TIBCO® Managed File Transfer Command Center

Installation

Version 8.7.0 | October 2025

Contents

Installation Requirements

Before installing TIBCO® Managed File Transfer Command Center, ensure that your system meets the hardware, and software requirements and you have appropriate privileges.

Activation

MFT Command Center requires activation using a license to start. Licenses can be generated from the TIBCO Software Downloads site at https://www.tibco.com/downloads. For complete details on activating TIBCO products, see the TIBCO Activation Service documentation at https://docs.tibco.com/products/tibco-activation-service.

Important: MFT Command Center shuts down when your product entitlement ends and does not restart until you replace the license file with one that has a new entitlement end date. Replace your license file before the entitlement end date to avoid business disruption.

There are two options for activating MFT Command Center:

- **In-product Activation** (local activation): Copy your license file to the host system where you are running MFT Command Center.
- **TIBCO Activation Service** (remote activation): Install TIBCO Activation Service and upload your license file to the running instance before installing MFT Command Center.

To connect MFT to the activation server, you need the connection URL. This is available in the README.txt file in the /usr/libexec/tiblm/ directory on the activation server. The installation process prompts you to enter this URL. See the TIBCO Activation Service documentation for installation instructions, and to determine the URL for your TIBCO Activation Service instance.

Note: Licenses generated for **TIBCO Activation Service** and **in-product Activation** are not compatible. For instructions on generating a license for your selected activation option, see the TIBCO Activation Service documentation.

Installation Account

To install TIBCO MFT Command Center, you must have appropriate privileges.

Platform	Account Privileges
Microsoft Windows	No special privileges are required if you do not install TIBCO MFT Command Center as a Windows service. You must be an administrator to install TIBCO MFT Command Center as a Windows service. **Note:** You must be an administrator to install MFT in Windows 10 or Windows 11.
UNIX	When installing TIBCO MFT Command Center on a UNIX platform, it is good practice to install TIBCO MFT Command Center under a non-root user. **Note:** If you need TIBCO MFT Command Center to listen on ports below 1025, use the `iptables` command to redirect requests from these ports to valid TIBCO MFT Command Center ports that are above 1024. If you are using FTP or FTPS for incoming connections and use the `iptables` command to redirect the FTP or FTPS ports, transfers may fail due to FTP RFC port requirements of the data connection. We recommend using SFTP instead of FTP or FTPS. Optionally, you can use non-standard FTP or FTPS ports such as 2021 and 2990.

System Requirements

Before installing TIBCO MFT Command Center, ensure that your system meets the hardware and software requirements.

For more information about the hardware and software requirements and the supported platforms, see the product `Readme.txt` file.

Network

In a production environment, you might have to make changes to the firewall and other security systems.

The following table lists default ports for the services required and used within TIBCO MFT Command Center:

Supported Database	Default Port
MS SQL Server	1433
Oracle	1521\1522 SSL: 2484
MySQL	3306
PostgreSQL DB	5432

Note: Check with the appropriate systems administrator to ensure that these default ports are used in your enterprise application.

By default, TIBCO MFT Command Center uses port 8443 HTTPS and port 8080 HTTP. These default values can be changed during the installation process.

On UNIX, TIBCO MFT Command Center must be installed under a root user if you configure TIBCO MFT Command Center to use ports below 1025. However, you can use the `iptables` command to redirect ports 443 and 80 to valid TIBCO MFT Command Center ports that do not require root access. For example, if you define port 8443 for HTTPS and port 8080 for HTTP, you can use the following `iptables` commands to redirect port 443 to 8443 and port 80 to 8080:

```
iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 443 -j REDIRECT --
to-port 8443
iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j REDIRECT --
to-port 8080
```

Note: When using the `iptables` command to redirect requests from ports below 1025, the `iptables` command must be run from a root user.

Database Guidelines

TIBCO MFT Command Center provides a utility that can create and populate the required database tables. However, before starting the installation, you must create a TIBCO MFT Command Center database in the database application you use. We suggest creating DB tables using a UTF-8 character set and a case-insensitive collation. For Oracle, you might need to create a log in trigger to make searches case insensitive. When running under PostgreSQL, the CITEXT extension must be installed. MFT uses the POSTGRES CITEXT data type to provide searches that are case insensitive. Additionally, for case insensitive searches, you should use the PostgreSQL JDBC driver at level 42-2.12 or higher.

