/12/31
Description:"Brian Smith from XYZ Inc." StartDate:2005/01/03 EndDate:2006/07/01
ValidDays:NYYYYYN ValidStartTime:1700 ValidEndTime:2100 AllowableProtocol:FTP
```

ChangePassword

The `ChangePassword` command action is used to change the password for an existing user in the system.

To use the `ChangePassword` action command, you must have the permission to change passwords. If you have AdministratorRight or HelpDeskRight, you can change the password of any user; If you have the ChangePassword right, you can only change your own password. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`Password (pw)`	Specifies the 1-to-30-character password assigned to this user. The password cannot contain any embedded spaces. The password is case sensitive.	None	Yes
`UserId (uid)`	Specifies the 1-to-64-character ID of the user to be altered.	None	Yes

Sample ChangePassword Command

This command changes the password for the user ACME0001.

```
java cfcc.CFAdmin a:ChangePassword UserId:ACME0001 Password:FORGOT
```

GetUser

The GetUser command action is used to display an existing user in the system.

Parameter	Description	Default	Required
UserId (uid)	Specifies the 1-to-64-character ID of the user to be displayed.	None	Yes

Sample GetUser Command

This command displays the definition for the user User001.

```
java cfcc.CFAdmin a:GetUser UserId:User001
```

RemoveUser

The RemoveUser command action is used to delete an existing user in the system.

To use the RemoveUser action command, you must have UpdateTransferUserRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
UserId (uid)	Specifies the 1-to-64-character ID of the user to be deleted.	None	Yes

Sample RemoveUser Command

This command deletes User001 from the database.

```
java cfcc.CFAdmin a:RemoveUser UserId:User001
```

RetrieveAllUsers

The `RetrieveAllUsers` command action is used to display configuration parameters from all user definitions within the definition table of users.

To use the `RetrieveAllUsers` action command, you must have UpdateTransferUserRight. For more information, see AddUserToRole.

When this command is executed successfully, each user that is in the user definition table will be displayed along with the configuration parameters defined for each definition.

No parameters are supported for this command action.

Sample `RetrieveAllUsers` Command

This command displays information for all users.

```
java cfcc.CFAdmin a:RetrieveAllUsers
```

UpdateUser

The `UpdateUser` command action is used to alter an existing user in the system.

Note: `UpdateUser` and `AddAdminUser` commands have common parameters. For `UpdateUser` command parameters, see AddAdminUser.

Sample `UpdateUser` Command

This command updates the user User001 to allow access to the system on weekends and only from 1 AM to 9 AM.

```
java cfcc.CFAdmin a:UpdateUser UserId:User001 ValidDays:YNNNNNY ValidStartTime:0100
ValidEndTime:0900 AllowableProtocol:All
```

User Profile Commands

The user profile commands are used to define, list, and delete user profile records in the system.

Action	Description
AddUserProfile	Adds a profile for a user.
GetUserProfiles	Displays a specific user profile.
RetrieveAllUserProfiles	Displays all user profiles.
RemoveUserProfile	Deletes a user profile.
UpdateUserProfile	Updates a profile for a user.

AddUserProfile

The `AddUserProfile` command action is used to add a server credential definition to the system.

No command line actions are provided to add definitions to banks.

The user profile definition contains user ID and password information that is used when communicating with the remote Platform Server system.

When a transfer is attempted to target TIBCO MFT Platform Server, TIBCO MFT Internet Server searches the server credential database for a match on the user or group that is requesting the transfer and the target server definition. If a match can be found, TIBCO MFT Internet Server extracts the remote user ID, remote password, and remote domain. This information is then sent to the remote Platform Server system.

The advantage of using server credential definitions is that you can define all logon information in a single place. Different users can be given different logon information.

The server credential overrides the default user and default password definitions defined on the transfer and server records.

To use the `AddUserProfile` action command, you must have UpdateServerCredentialRight. For more information, see AddUserToRole.

In the following table, parameters for this command are provided in alphabetical order.

Note: The parameters provided in this table are also used for the `UpdateUserProfile` command.

Parameter	Description	Default	Required
`GroupId (gid)`	Specifies the 1-to-64-character group ID that has been defined in the group database. If the defined group is not in the group database, the request will fail. This parameter is exclusive with the `UserId` parameter. When a transfer is done, TIBCO MFT Internet Server checks all of the groups that a user is a member of to determine whether a match can be found in the user profile database. The advantage of defining a group ID user profile is that you can use a single user profile record to define user IDs and passwords for many users.	None	Either the `GroupId` or `UserId` parameter must be defined.

Parameter	Description	Default	Required
	Note: If user profiles are defined for both the `GroupId` and `UserId` parameters for a user performing a file transfer, the user ID definition will be used first.		
`NodeName (nn)`	Specifies the 1-to-32-character server name that has been defined in the server database. This parameter defines the target Platform Server definition for a file transfer. If the defined server is not in the server database, the request will fail.	None	Yes
`RemotePassword (rp)`	Specifies the 1-to-32-character remote Platform Server password. This parameter defines the password that is sent to the target Platform Server system when the file is transferred. This password must be valid on the target Platform Server system, or the file transfer request will fail. The target Platform Server system will validate the	None	Yes

Parameter	Description	Default	Required
	`RemoteUserId` parameter along with the `RemotePassword` parameter to make sure that it is valid. On some systems, such as UNIX and Windows, this parameter is case sensitive. On some other systems, such as z/OS and AS/400, this parameter is not case sensitive.		
`RemoteUserId (ru)`	Specifies the 1-to-32-character remote Platform Server user ID. This parameter defines the user ID that will be sent to the target Platform Server system when the file transfer is performed. This user ID must be defined on the target Platform Server system, or the file transfer request will fail. The target Platform Server system will validate the `RemoteUserId` parameter along with the `RemotePassword` parameter to ensure that it is valid. On some systems, such as UNIX, this parameter is case-sensitive. On some other systems, such as z/OS, AS/400, and Windows,	None	Yes

Parameter	Description	Default	Required
	it is not case-sensitive.		
`RemoteUserWinDomain (nt)`	Specifies the 1-to-256-character remote Platform Server Windows domain. This parameter is only used when the target Platform Server system runs on Windows platforms. This parameter is ignored for all other platforms. This parameter defines the domain where the remote user ID is defined. If this parameter is not defined, or is defined incorrectly, the user ID and password validation on TIBCO MFT Platform Server for Windows will fail.	None	No
`UserId (uid)`	Specifies the 1-to-64-character ID to be assigned to this user. The user ID cannot contain embedded spaces. If the defined user is not in the user database, the request will fail. This parameter is exclusive with the `GroupId` parameter. This parameter references the client user ID that is	None	The `GroupId` or `UserId` parameters must be defined.

Parameter	Description	Default	Required
	performing the file transfer request.		

Sample `AddUserProfile` Command

This command adds a user profile. That user profile is used when the user mftuser1 is communicating with the NYNode1 node. When TIBCO MFT Internet Server communicates with TIBCO MFT Platform Server, it will pass the defined `RemoteUserId`, `RemotePassword`, and `RemoteUserWinDomain` parameters to the target Platform Server system.

```
java cfcc.CFAdmin a:AddUserProfile UserId:mftuser1 NodeName:NYNode1
RemoteUserId:NYUser1 RemotePassword:NYPassword RemoteUserWinDomain:NYWinDomain
```

GetUserProfiles

The `GetUserProfiles` command action is used to display configuration parameters from a specified user profile definition from the server definition table.

No command line actions are provided to retrieve definitions from banks.

To use this command, will search for a match on the `GroupId` or `UserId` parameter and the `Server` parameter. If a match cannot be found, the request will fail.

To use the `GetUserProfiles` action command, you must have UpdateServerCredentialRight. For more information, see AddUserToRole.

When the `GetUserProfiles` command is executed successfully, the user profile is displayed along with the configuration parameters defined.

Parameter	Description	Default	Required
`GroupId (gid)`	Specifies the 1-to-64-character group ID that has been defined in the group database. For this command, a server	None	Either the `GroupId` or `UserId` parameter must be defined.

Parameter	Description	Default	Required
	credential definition with this group ID and the defined server definition must be on the server credential table; otherwise the request will fail. This parameter is exclusive with the `UserId` parameter.		
`NodeName (nn)`	Specifies the 1-to-32-character name of the server that has been defined in the server database. This parameter defines the target Platform Server definition for a file transfer. If the defined server along with the `GroupId` or `UserId` parameter is not in the user profile's database, the request will fail.	None	Yes
`UserId (uid)`	Specifies the 1-to-64-character user ID that has been defined in the user database. For this command, a server credential definition with the user ID and the defined server definition must be on the server credential table. This parameter is exclusive with the `GroupId` parameter.	None	Either the `GroupId` or `UserId` parameter must be defined.

Sample `GetUserProfiles` Command

This command displays information for the user profile for the user mftuser1 and the NYNode1 node. All parameters associated with this profile are displayed.