A PostgreSQL superuser must run the following command to add the extension to the database that MFT uses:

```
CREATE EXTENSION IF NOT EXISTS citext;
```

or the following command when specifying a specific schema:

```
CREATE EXTENSION IF NOT EXISTS citext SCHEMA my_schema;
```

Note: For more information about the supported databases, see Installation Requirements.

Ensure that your database administrator creates a database and a username and password on the server that hosts the TIBCO MFT Command Center database tables. This user must have the ability to read, write, and create tables in the TIBCO MFT Command Center database. The exact steps to create a database vary significantly depending on the database application you use. See the documentation provided by your database vendor for information about creating a database.

Follow these guidelines when you create a database:

- Database password must not contain an equal sign (=).

- If you use an MSSQL server database, you can configure the MSSQL server properties on the **Security** tab to perform authentication via SQL Server or Windows. (Default: Windows)
- If you use an Oracle 10i or later, use Cost Based Optimization (CBO) to tune the optimization for the first_rows of the TIBCO MFT Command Center database. To do this, enter the following command from `SQL*Plus` as SYSDBA after creating the database:

  ```
  alter system set optimizer_mode=first_rows_100;
  ```

- When using Oracle in FIPS mode, the database password must be at least 14 characters.

Warning: Do not to create database triggers on tables in the MFT database. Creating database triggers can cause unintended consequences and impede the support process. If you need to create a trigger, notify TIBCO Support.

Using a PostgreSQL Database

When installing a PostgreSQL database, MFT requires support for the CITEXT data type. To enable the CITEXT data type, perform the following procedure:

Procedure

1. Install the PostgreSQL CITEXT extension.
2. Run the following SQL command as a superadmin for the PostgreSQL MFT database:

   ```
   CREATE EXTENSION IF NOT EXISTS CITEXT;
   ```

Result

When PostgreSQL is selected as the database, MFT verifies that the CITEXT extension is installed for the MFT database. If it is not installed for the MFT database, the installation terminates.

Configuring Java On Windows or UNIX

TIBCO MFT Command Center supports Java 17 and above. We recommend that you install the newest version of the Java JDK. IBM Semeru Java, OpenJDK, Amazon Corretto, and Oracle Java are supported. GNU Java is not supported.

TIBCO MFT Command Center installation and configuration requires the `bin` directory of the JDK to be in your *PATH* environment variable. Instructions on how to do this are as follows.

Note:

- If you want to run the application server as a Windows service, you must set the *JAVA_HOME* environment variable for your system. For more information, see Appendix C. Starting on Startup Automatically.
- The Java version used to execute TIBCO MFT Command Center must be at the installation version or higher. Likewise, after executing TIBCO MFT Command Center do not downgrade Java to a prior version.

Before you begin

You must install a Java Software Development Kit (JDK) before you install MFTIS.

Procedure

1. Set the *JAVA_HOME* environment variable to point to the `Java\jdk` directory.

   For example,

   - On Windows:

     set `JAVA_HOME=C:\Program Files\Java\jdk-21`

   - On UNIX:

     export `JAVA_HOME=/opt/java/jdk-21`

2. Set the *PATH* variable to point to the `Java\bin` directory:

   For example,

   - On Windows:

     set `PATH=%JAVA_HOME%\bin:%PATH%`

or

set `PATH=C:\Program Files\Java\jdk-21\bin;%PATH%`

- On UNIX:

  export `PATH=$JAVA_HOME/bin:$PATH`

  or

  export `PATH=/opt/java/jdk-21/bin:$PATH`

3. Verify that the path is correctly set by using the following command:

   Windows and UNIX: `java -version`

FIPS Support

FIPS 140-2 is a standard defined by the National Institute of Standards and Technology (NIST) to set minimum standards for cryptographic processing. It provides a certification process to ensure that cryptographic routines conform to the standards and to use only the FIPS supported algorithms.

MFT does not provide its own cryptographic modules. MFT supports FIPS 140-2 by using the BouncyCastle FIPS 140-2 certified security provider and enabling BouncyCastle FIPS mode.

BouncyCastle FIPS 140-2

BouncyCastle FIPS 140-2 supports both IBM Semeru Java, OpenJDK, Amazon Corretto, and Oracle Java. The BouncyCastle FIPS files are distributed with MFT.