```
java cfcc.CFAdmin a:GetUserProfile UserId:mftuser1 NodeName:NYNode1
```

RetrieveAllUserProfiles

The `RetrieveAllUserProfiles` command action is used to display configuration parameters from all server credential definitions in the system.

No command line actions are provided to retrieve definitions from banks.

To use the `RetrieveAllUserProfiles` action command, you must have UpdateServerCredentialRight. For more information, see AddUserToRole.

When the `RetrieveAllUserProfiles` command is executed successfully, each server credential that is in the server credential table will be displayed along with the configuration parameters defined for each definition.

No parameters are supported for this command action.

Sample `RetrieveAllUserProfiles` Command

This command displays information for all server credentials.

```
java cfcc.CFAdmin a:RetrieveAllUserProfiles
```

RemoveUserProfile

The `RemoveUserProfile` command action is used to delete a predefined server credential definition.

No command line actions are provided to remove definitions from banks.

To use this command, TIBCO MFT Internet Server will search for a match on the `GroupId` or `UserId` parameter and the `Server` parameter. If a match is not found, the request will fail.

To use the `RemoveUserProfile` action command, you must have UpdateServerCredentialRight. For more information, see AddUserToRole.

Parameter	Description	Default	Required
`GroupId (gid)`	Specifies the 1-to-64-character group ID. This parameter is exclusive with the `GroupId` parameter. For this command, a server credential definition with this group ID and the defined server definition must be on the server credential table; otherwise the request will fail.	None	Either the `GroupId` or `UserId` parameter must be defined.
`NodeName (nn)`	Specifies the 1-to-32-character server name that has been defined in the server database. This parameter defines the target Platform Server definition for a file transfer. For this command, a server credential definition with this group ID and the defined server definition must be on the server credential table; otherwise the request will fail.	None	Yes
`UserId (uid)`	1-to-64-character user ID. This parameter is exclusive with the `GroupId` parameter. For this command, a server credential definition with this user ID and the defined server definition must be on the server credential table.	None	Either the `GroupId` or `UserId` parameter must be defined.

Sample RemoveUserProfile Command

This command deletes the server credential for the user mftuser1 and the NYNode1 node.

```
java cfcc.CFAdmin a:RemoveUserProfile UserId:mftuser1 NodeName:NYNode1
```

UpdateUserProfile

The `UpdateUserProfile` command action is used to change a predefined server credential definition.

No command line actions are provided to update definitions in banks.

To use this command, will search for a match on the `GroupId` or `UserId` parameter and the `Server` parameter. If a match is not found, the request will fail.

To use the `UpdateUserProfile` action command, you must have UpdateServerCredentialRight. For more information, see AddUserToRole.

ℹ **Note:** `UpdateUserProfile` and `AddUserProfile` commands have common parameters. For `UpdateUserProfile` command parameters, see AddUserProfile.

Sample UpdateUserProfile Command

This command updates a server credential for the user mftuser1 and the NYNode1 node.

```
java cfcc.CFAdmin a:UpdateUserProfile UserId:mftuser1 NodeName:NYNode1
RemoteUserId:NYUser2 RemotePassword:NYPassword123 RemoteUserWinDomain:NYWinDomain
```

Miscellaneous Commands

The commands retrieve system information from the system.

Action	Description
GetCopyrightInfo	Displays copyright information.

Action	Description
	Note: This command is not supported when using REST web service.
GetProductNameVersion	Gets version information.

GetCopyrightInfo

The `GetCopyrightInfo` command action is used to display copyright information about .

No parameters are supported for this command action.

Sample `GetCopyrightInfo` Command

This command displays the copyright information.

```
java cfcc.CFAdmin a:GetCopyrightInfo
```

Note: This command is not supported when using REST web service.

GetProductNameVersion

The `GetProductNameVersion` command action is used to display version information about .

No parameters are supported for this command action.

Sample `GetProductNameVersion` Command

This command displays the version of the product.

```
java cfcc.CFAdmin a:GetProductNameVersion
```

Help

The `Help` command action is used to get information on the commands that are used by Admin Client Utility.

You might enter the following command:

```
java cfcc.CFAdmin help:xxxxxxx
```

The field *xxxxxxxx* must match one of the command actions.

Sample `Help` Command

This command lists all parameters supported by the `AddGroup` command action.

```
java cfcc.CFAdmin help:addgroup
```

You will receive the following output.

```
Please provide following parameters via command line or in action file:
GroupId --- group id
Description --- group description
Department --- Group's department. The value is ignored for department
admin
Visibility --- Group's visibility; PUB-public, PRI-private
```

Action File (Admin Client Utility)

The action file is an XML file specified by the `T` parameter on the command line. By using an action file, you can put multiple actions in one file specified using XML format.

The format of the action file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE actions SYSTEM "siftactions.dtd">
<actions>
    <action name="action1" output=”action2:file1”>
        <arg name="arg1" value="somevalue" sc="a1"/>
        ……
    </action>
    ……
</actions>
```

The <action> element defines an action. The <arg> element defines a parameter needed for this action. If multiple <action> elements are defined in the file, the program will execute them one by one.

The name attribute for the <action> element specifies the action name. This action name must be a valid action. The XML file names are all valid actions.

The name attribute for the <arg> element specifies the parameter name for an action. The name is case-sensitive and cannot be edited. The sc attribute for the <arg> element specifies a shortcut name for the name attribute, and it is case-insensitive. You can use shortcut names to specify values in the command line to replace the default values specified in this file. If the action is specified by the `A` parameter in the command line, you must specify a parameter name for that action rather than a shortcut name. Shortcut names can be found in each XML file.

For actions that retrieve information from a web service, you can specify an output file in the output attribute for the <action> element. The program will save the retrieved information into the file in action file format. This file can be used as an action file.

Sample Action File

If you want to add user B to the database, and user A whose information can be used for user B already exists in the database, you can perform the following operations:

1. Build an action file `userA.xml` to retrieve the information of user A, and save the information into an `addUser` command action in the file `userB.xml`.

   The syntax of the `userA.xml` file will be as follows:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <!DOCTYPE actions SYSTEM "siftactions.dtd">
   <actions>
       <action name=”getUser” output=”addUser:userB.xml”>
           <arg name=”UserId” value=”userA” sc=”UID”/>
       </action>
   </actions>
   ```

   The value for the output attribute is `ActionName:`*`FileName`* or `ActionName>`*`FileName`*. Because the generated file is in action file format, both the action name and file name are needed. Use a colon (:) to generate a new output file, or use the greater than symbol (>) to append to an existing file.

2. Run the program to get the information of user A, and generate an action file `userB.xml`.

```
java –classpath %cp% cfcc.CFAdmin U:userA P:pwdA T:userA.xml
```

3. Run the program again with the generated action file to add user B.

   ```
   java –classpath %cp% cfcc.CFAdmin U:userB P:pwdB T:userB.xml UID:userB
   ```

4. Use `UID:userB` to overwrite the `UserId` parameter from the action file, in which the value is userA.

Currently, `GetTransfer`, `GetGroup`, `GetServer`, `GetUser`, and `GetUserProfile` command actions support writing output into an XML file. The sample XML files included in the product create the `afTmpl.xml`, `agTmpl.xml`, `anTmpl.xml`, `asTmpl.xml`, `auTmpl.xml`, and `aupTmpl.xml` files respectively.

Shortcuts Usage in the Action File

The advantage of using an action file template is that you can use shortcuts to define the parameter names.

An example of command line using shortcuts is as follows:

```
java cfcc.CFAdmin U:xyz P:xyz KN:certificate KP:pswd a:addFile
CFN:clientfile.txt SFN:serverfile.txt UID:user1 AuthGroupId:TransferRight
TKN:cacerts TKP:changeit
```

You can change the shortcut names. The shortcut names defined in the XML template are the default shortcut names. In the above text, the `CFN` parameter is defined as the shortcut name for the `ClientFileName` parameter. You can change this value to any value that you want, as long as the value does not conflict with an existing parameter name or shortcut value. For example, you can use a text editor to change the value `CFN` to `CN`. Therefore, you can use the value `CN` in the command line to reference the `ClientFileName` parameter whenever you use that XML template file.

If the `Global.xml` file has been updated to contain the user ID, password, and keystore information, you can simply execute the following command line.