Follow the instructions defined in Configuring BouncyCastle FIPS 140-2 to enable or disable BouncyCastle FIPS mode.

Below are some of the restrictions when using BouncyCastle FIPS:

- System key sizes less than 2048 bits are not supported.
- When generating system keys, only 2048-bit and 3072-bit keys are supported in FIPS mode.
- PKCS12 is not supported in FIPS mode. JKS and BCFKS are supported, but the

Tomcat keystore must be in BCFKS format.

- The following PGP protocols are not supported:
  - El Gamal, CAST5, MD2, MD5, RipeMD
- Many SSL Ciphers are not supported when running in BouncyCastle FIPS mode. When running in BouncyCastle FIPS 140 mode, the list of SSL ciphers that are supported is displayed in the `catalina.out` file.

Note: The MFT fips script updates the `java.security` file, therefore it is important that MFT is the only application that uses this Java installation. If necessary, we suggest installing a Java version that only MFT uses. To run TIBCO MFT Command Center in different FIPS modes, separate JAVA is required for each one.

Installing TIBCO Managed File Transfer (MFT) Command Center in Console Mode

To install TIBCO MFT Command Center in console mode, perform the following operations:

1. Starting Automated Installation
2. Installing TIBCO MFT Command Center

Note: TIBCO MFT Internet Server and TIBCO MFT Command Center version 8.6.0 and above include changes to the way that passwords are encrypted and decrypted. These changes take effect only after all TIBCO MFT Internet Server and TIBCO MFT Command Center servers have been upgraded to version 8.5.0 or above. If all servers are at 8.5.0 or above version, the passwords are encrypted using the new password algorithms. If you change a password and revert one or more servers to version 8.4.0 or below, the old versions will not be able to decrypt passwords that are encrypted with the new password encryption algorithms. We strongly suggest backing up the database prior to upgrading all servers to version 8.5.0 or above.

Automated Installation Pre-requisites

To start the automated installation process of TIBCO MFT Command Center, you must perform certain steps.

Note: If you are installing TIBCO MFT on a machine that already has TIBCO Managed File Transfer Command Center installed, we strongly recommend installing in a different directory.

Before you begin

- You must install a Java JDK. For more information, see Configuring Java on Windows or UNIX.
- Extract the MFT Command Center Distribution compressed file into a directory. On Windows, we do not suggest installing TIBCO MFT Command Center in the "Program Files" or "Program Files (x86)" directories. We suggest creating a directory and extracting the files into that directory. Example: `c:\MFTCC`
- Copy the JDBC driver jar file(s) into the `database-lib` directory, which is present in the Installation directory and ensure the following:
  - Copy the proper JDBC driver to the `database-lib` directory.
  - Only one JDBC driver is in the `database-lib` directory.
  - The JDBC driver supports the installed Java version.
- If you runTIBCO MFT Command Center on a UNIX environment, ensure that the `install.sh` script has the execute attribute.
- If you are installing TIBCO MFT Command Center on one of the supported UNIX platforms, you should set the default permissions as follows:

```
-r-- r-- r-- cfcc.jar
-rw- r-- r-- CFIConfig.xml
drwx r-x r-x cloud
-r-- r-- r-- CMSInstall.jar
-r-- r-- r-- connmgr.jar
drwx r-x r-x database-lib
drwx r-x r-x distribution
-r-- r-- r-- EULA.txt
-rw- r-- r-- expressinstall.bat
-rwx r-- r-- expressinstall.sh
-rw- r-- r-- express.jar
-rw- r-- r-- fips.bat
-rwx r-- r-- fips.sh
-rw- r-- r-- install.bat
-r-- r-- r-- install-config.xml
-r-- r-- r-- installer.jar
-r-x r-x r-x install.sh
drwx r-x r-x keystore
-rw- r-- r-- log4j-1.2-api-2.23.1.jar
-rw- r-- r-- log4j2.xml
-rw- r-- r-- log4j-api-2.23.1.jar
-rw- r-- r-- log4j-core-2.23.1.jar
```

```
-rw- r-- r-- log4j-slf4j-impl-2.23.1.jar
-r-- r-- r-- server.jar
-rw- r-- r-- silent-setup.bat
-rwx r-- r-- silent-setup.sh
-rw- r-- r-- tibaclfix.exe
```

Procedure

1. Enter the following command on the command line to start the automated installation:
   - On Windows: `install`
   - On UNIX: `./install.sh`

```
                                MFT Installer Release 8.6.0
                                (supports all 8.6 versions)
Please note that this install will perform multiple App Server
restarts.
For this install, press the ENTER key to accept defaults and
continue.
You must read the license agreement before proceeding with the
installation.
Press enter to display the agreement.
```

2. Press **Enter** to display the End User License Agreement (EULA), and type yes to accept the license agreement.

   You can type `s` to skip to the end of the agreement.

3. Press **Enter** to continue.

   If you have added the *JAVA_HOME* variable and set the *PATH* variable as instructed in Configuring Java on Windows or UNIX, the product detects the version at this point. It is required that the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files must be installed before the TIBCO MFT Command Center installation if using an Oracle Java version prior to 8.u161. If they are not installed, the message shown as follows is displayed and the installation stops.

```
In order to use 256-bit secure keys, you must download the JCE
Unlimited Strength
Jurisdiction Policy Files from
http://www.oracle.com/technetwork/java/index.htm
l. After downloading, place the files into C:\Program
```

```
Files\Java\jdk-21\jre
\lib\security.
Installation failed! Details are in the install.log file.
```

Installing TIBCO MFT Command Center

After starting the automated installation, you must complete all installation steps to complete the installation.

Note:

- When installing TIBCO MFT Command Center on a Windows system, a Java window labeled MFT server is displayed during installation. This window must be kept open for the MFT server to continue running. Closing this window shuts down the web application.
- If you are installing TIBCO MFT Command Center on a machine that already has a version of TIBCO MFT Internet Server or TIBCO MFT Command Center installed, you must install TIBCO MFT Command Center in a new empty directory and not over an existing old MFT installation. This does not apply when installing Service Packs as hotfixes.
- If you are installing TIBCO MFT Command Center on a machine that has a version of TIBCO MFT Command Center executing, you must stop the current server before starting the installation. If the current server uses different TCP Ports, then you do not need to shutdown the current server.

Before you begin

The JDBC jar files must be copied to the `database-lib` directory that can be found in the installation directory. If this step is not done, copy the JDBC jar files into the `database-lib` directory and restart the installation.

Procedure

1. This step extracts the distribution file named `cfcc.jar`.

   In this step, the application server is installed and configured, and the *JAVA_HOME*

environment variable is detected.

If you install TIBCO MFT Command Center on a Windows system, you are prompted to define whether to run the application server as a Windows service.

Note: If you choose to install TIBCO MFT Command Center as a Windows service, and a TIBCO MFT Command Center Windows service already exists, the existing TIBCO MFT Command Center Windows service is stopped and a new Windows service is installed and started.

```
Is the application server installed as a Windows service
or do you want to run it as a Windows service? y/n/? [y]:
Stopping service MFT Command Center...................
```

```
Found distribution file c:\MFTCC\cfcc.jar
Use C:\MFTCC\cfcc.jar as the distribution? y/n [y]:
Extracting distribution file: C:\MFTCC\cfcc.jar

.............................................................
............
Distribution extracted successfully!

Is the application server installed as a Windows service or do you
want to run it as a Windows service? y/n/? [y]:

Installing application server to C:\MFTCC\server

.............................................................
.........................

Using C:\MFTCC\server as path to the application server
installation.
C:\MFTCC\server\conf\Catalina\localhost
```

2. This step sets up the connection to the TIBCO MFT Command Center database that you created.

   In this example, Oracle is used as the database server.

Note: If you use a MySQL database, you may get the following exception when connecting to MySQL 8.

Failed to establish a connection to the database: com.mysql.jdbc.exceptions.jdbc4.MySQLNonTransientConntionException: Public Key Retrieval is not allowed-error-in-java/

There are 3 possible workarounds for this issue (ranked most secure to least secure):

a. Use a TLS connection from TIBCO MFT Command Center to MySQL.

b. Change the `default_authentication_plugin` section to `mysql_native_password`.

c. Add the following parameter to the database connection URL:

`&allowPublicKeyRetrieval=true`.