```
java cfcc.CFAdmin a:addFile
```

For client certificate authentication, the client must specify the keystore for its certificate via the Java system parameter, or via the `KN` and `KP` parameters of the command line. To run the program over an SSL connection, the certificate authority (CA) that signed the certificate of the client must be a trusted CA. This might require you to update your keystore.

Note: The batch file used to set up classpath overwrites the default system classpath. Experienced users are encouraged to use other environment variables for classpath, and specify classpath in the Java command.

Name	Description
U	The user ID is sent to the web service for authentication to use the web service. This parameter might be specified in the `Global.xml` file.
P	The user password is sent to the web service for authentication to use the web service. This parameter might be specified in the `Global.xml` file.
A	The action to take. For example, add file. If the parameter is specified, the program will ignore the `T` parameter that specifies the action file name. The program only accepts one action from the command line.
T	The action file name. The file can contain multiple actions in XML format. The program will execute all actions specified in the file. If the program specified the `A` parameter, this parameter will be ignored.
TL	The trace level. This value only affects this utility. This parameter should only be set when instructed to do so by TIBCO technical support. The valid value range is 0 to 10.
TD	The trace directory. This value only affects this utility. Sets the directory where the trace files will be written.
G	The global template file name. The default one is `Global.xml` in the current directory.
S	The web service address. For example, https://ip:port/cfcc/…..
KN	The Java keystore name for client certificate authentication. The keystore name can be specified as a Java parameter, in

Name	Description
	which case, it is not necessary to use this parameter again. This parameter might be specified in the `Global.xml` file.
`KP`	The Java keystore password for client certificate authentication. The keystore password can be specified as a Java parameter, in which case, it is not necessary to use this parameter again. This parameter might be specified in the `Global.xml` file.
`TKN`	The trusted Java keystore name for certificate authentication. This file should contain the name of the keystore file that contains the Java trusted certificate authorities. You can leave this parameter blank if you want to use the default trusted keystore. This parameter might be specified in the `Global.xml` file.
`TKP`	The trusted Java keystore password for client certificate authentication. If the default password is used, you can leave this parameter blank. This parameter might be specified in the `Global.xml` file.
`help`	The program will display the command line parameter list.
`help:action`	The program will display the parameters needed for the action if the action is a valid action; otherwise, the program will display all currently supported actions.
`name:value`	Other `name:value` pairs. These values will be used to assign the values of parameters if the action is specified by the `A` parameter, or to replace the default values if the `T` parameter is used. The name is case sensitive if *name* is a parameter name for an action. The name is not case sensitive if *name* is a shortcut for a real parameter name.

In the following example, four entries are defined in the `addFile.xml` file.

```
<arg name="ClientFileName" value="clientFileName" sc="CFN"
description="Client File Name"/>
```

```
<arg name="ServerFileName" value="serverFileName" sc="SFN"
description="Server File Name"/>
<arg name="Description" value="fileDesc" sc="D" description="File
Description"/>
<arg name="UserId" value="user id" sc="UID" description="UserID
authorized to transfer this file"/>
```

Note: The parameter that starts with the value sc= is the shortcut name that has been defined by the XML file. When executing Admin Client Utility with the action file parameter (`T:`) defined, you can use the shortcut name instead of the actual parameter name. For example in the above example, when defining the client file name, you can use the `CFN` parameter instead of the `ClientFileName` parameter.

Sample Shortcuts Usage

The following examples show describe the process of using standard parameters and shortcuts in the commands:

Using standard parameter names:

```
java cfcc.CFAdmin a:addFile ClientFileName:client.file1
ServerFileName:prod.file.name Description:”file upload” Userid:acctuser
```

Using shortcut parameter names:

```
java cfcc.CFAdmin t:addFile.xml CFN:client.file1 SFN:prod.file.name D:”file
upload” uid:acctuser
```

The parameter names are much shorter when using the shortcut parameters. The shortcut parameter names can only be used when the action file template (`T:`) parameter is used in the `CFAdmin` command. The shortcut values must be defined by the sc= value in the template.

CFAdmin XML Files

The `genExample` command is run as part of the Config program. You can create various XML files that can be used in conjunction with the command line.

If you choose not to run this when running the Config program, it can be run any time using the following command:

```
java cfcc.CFAdmin genExample
```

This will create various XML files that can be used in conjunction with the command line. The following table contains the names of the XML files that are created and their brief description.

Audit XML files	
`GetAudit.xml`	Displays a specific audit record.
`RemoveAudit.xml`	Removes an audit record.
`SearchForAudits.xml`	Searches for audit records.
Department XML files	
`AddDepartment.xml`	Creates a department.
`GetDepartment.xml`	Displays a department.
`RemoveDepartment.xml`	Deletes a department.
`RetrieveAllDepartments.xml`	Displays all departments.
`RetrieveAllUsersInDept.xml`	Displays users assigned to this department.
`UpdateDepartment.xml`	Updates a department.
Group XML files	
`AddGroup.xml`	Defines a group.
`AddUserToGroup.xml`	Adds a user to a group.
`GetGroup.xml`	Displays a group.
`RemoveGroup.xml`	Deletes a group.
`RemoveUserFromGroup.xml`	Deletes a user from a group.

`RetrieveAllGroups.xml`	Displays all groups.
`RetrieveAllGroupsForUser.xml`	Displays groups that the user is a member of.
`RetrieveAllUsersInGroup.xml`	Displays all users in a group.
Role XML files	
`AddUserToRole.xml`	Adds a right to a user.
`GetRole.xml`	Displays a right.
`RemoveUserFromRole.xml`	Removes a right from a user.
`RetrieveAllRoles.xml`	Displays all rights.
`RetrieveAllRolesForUser.xml`	Displays the rights assigned to a user.
`RetrieveAllUsersInRole.xml`	Displays users that have a specific right.
Server XML files	
`AddServer.xml`	Creates a server.
`GetServer.xml`	Displays a server.
`RemoveServer.xml`	Deletes a server.
`RetrieveAllServers.xml`	Displays all servers.
`UpdateServer.xml`	Updates a server.
Session XML files	
`DeleteExpiredSessionIds.xml`	Deletes all expired session IDs.
`DeleteSessionId.xml`	Deletes a session ID.
`GetExpiredSessionIds.xml`	Lists expired session IDs.

`ListActiveSessionIds.xml`	Lists active session IDs.
Transfer XML files	
`AddTransfer.xml`	Adds a transfer definition.
`GetTransfer.xml`	Lists a specific transfer definition.
`RemoveTransfer.xml`	Deletes a transfer definition.
`RetrieveAllTransfers.xml`	Lists all transfer definitions.
`RetrieveAllTransfersForUser.xml`	Lists all transfer definitions for a user.
`SearchForTransfers.xml`	Searches for transfer records.
`UpdateTransfer.xml`	Alters a transfer definition.
User XML files	
`AddAdminUser.xml`	Adds a user with administrator rights.
`AddTransferUser.xml`	Adds a user with transfer rights.
`ChangePassword.xml`	Changes a user's password.
`GetUser.xml`	Displays a specific user.
`RemoveUser.xml`	Deletes a user.
`RetrieveAllUsers.xml`	Displays all users.
`UpdateUser.xml`	Updates a user.
User Profile XML files	
`AddUserProfile.xml`	Adds a profile for a user.
`GetUserProfile.xml`	Displays a specific user profile.

`RemoveUserProfile.xml`	Deletes a user profile.
`RetrieveAllUserProfiles.xml`	Displays all user profiles.
`UpdateUserProfile.xml`	Updates a profile for a user.
Miscellaneous XML files	
`GetCopyrightInfo.xml`	Displays copyright information.
`GetProductNameVersion.xml`	Gets the version information.
PGP Public Key XML Files	
`AddPGPPublicKey.xml`	Adds a PGP public key.
`DeletePGPPublicKey.xml`	Deletes a PGP public key.
`GetPGPPublicKey.xml`	Displays a PGP public key.
`UpdatePGPPublicKey.xml`	Updates a PGP public key.
`RetrievePGPPublicKey.xml`	Retrieves a PGP public key.
Protocol Public Key XML Files	
`AddProtocolPublicKey.xml`	Adds a protocol public key.
`DeleteProtocolPublicKey.xml`	Deletes a protocol public key.
`GetProtocolPublicKey.xml`	Gets a protocol public key.
`UpdateProtocolPublicKey.xml`	Updates a protocol public key.
`RetrieveProtocolPublicKey.xml`	Retrieves a protocol public key.

Promotions Utility

The Promotion Utility is designed for the end-user to copy definitions from one TIBCO MFT system to another TIBCO MFT system using the GUI mode that can run on Windows or UNIX with a GUI interface or the command line mode. It has the following features, components, and modes.

Features:

- Promotes components from one MFT system to another MFT System (i.e. from MFT systems connected to different databases)
- Works when the MFT systems are at different levels
- Promotes from a lower version to a higher version
- Promotes from a higher version to a lower version

Components:

- Transfers
- Users
- Servers
- Departments
- Groups
- Protocol and PGP Public Keys
- Platform Transfers

Modes:

- GUI mode that can run on Windows or UNIX with a GUI interface
- Command line mode

Installing and Configuring Promotion Utility (GUI mode)

You can install and configure MFT Promotion Utility in the GUI mode.

Before you begin

You must set the *JAVAFX_HOME* environment variable before you install and configure the Promotions Utility.

> **Note:** If you are using Java 8, this prerequisite is not applicable and you do not have to set up the *JAVAFX_HOME* environment variable.

1. Go to URL: https://openjfx.io and download JavaFX.

   > **Note:** If you are using Java 11 or higher, download JavaFX because Oracle does not ship JavaFX with Java.

2. Extract the contents of the downloaded JavaFX zip file to a new directory.
3. Set an environment variable with the name *JAVAFX_HOME* that points to the runtime directory.
   - For Linux:

     ```
     export JAVAFX_HOME=/path/to/javafx-sdk-12.0.2
     ```

   - For Windows

     ```
     set JAVAFX_HOME="\path\to\javafx-sdk-12.0.2"
     ```

Procedure

1. Download MFT Promotion Utility from the following location and save it.

   `MFT-Install/distribution/MFTPromotionUtility/MFTPromotion.zip`

2. Unzip `MFTPromotion.zip` to a new directory. For example:

   `c:\MFTPromote`

3. Choose one of the following ways to start the MFT Promotion Utility GUI and use the

GUI mode:

- Through Windows Explorer

  Navigate to the following folder:

  `c:\MFTPromote\bin`

  Open the following file:

  - `promoteGUI.bat` (or `promoteGUI.sh` on UNIX) if you are using JAVA 11 or higher
  - `promoteGUI-java8.bat` (or `promoteGUI-java8.sh` on UNIX) if you are using JAVA 8

- Through a DOS prompt

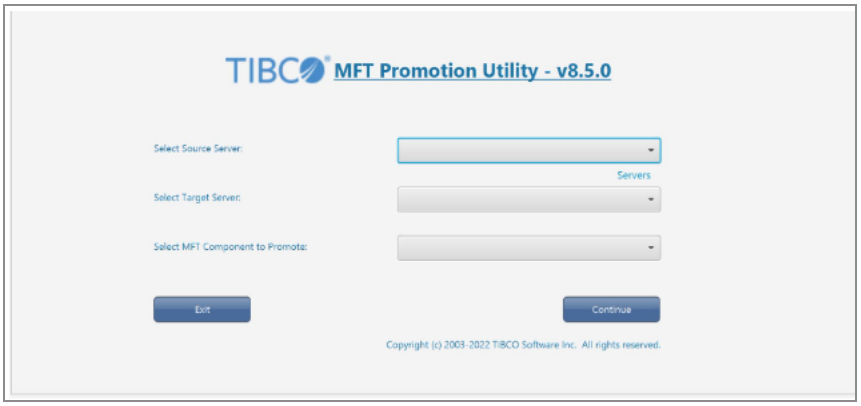
  Enter the following command to change the directory:

  cd \MFTPromote\bin

  Enter the following command:

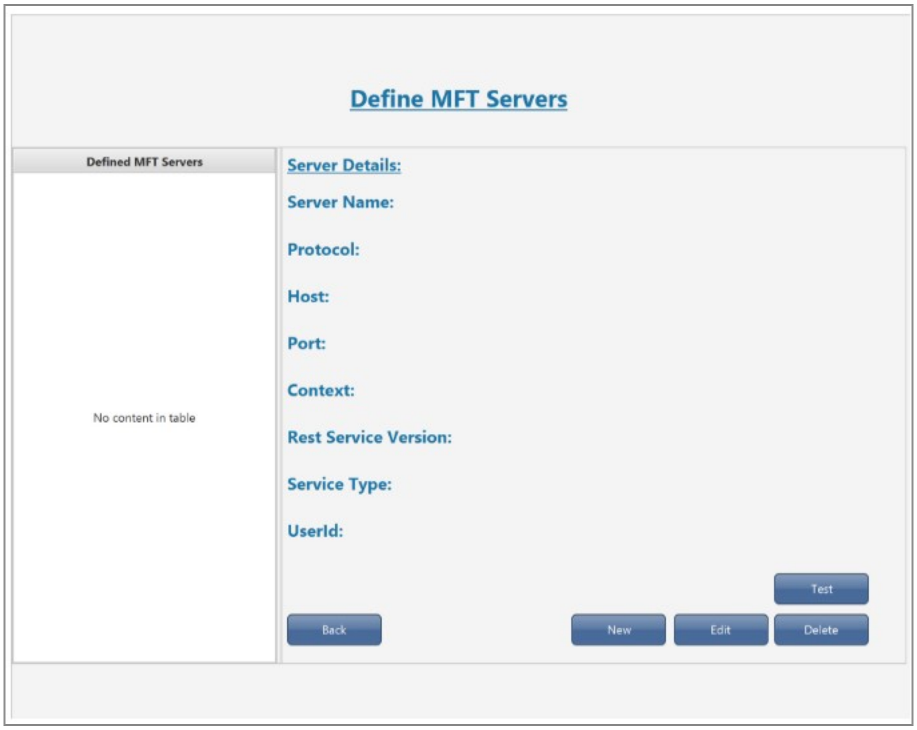
  - promoteGUI.bat (or promoteGUI.sh on UNIX) if you are using JAVA 11 or higher
  - promoteGUI-java8.bat (or promoteGUI-java8.sh on UNIX) if you are using JAVA 8

The MFT Promotion Utility main screen is displayed.


TIBCO MFT Promotion Utility - v8.5.0
Select Source Server:
Servers
Select Target Server:
Select MFT Component to Promote:
Exit
Continue
Copyright (c) 2003-2022 TIBCO Software Inc. All rights reserved.

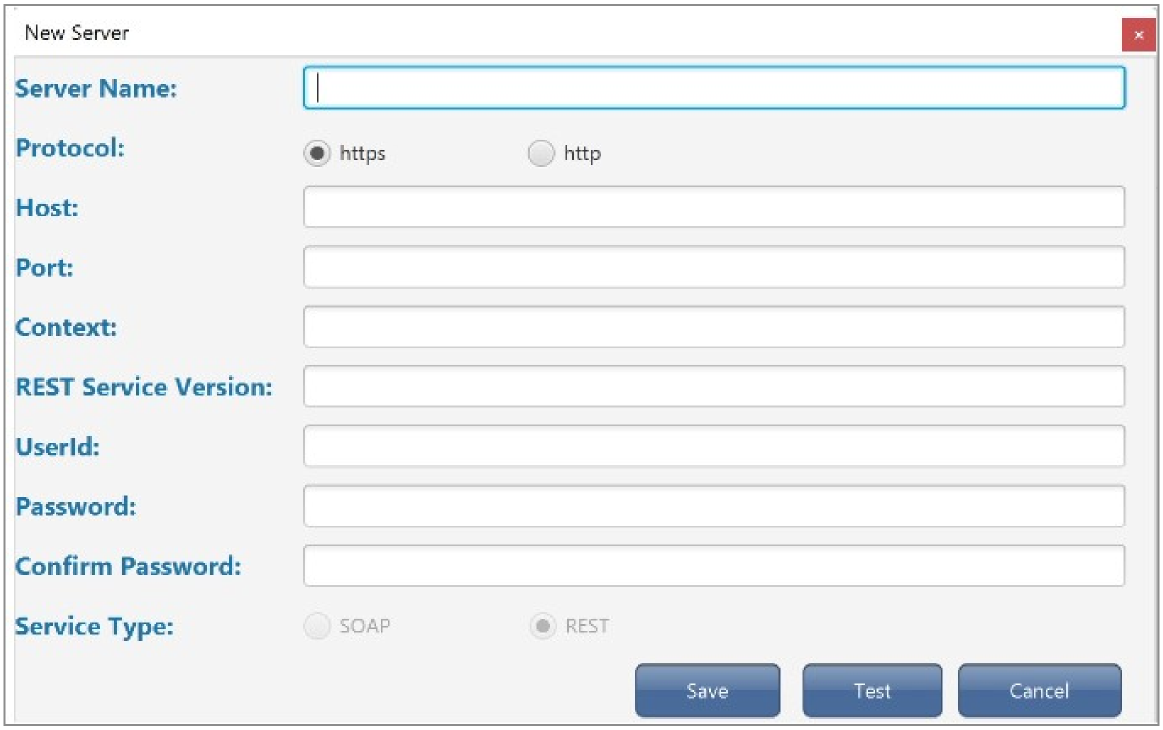
4. On the main screen, click **New Server** to create configuration entries for your source and target servers.

   The Define MFT Servers screen is displayed.



5. On the Define MFT Servers screen, click **New**.

   The New Server screen is displayed.



New Server

Server Name:

Protocol: ● https ○ http

Host:

Port:

Context:

REST Service Version:

UserId:

Password:

Confirm Password:

Service Type: ○ SOAP ● REST

6. Enter the details in the New Server screen and click **Test**.

   The details entered are then validated by connecting and authenticating to the required server.

7. If the test is successful, click **Save** to save the server.

8. Repeat steps 1-5 to save a second server.

   The MFT Promotion Utility must have two servers, that is, a source server and a target server, defined and saved to promote components.

Promoting Records (GUI mode)

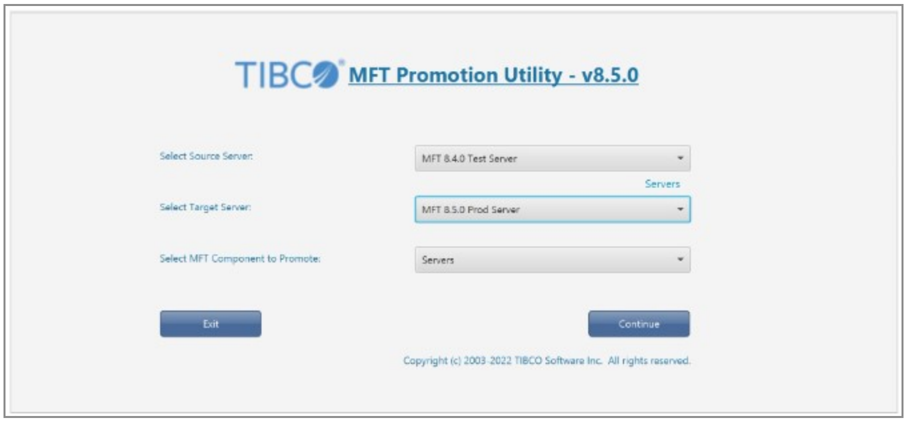
After two servers have been defined, you can promote records from a source to a target server. In this mode, you can filter and promote 100 records at a time. You can use this mode on Windows or UNIX with a GUI interface.

Before you begin

Start the MFT Promotion Utility and define the source and the target servers. See Installing and Configuring the Promotion Utility (GUI Mode).

Procedure

1. In the MFT Promotion Utility main screen, select the source server, the target server, and the component you want to promote.

   MFT Promotion Utility - v8.5.0

   Select Source Server: MFT 8.4.0 Test Server

   Select Target Server: MFT 8.5.0 Prod Server

   Select MFT Component to Promote: Servers

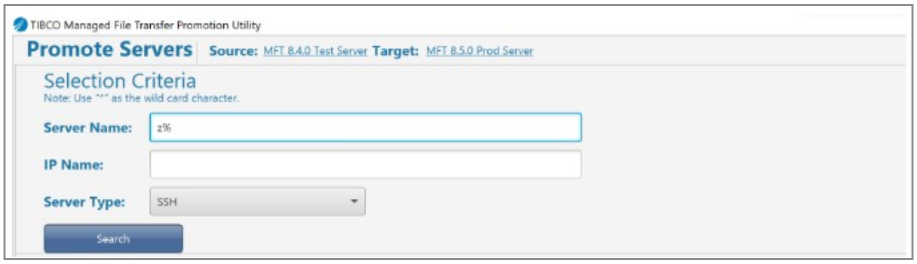
   Copyright (c) 2003-2022 TIBCO Software Inc. All rights reserved.

2. Click **Continue**.

   The Promote Servers screen is displayed.

3. Enter the selection criteria to filter records.

   > **Note:** You can use the % wildcard character to filter requests. For example, as shown below, you can filter for SSH Servers that start with the letter "z".

   TIBCO Managed File Transfer Promotion Utility

   Promote Servers Source: MFT 8.4.0 Test Server Target: MFT 8.5.0 Prod Server

   Selection Criteria

   Note: Use "*" as the wild card character.

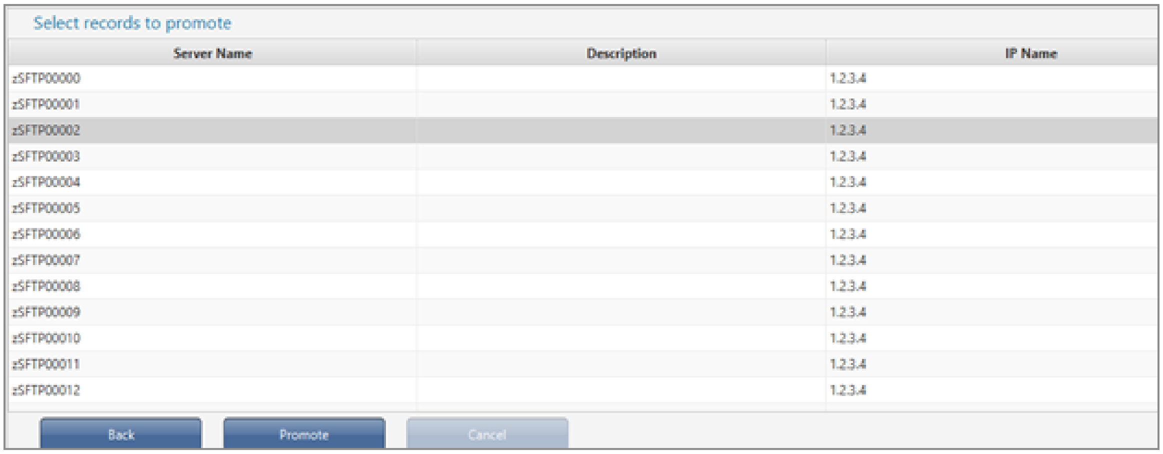
   Server Name: z%

   IP Name:

   Server Type: SSH

4. Click **Search**.

   A list of all the servers that match the selection criteria is displayed.



Select records to promote

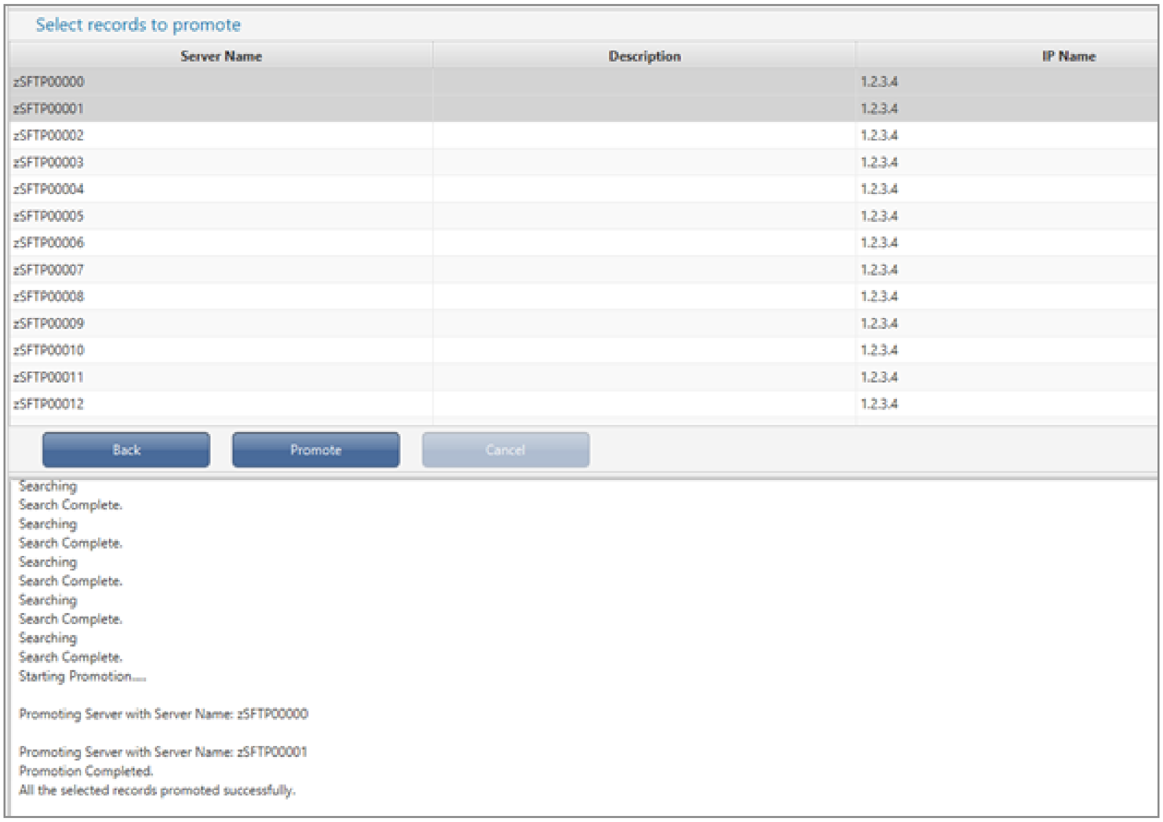
Server Name	Description	IP Name
zSFTP00000		1.2.3.4
zSFTP00001		1.2.3.4
zSFTP00002		1.2.3.4
zSFTP00003		1.2.3.4
zSFTP00004		1.2.3.4
zSFTP00005		1.2.3.4
zSFTP00006		1.2.3.4
zSFTP00007		1.2.3.4
zSFTP00008		1.2.3.4
zSFTP00009		1.2.3.4
zSFTP00010		1.2.3.4
zSFTP00011		1.2.3.4
zSFTP00012		1.2.3.4

5. Click to select the records you want to promote.

   You can select continuous rows using the Shift key and non-continuous rows using the Ctrl key. You can select 1-100 records to promote at a time.

6. Click **Promote** to promote the records.

   The status of the promotion is shown in the box below.



Select records to promote

Server Name	Description	IP Name
zSFTP00000		1.2.3.4
zSFTP00001		1.2.3.4
zSFTP00002		1.2.3.4
zSFTP00003		1.2.3.4
zSFTP00004		1.2.3.4
zSFTP00005		1.2.3.4
zSFTP00006		1.2.3.4
zSFTP00007		1.2.3.4
zSFTP00008		1.2.3.4
zSFTP00009		1.2.3.4
zSFTP00010		1.2.3.4
zSFTP00011		1.2.3.4
zSFTP00012		1.2.3.4