Note that `allowPublicKeyRetrieval=true` could allow a malicious proxy to perform a man in the middle attack to get the plaintext password, that is why it is false by default.

```
Stopping the application server.............................
[OK]
A previous installation of the application server was found which
is
incompatible with
this version.
The current server will be upgraded. Please back up the current
server directory
before proc
eeding.
Do you wish to continue? y/n [y]
Installing application server to
/home/rkundu/MFT810/MFTCC810/server

.............................................................
.............
.......... ...
Using c:\MFTCC as path to the application server
installation.
c:\MFTCC/server/conf/Catalina/localhost
```

```
************************************************************
Step 2 Verifying database connection
Select database server type:
Enter 1 for MSSQL
Enter 2 for MySQL Enterprise Server or Community Server
Enter 3 for Oracle
Enter 4 for PostgreSQL
: 3
Oracle selected as database server type.
```

3. This step installs the database and populates the database tables. You must enter the necessary database connectivity and authentication information when prompted.

```
Step 3  Configuring the database
Executing database creation utility....
cmd /E:1900 /c setupdb.bat
"amRiYzpvcmFjbGU6dGhpbjpAMTkyLjE2OC43OC44OToxNTIxOm1mdDhz" oracle
bWZ0cWE? ******** oracle.jdbc.driver.OracleDriver BASE64
Allocating DBSetup object...
Determining database version....
Installing database...
Updating database...
Updating tables...
…
…
Updating records...
Done updating database.
Successfully installed database:
jdbc:oracle:thin://oracleserver:1521:cfcc
Successfully populated DB tables with default information.
```

If you already have a TIBCO MFT Command Center database, you are prompted to back up the database that you are using, because the database is updated during the TIBCO MFT Command Center installation process.

```
Step 3  Configuring the database
Database will be modified for new features. Please backup database
before proceeding.
Do you wish to continue? y/n [y]
```

4. This step configures TIBCO MFT Command Center for SSL communication.

   If you do not have a certificate, a self-signed certificate is created during the

installation process. You can either use a certificate issued by a certificate authority (CA) or use a self-signed certificate. During the installation process, you can choose the signature algorithm that is used to sign the self-signed certificate; the highest strength is SHA512 with RSA and the lowest strength is SHA256.

Note:

- Self-signed certificates are only practical for testing purposes and allow you to install and run the server quickly. It is recommended that you use a certificate created by a trusted Certificate Authority.
- Assigning port numbers below 1024 (so-called low-numbered ports) can only be bound to by root on UNIX systems.

```
Step 4  Evaluating the application server installation for HTTPS
connectors
Reading the application server configuration file:
C:\MFTCC\server\conf\server
xml
Found no pre-existing HTTPS connectors!
Do you have a pre-existing Java Keystore to be used as a server key
for SSL co
munication? y/n/? [n]:

Creating keystore for SSL communication
Enter the keystore path and filename..
[C:\MFTCC\keystore\keystore.jks]:
Directory C:\MFTCC\keystore does not exist! Create? y/n [y]:
Enter the keystore password (at least 6 characters)..[changeit]:
 Enter the alias of your private key.....................[cfcc]:
Enter the DNS Name or IP Address of your
server............:10.97.142.191
Select the signature and key algorithms you wish to use........:
1. SHA256 with RSA
2. SHA384 with RSA
3. SHA512 with RSA
Please enter your selection. [1]:
Enter your Company Name.............................
[Optional]:TIBCO
Enter your Organizational Unit Name.................[Optional]:Web
Debt
Enter the City where your company is located.........
```

```
[Optional]:Palo Alto
Enter the State where your company is located.....[Optional]:CA
Enter the two-letter country code for this unit.....[Optional]:US

Keystore filename    : C:\MFTCC\keystore\keystore.jks
Keystore password    : ********
Key alias            : cfcc
Server address       : 10.97.142.191
Signature and key alg: SHA256withRSA
Organization         : TIBCO
Organizational Unit  : Web Debt
Locality             : Palo Alto
State                : CA
Country              : US
Create a keystore with the above information? y/n [y]:

Creating keystore......
C:\Program Files\Java\jdk-21\bin\keytool -genkey -keystore
C:\MFTCC\keys
ore\keystore.jks -storepass ******** -keypass ******** -keyalg RSA
-sigalg SHA2
56withRSA -alias cfcc -keySize 2048 -validity 3650 -dname
CN=10.97.142.191, O=TI
CO, OU=Web Dept, L=Palo Alto, ST=CA, C=US

Enter the HTTPS Port to listen for connections… [8443]:
```