```
Searching
Search Complete.
Searching
Search Complete.
Searching
Search Complete.
Searching
Search Complete.
Searching
Search Complete.
Starting Promotion.....

Promoting Server with Server Name: zSFTP00000

Promoting Server with Server Name: zSFTP00001
Promotion Completed.
All the selected records promoted successfully.
```

Installing and Configuring Promotion Utility (CLI mode)

You can install the MFT Promotion Utility in the CLI mode. In this mode, you can use the config.bat or config.sh utilities to configure entries. You can also use any configuration already created in the GUI mode.

Procedure

1. Download MFT Promotion Utility from the following location and save it.

   `MFT-Install/distribution/MFTPromotionUtility/MFTPromotion.zip`

2. Unzip `MFTPromotion.zip` to a new directory. For example:

   `c:\MFTPromote`

3. Create a DOS prompt.

4. Enter the following command to change directory:

   cd \MFTPromote\bin

5. Enter the following command to start the config utility and use MFT Promotion Utility in the CLI mode.

   config (or ./config.sh on UNIX)

Result

The MFT Promotion CLI config utility is displayed. You can now configure entries in the CLI mode, such as add, delete, update, or list definitions of components.

Promoting Records (CLI mode)

After two servers have been defined, you can promote records from a source to a target server. In this mode, you can promote one record at a time. This mode can run on any system with a supported Java. It is intended to be used by a back-end business process.

Before you begin

You must start the MFT Promotion Utility and define the source and the target servers. See Installing and Configuring the Promotion Utility (CLI Mode).

Procedure

1. To get general help on promoting components using the CLI, enter the command:

   promote help

   The following information is displayed.

```
MFT Promotion Utility 8.5.0
Copyright (c) 2003-2022 Cloud Software Group, Inc.  All rights
reserved.
Usage:
The MFT Promotion Utility allows you to promote definitions from
one MFT database to another MFT database.
The format of the Promotion Utility Command Line is:
Promote source:[source server] target:[target server] component:
[component type] id:[component id or name] [optional parameters]
source:    Defines the MFT Server where the definition is retrieved
target:    Defines the MFT Server where the definition is added
id:        Defines the id or name of the component.
component: Defines the component to be promoted.
           The following components are supported:
           : server
           : transfer
           : platformtransfer
           : serverkey
           : department
           : user
           : group
           : userkey

Ex. Promote source:oldServer target:newServer component:user
id:UserA

For additional help information, enter one of the following
commands:
Promote -help server            ==> displays help for component
server
Promote -help transfer          ==> displays help for component
transfer
Promote -help platformtransfer  ==> displays help for component
platformtransfer
Promote -help serverkey         ==> displays help for component
serverkey
Promote -help department        ==> displays help for component
department
```

```
Promote -help user              ==> displays help for component
user
Promote -help group             ==> displays help for component
group
Promote -help userkey           ==> displays help for component
userkey
```

2. To get help on promoting a specific component, enter the following command:

   Promote help *component*

   For example, to get help on promoting servers, enter the following command:

   Promote help server

   The following information is displayed.

```
MFT Promotion Utility 8.5.0
Copyright (c) 2003-2022 Cloud Software Group, Inc.  All rights
reserved.
Usage:
Promote server allows you to promote servers from one MFT server to
another.
Because passwords are not promoted, you can update the passwords by
including the password parameters, or you can update the server in
the target server using the browser admin or command line utility.