5. This step configures the TIBCO MFT Command Center components and ports on the application server.

   In this example, to provide the most secure environment, the connector is set to only allow secure ciphers by default. To view those ciphers, type `V` for them to be displayed. If you want the server to support all ciphers, you can select option 2.

```
Step 5  Updating the application server Connector Configuration

Default HTTPS Connector parameters for port 8443:
The Default Verbosity Level                          - 2
The Default Debug Level                              - 2
The Default Buffer size                              - 2048
The Default Connection Timeout                       - 60000
The Default DNS Lookup set to                        - true
The Default Max active requests                      - 128
```

```
The Default Min Processors                              - 5
The Default Max Processors                              - 100

Accept these parameters? y/n [y]:
Select the SSL ciphers you wish to the server to support.
1. Most Secure ciphers (excludes CBC ciphers)
2. All Secure ciphers (including CBC ciphers)
3. All ciphers
Please enter your selection or v to view secure ciphers. [1]:
Enter the HTTP port to listen for connections... [8080] :

Enter the port to listen for shutdown requests... [8005] :
```

6. This step configures the context root that will be used in the URL.

   The context name must be set to an alphanumeric name. Using special characters within a context name can cause unpredictable results. Using the default value for cfcc is recommended.

```
Step 6  Evaluating the application server installation for contexts
Enter the context root for this installation ........[cfcc]

Reading context configuration file:
C:\MFTCC\server\conf\Catalina\localhost\cfcc.xml
Found no pre-existing Contexts
```

   Note: If you are upgrading, you will be prompted to back up your present settings because only one instance of `cfcc` can exist on the server.

7. This step extracts the `cfcc.war` file to install the TIBCO MFT Command Center application.

```
Step 7  Installing web application
Use C:\MFTCC\server\webapps\cfcc as the installation directory?
y/n/? [y]:

Extracting distribution\cfcc.war to C:\MFTCC\server\webapps\cfcc
```

8. This step verifies the context configuration for TIBCO MFT Command Center.

```
Step 8  Updating the application server context configuration

Default Context parameters:
The Default Log File Prefix                          - localhost_
cfcc_
The Default Log File Suffix                          - .txt
The Default Log File Timestamp                       - true
The Default Log File Verbosity Level                 - 2
The Default Log File Debug Level                     - 0

Add a new context with the above parameters? y/n/? [y]:
```

9. This step updates the TIBCO MFT Command Center `web.xml` file and installs the TIBCO MFT Command Center Administrator service.

```
Step 9 Configuring web.xml

Enter the name of the host on which the application will run.
[SystemA]:


Enter a directory to store log files......[c:\MFTCC\logs]:

Configure web.xml with the above parameters? y/n [y]:
Starting the application server.................. [OK]
```

```
Administrator service is used to manage the application.
You should only install this service inside your internal network.
Install this service? y/n? [n]:

Enter a directory to store log files......[c:\MFTCC\logs]:

Configure web.xml with the above parameters? y/n [y]:
Starting the application server.................. [OK]
```

10. This step explains how to activate TIBCO MFT Command Center.

    This product requires an activation license. The URL for your TIBCO Activation Service instance is stored in the `/usr/libexec/tib-lm/README.txt` file on the system or virtual machine where your instance is running.

    You can validate the license using either of two ways:

- **Activation Server**: Enter the connection URL from the `/usr/libexec/tib-lm/README.txt` file on the activation server.
- **Local License File**: Enter the full path to the directory containing a valid local license file.

The MFT server does not start without a valid license.

11. This step explains how the JMS jar files should be copied after the installation is complete.

    If you are using the JMS interface, you must copy the JMS jar files to the *`MFTCC_Install`*`\server\webapps\cfcc\WEB-INF\lib` directory.

```
Step 10 Copy JMS files

If you are using the JMS interface, you must copy the JMS jar files
to the
following location:
C:\MFTCC\server\webapps\cfcc\WEB-INF\lib
These jar files are typically found in the JMS Server installation.
Restart the MFT server after copying the jar files.
You can configure and test the JMS settings through the Command
Center.
Go to the Management > Command Center Services > JMS Service
> Configure JMS Service page.
On that page you can click on help for a list of the provider
specific jar files.

Press the enter key to continue.

Installation completed!  Details are in the install.log file.
```

What to do next

After TIBCO MFT Command Center is installed, you can access the TIBCO MFT Command Center Administrator web pages using one of the following URLs:

- https://[*DNS_HostName*]:*[httpsPort*]/[*context*]/control?view=view/admin/start.jsp
- https://[*DNS_HostName*]:*[httpsPort*]/admin

Note: If the default context is not used during the installation process, the redirection file for this shortcut and others mentioned later in this manual needs to be updated to redirect to the non-standard context. Follow the following instructions to make these changes:

The redirection files can be found in the *MFTCC_Install*`\server\webapps\ROOT` directory. Use a text editor to open and change the cfcc context in these files to the new context chosen during the installation process. After making the changes, save and close the files.