The format of the Promotion Utility Command Line for server is:
Promote source:[source server] target:[target server]
component:server id:[server name] pwd:[default password] proxyPwd:
[proxy password] DNIPwd:[DNI password]
source:    Defines the MFT Server where the definition is retrieved
target:    Defines the MFT Server where the definition is added
id:        Defines the server name
component: Sets the component as server
pwd:       Defines the Default Password (Optional)
proxyPwd:  Defines the Proxy Password (Optional)
DNIPwd:    Defines DNI Password (Optional)
Ex. Promote source:oldServer target:newServer component:server
id:serverA pwd:DefaultPassword proxyPwd:ProxyPassword
DNIPwd:DNIPassword
```

Appendix A. Command Line Manual Configuration

This appendix describes how to manually configure the `Global.xml` file for both Admin Client Utility and Platform Transfer Client Utility, as well as how to create the keystore in order for the command line utility to function properly on any Windows or UNIX machine.

These instructions are given as an alternative to running the configuration program described in the Command Line Utilities.

- Administrator Global Settings
- File Transfer Global Settings
- Java Keystores Settings
- Environment Settings

Administrator Global Settings

Admin Client Utility can utilize the `Global.xml` file to hold parameters that are required for all commands.

By setting these values in the global, it eliminates the need to specify them each time you run the utility. The following command line parameters can be configured in the `Global.xml` file:

- Service: the URL and service type of the Admin Client Utility service.
- U: the user ID under which the utility changes are performed.
- P: the password for the user ID.
- KN: the Java keystore name.
- KP: the Java keystore password.
- TKN: the trusted Java keystore name.
- TKP: the trusted Java keystore password.

Perform the following steps to configure the administrator global settings:

1. To edit the `Global.xml` file, you can use the following editors:
    - For Windows: Notepad
    - For UNIX: vi
2. To add the service address and service type, locate the following lines in the `Global.xml` file.

    ```
    <!-- default service address -->
    <msg name="service" value=""/>
    <!-- servicetype (SOAP/REST) -->
    <msg name="servicetype" value="REST"/>
    ```

3. Modify the value attribute to specify the location of your service.

    For example:

    ```
    <!--  default service address -->
    value="https://YOUR.SERVER.HERE:8443/ContextName/rest/admin/v<REST
    VERSION>"/>
    ```

    Note: The service address and service type must be added between double quotation marks (" "). The REST version for MFT 8.4 is v4.

4. Repeat these changes for the user ID, password, keystore name, keystore password, trusted keystore, and trusted keystore password.

    For example:

    ```
    <!-- default user id -->
    <msg name="userid" value="USERID"/>
    <!-- default user pwd -->
    <msg name="userpwd" value="PASSWORD"/>
    <!-- the encrypted user password, if has value, will overwrite
    userpwd -->
    <msg name="encrypteduserpwd" value="9abe8f97ebf00295" />
    <!-- default java keystore name -->
    <msg name="jksname" value="C:\keystore\cacerts"/>
    <!-- default java keystore password -->
    ```

```
<msg name="jkspwd" value="changeit"/>
<!-- encrypted java keystore password -->
<msg name="encryptedjkspwd"
value="48d938b0ba29fb4d0b47bb121441a37f"/>
<!-- default trusted java keystore name -->
<msg name="trustedjksname" value="C:\keystore\cacerts"/>
<!-- default trusted java keystore password -->
<msg name="trustedjkspwd"
value="0a095e1e7ff74c8e8cdfc5e73ab442f4"/>
<!-- encrypted trusted java keystore password -->
<msg name="encryptedtrustedjkspwd" value=""/>
```

5. If you do not want clear text passwords stored in the `Global.xml` file, you can use Config Utility to generate encrypted keys in this file.

File Transfer Global Settings

Platform Transfer Client Utility utilizes the `Global.xml` file to hold parameters that are required for all commands.

By setting these values in the global, it eliminates the need to specify them each time you run the utility. The following command line parameters might be configured in the `Global.xml` file:

- Service: The URL of the Platform Transfer Client Utility service.
- U: The user ID under which the utility changes will be performed.
- P: The password for the user ID.
- KN: The Java keystore name.
- KP: The Java keystore password.
- TKN: The trusted Java keystore name.
- TKP: The trusted Java keystore password.
- AD: The audit directory.

You can configure the administrator global settings in following steps:

1. To edit the `Global.xml` file, you can use following editors:
    - For Windows: Notepad

   - For UNIX: vi

2. To add the service address, locate the following lines in the `Global.xml` file.

   ```
   <!-- default service address -->
   <msg name="service" value=""/>
   ```

3. Modify the value attribute to specify the location of your service.

   For example:

   ```
   <!-- default service address -->
   <msg name="service" value="https://MFT Command
   Center.MYCOMPANY.COM:8443/cfcc/control?view=services/FTService"/>
   ```

   > **Note:** Make sure that the service address is added between the double quotation marks (“).

4. Repeat these changes for the audit directory, user ID, password, keystore name, keystore password, trusted keystore, and trusted keystore password.

   For example:

   ```
   <!-- default user id -->
   <msg name="userid" value="admin"/>
   <!-- default user pwd -->
   <msg name="userpwd" value="admin"/>
   <!-- default java keystore name -->
   <msg name="jksname" value="D:\keystore\mykeystore.jks"/>
   <!-- default java keystore password -->
   <msg name="jkspwd" value="changeit"/>
   <!-- default trusted java keystore name -->
   <msg name="trustedjksname" value="D:\keystore\cacerts"/>
   <!-- default trusted java keystore password -->
   <msg name="trustedjkspwd" value="changeit"/>
   <!-- default audit directory -->
   <msg name="auditdirectory" value=""/>
   ```

5. If you do not want clear text passwords stored in the `Global.xml` file, you can use Config Utility to generate encrypted keys in this file.

Java Keystores Settings

TIBCO MFT Internet Server supports the use of two Java keystores. The file names for both keystores are defined in the `Global.xml` file.

The `trustedjksname` file defines the certificate authorities that this Java client will trust when performing the initial handshake. The `jksname` file defines the certificate that will be used when the web server is defined to require client certificates.

Both the types of certificate files will now be discussed. Included in the discussion is an explanation of what the file is used for, when it should be used and how to update or create it.

- The Java Trusted Authority Certificate File
- The Java Certificate File
- The SSH Java Certificate Keystore

The Java Trusted Authority Certificate File

The `trustedjksname` parameter defines the file that contains the list of certificate authorities that are trusted when validating a certificate.

All certificates are issued by certificate authorities (CA). When you want to validate a certificate, in addition to validating the certificate itself, ensure that the CA that issued the certificate is also valid.

By default, Java has a `trustedjksname` file that contains a group of common certificate authorities. The file name is cacerts and this file is contained in the JRE runtime library under the …`lib/security` directory. In many, if not most cases, the certificate authorities that are contained in the default Java certificate file are sufficient, and no further work has to be done. In this case, you can let the `trustedjksname` parameter default. Java will then pick up its default trusted certificate authority file called: …`lib/security/cacerts`. You should however, specify the `trustedjkspwd` file to define the password of the default certificate file. This can be done in clear text in the `Global.xml` file or encrypted by the `EncryptPassword` action command.

In cases where the server certificate was not issued by one of the default trusted authorities, add the server certificate to the Java trusted certificate authority file (cacert). To do this, the server CA certificate must be in Base64 format. Then you can issue the following Java command to add this certificate to the trusted certificate authority file:

```
keytool -import
         -keystore c:\program files\java\jre1.8.0_
66\lib\security\cacerts
         -alias MFTCommandCenterServerKey
         -file cacert.file
         -storepass changeit
```

Note: This command should be typed as a single line.

- `-keystore`: specifies the name and location of a keystore.

  It must point to the default Java keystore.

- `-alias`: specifies the unique name for this certificate key.

  If you do not specify this parameter, a default value of mykey is assigned.

- `-file`: specifies the certificate file name in Base64 format.

- `-storepass`: specifies the password for the cacerts keystore.

  This parameter is the password that you must configure as `trustedjskpwd` within the `Global.xml` file. The default password is changeit.

After entering the command, you are prompted to confirm the request. After confirming the request, the certificate will be added to the trusted certificate authority file. Now, when your client makes a request to the server, the certificate of the server will authenticate correctly.

The Java Certificate File

When communicating with a web server that requires client certificates, you must configure the `jskname` and `jskpwd` parameter in the `Global.xml` file.

If you have a Java keystore that contains the client certificate, you must define the `jskname` parameter to point to the Java keystore file that contains the client certificate, and define the `jskpwd` parameter to specify the password for the keystore.

If the web server does not require client certificates, use the default values for `jskname` and `jskpwd` parameters. You do not have to create any Java keystores or define the `jskname` and `jskpwd` parameters in the `Global.xml` file.

When the web server requires a Java certificate and you do not have a Java keystore that contains a Java certificate, you will have to create one. The Java keystore is typically created in the home directory of the user, however it can be created in any directory.

To create a Java keystore, you must execute the following command:

```
keytool -genkey {-alias alias} [-dname dname] [-keypass keypass]
    {-keystore keystore} [-storepass storepass] [-keyalg rsa]
```

Note: This command should be typed as a single line.

- `-alias`: specifies the unique name for this certificate chain and the private key in this new keystore entry.

  If you do not specify this parameter, a default value of mykey will be assigned.

- `-dname`: specifies the X.500 distinguished name to be associated with alias.

  This parameter is used as the issuer and subject fields in the self-signed certificate. You must set the common name (`CN=`) to the host or IP name of client. The name will be used to access the server.

  If no distinguished name is provided at the command line, the user will be prompted for one.

- `-keypass`: specifies a password used to protect the private key of the generated key pair.

  If no password is provided, the user is prompted for it. If you press Enter at the prompt, the key password is set to the same password as that used for the keystore.

  This parameter must be at least 6 characters long.

- `-keyalg`: specifies the algorithm to use when creating the key.

  RSA is typically used.

- `-keystore`: specifies the name and location of a keystore.

  If no keystore is provided on the command line, the file named `.keystore` in the home directory of user will be assigned.

- `-storepass`: specifies a password for the new keystore.

  This password must be configured as `jskpwd` within the `Global.xml` file.

After the keystore has been created, you must generate a certificate request. You can issue the following Java command to generate a certificate request:

```
keytool -certreq {-alias alias} {-file certreq_file} [-keypass keypass]
    {-keystore keystore} [-storepass storepass]
```

- `-alias`: specifies the alias that you defined for this certificate request.

  If you do not specify this parameter, a default value of mykey will be assigned.

- `-file`: specifies the output file for this command.

  This parameter is the CSR file that you will have to provide to your CA.

- `-keypass`: specifies a password used to protect the private key of the generated key pair.

  This parameter must match what you defined as the keypass when you generated the key pair.

- `-keystore`: specifies the name and location of a keystore.

- `-storepass`: specifies a password to a keystore.

At this point, you have created a certificate request file. This file must be sent to the certificate authority or the department responsible for creating certificates. When the certificate authority completes processing the certificate request, they return a certificate file in Base64 format. Then this certificate must be imported into the Java keystore as shown in the next step.

Now that the certificate has been created, you must import the certificate into the keystore. To do this, you have to have the client certificate in Base64 format. Then you can issue the following Java command to add this certificate to the trusted certificate authority file:

```
keytool -import
        -keystore c:\home\mftuser\keystore.jsk
        -alias MFT Command CenterClientKey
        -file cert.file
        -storepass changeit
```

Note: This command should be typed as a single line.

- `-keystore`: specifies the name and location of a keystore.

  You should point to the Java keystore. This file name should be added to the `jskname` parameter in the `Global.xml` file.

- `-alias`: specifies the unique name for this certificate.

  The value defined should match the alias defined in the `certreq` command.

- `-file`: specifies the certificate file name in Base64 format.

- `-storepass`: specifies the password for the cacerts keystore.

  This password must be configured as `jskpwd` within the `Global.xml` file. The default password is changeit.

After entering the command, you will be asked to confirm the request. After confirming the request, the certificate will be added to the Java keystore. Now, when your client makes a request to the server, the certificate can be passed to the web server.

The SSH Java Certificate Keystore

When installed, a default SSH keystore is installed. The SFTP transfers will work using this default keystore, or the user can create another keystore.

There are two types of keystores that can be used:

- DSA keystore uses the DSA key algorithm to create the public/private key pair.
- RSA keystore uses the RSA key algorithm to create the public/private key pair.

Note:

- The default SSH keystore uses the DSA key algorithm.
- DSA is required for SSH operation and that virtually all SSH clients and servers support the DSA key algorithm.
- Some SSH client or server software does not support the RSA algorithm.
- If keystores for both DSA and RSA are defined, then the SSH client and server will negotiate to define which SSH key will be used.

The Java keytool utility can be used to create the SSH certificate. Below is the format of the keytool command. When you have created the SSH certificate, you must update the **Management > SSH Server > Configure SSH Server** web page with the following information:

- DSA Keystore: specifies the DSA keystore file defined by the `keystore` parameter.

- DSA Keystore Password: specifies the DSA keystore password defined by the `storepass` parameter.
- Confirm Password: specifies the confirm password which should be the same as the DSA keystore password.
- DSA Private Key Alias: specifies the DSA alias name created by the `alias` parameter.
- RSA Keystore: specifies the RSA keystore file defined by the `keystore` parameter.
- RSA Keystore Password: specifies the RSA keystore password defined by the `storepass` parameter.
- Confirm Password: specifies the confirm password which should be the same as the RSA keystore password.
- RSA Private Key Alias: specifies the RSA alias name created by the `alias` parameter.

```
keytool -genkey {-alias alias} [-dname dname] [-keypass keypass]
    {-keystore keystore} [-storepass storepass] [-keyalg dsa]
```

Note: This command should be typed as a single line.

- `-alias`: specifies the unique name for this certificate chain and the private key in this new keystore entry.

  If you do not specify this parameter, a default value of mykey will be assigned.
- `-dname`: specifies the X.500 distinguished name to be associated with alias, and is used as the issuer and subject fields in the self-signed certificate.

  You should set the common name (`CN=`) to the host or IP name of client. This name will be used to access the server.

  If no distinguished name is provided at the command line, the user will be prompted for one.
- `-keypass`: specifies the password used to protect the private key of the generated key pair.

  This parameter must be the same as the `storepass` parameter defined. If no password is provided, the user is prompted for it. If you press ENTER at the prompt, the key password is set to the same password as that used for the keystore.
- `-keyalg`: specifies the algorithm to use when creating the key.

The valid values are DSA or RSA. DSA is typically used with SSH, because all SSH clients support DSA, but only part of them support RSA.

- `-keystore`: specifies the name and location of a keystore.

  If no keystore is provided on the command line, the `.keystore` file in the home directory of user will be assigned.

- `-storepass`: specifies a password for the new keystore.

  You can configure this parameter in the Configure SSH Server page. This password must be the same as the `keypass` parameter.

Example:

```
keytool -genkey -alias CFCCSSH -dname "CN=yourmachine, O=yourcompany,
OU=yourorganization, L=yourcity, ST=yourstage, C=yourcountry" -keypass
changeit
-keystore "c:\cfccinstall\keystore\keystore.dss" -storepass changeit
-keyalg DSA -keySize 1024 -validity 3650
```

Note: This command should be typed as a single line.

Environment Settings

You can run a batch file to set up classpath for the program in both Windows and UNIX operating systems.

Run the following batch file to set up a classpath for the program.

- For Windows: `setutilcp.bat`
- For UNIX k-shell: `setutilcp.sh`

The `setutilcp` file must be run each time you open a new command shell.

If you do not configure the environment settings, you have to specify all necessary `.jar` files in the classpath when running the Java program.

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

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

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for TIBCO® Managed File Transfer Internet Server is available on the TIBCO® Managed File Transfer Internet Server Product Documentation page.

- TIBCO® Managed File Transfer Internet Server *Managed File Transfer Overview*
- TIBCO® Managed File Transfer Internet Server *Installation*
- TIBCO® Managed File Transfer Internet Server *Quick Start Guide*
- TIBCO® Managed File Transfer Internet Server *User Guide*
- TIBCO® Managed File Transfer Internet Server *Utilities Guide*
- TIBCO® Managed File Transfer Internet Server *API Guide*
- TIBCO® Managed File Transfer Internet Server *Transfer and File Share Clients User Guide*
- TIBCO® Managed File Transfer Internet Server *Desktop Client User Guide*
- TIBCO® Managed File Transfer Internet Server *Security Guide*
- TIBCO® Managed File Transfer Internet Server *Container Deployment*
- TIBCO® Managed File Transfer Internet Server *Release Notes*

How to Contact TIBCO Support

Get an overview of TIBCO Support. You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to TIBCO Support website. If you do not have a user name, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

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

Legal and Third-Party Notices

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

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

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SOFTWARE GROUP, 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.

This and other products of Cloud Software Group, Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (https://www.tibco.com/patents) for details.