When you are prompted for a user ID and password, you must log in with the administrator credentials of `admin/ changeit`. You should change this password to a more secure password as soon as possible.

Installing TIBCO MFT Command Center in Silent Mode

You can install TIBCO MFT Command Center in silent mode by using the `SilentInstall.xml` file.

You must start the installation process using the proper user authorization. For more information, see Installation Requirements.

Procedure

1. Download and extract the installation package to an installation directory on your computer.
2. Copy the JDBC driver jar file(s) into the `database-lib` directory, which is present in the Installation directory.
    a. Make sure that you have copied the proper JDBC driver to the `database-lib` directory.
    b. Make sure that only one JDBC driver is in the `database-lib` directory.
    c. Make sure that the JDBC driver supports the installed Java version.
3. Execute the `silent-setup` program to create your `SilentInstall.xml` file. This program will request information about the installation environment, similar to the questions asked during the standard install. The output of this script will be the `SilentInstall.xml` file. If you want to run the installation in a different directory from the directory where the silent install was performed, you must copy these files and / or directories to the installation directory:
    - `SilentInstall.xml`
    - The keystore created in the silent install
    - The `database-lib` directory, which contains the JDBC JAR file(s)

    On a command line, navigate to the `MFTCC_Install` directory and start the silent mode configuration by executing the following command:
    - On Windows: `silent-setup`

- On UNIX: `./silent-setup.sh`

For more information about the format and parameters of the `SilentInstall.xml` file, see SilentInstall.xml File Parameters.

4. On a command line, navigate to the *`MFTCC_Install`* directory and start the installation in silent mode by executing the following command:
   - On Windows: `install.bat silent`
   - On UNIX: `./install.sh silent`

SilentInstall.xml File Parameters

All the parameters in the `SilentInstall.xml` file are required unless indicated otherwise. Do not update this file unless instructed to by TIBCO Support.

The following example shows a sample `SilentInstall.xml` file:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<silentinstall>
<!-- Database Settings -->
<arg name="db_type" value="mysql"/>
<arg name="db_host" value="localhost"/>
<arg name="db_port" value="3306"/>
<arg name="db_ssl" value=""/>
<arg name="db_ciphers" value=""/>
<arg name="db_name" value="mft860"/>
<arg name="db_oracle_type" value=""/>
<arg name="db_user" value="root"/>
<arg name="db_password" value="$$ENCODED:dCbWgIvpILRQgr5QivE1d8L7F2A="/>
<arg name="db_url" value=""/>
<arg name="db_schema" value="cfcc"/>
<arg name="db_tablespace" value=""/>
<arg name="db_classname" value="com.mysql.cj.jdbc.Driver"/>
<!-- MFT Server Port Settings -->
<arg name="httpsport" value="8443"/>
<arg name="httpport" value="8080"/>
<arg name="shutdownport" value="8005"/>
<!-- Keystore Settings -->
<arg name="keystore" value="keystore.jks"/>
<arg name="keystorepassword"
value="$$ENCODED:QjhpudCGsR+s7YD91UB7ZKI0UV8="/>
<arg name="keystorealias" value="cfcc"/>
```

```
<!-- Miscellaneous Settings -->
<arg name="server_ciphers" value="most_secure"/>
<arg name="admininstall" value="false"/>
<arg name="hostname" value=""/>
<arg name="context" value="cfcc"/>
<arg name="adminuser" value="admin"/>
<arg name="logs_directory" value="logs"/>
<arg name="adminpassword"
value="$$ENCODED:/8juoLVihSCqRDVNaq1moV7SD38="/>
<arg name="allow_root" value="false"/>
<arg name="win_service" value="false"/>
<arg name="tib_activation"
value="https://activation.example.com:7070?fp=1234567890abcdef1234567890
abcdefc263b10987654321fedcba0987654321"/>
</silentinstall>
```

Database Settings

Parameter	Description
`db_classname`	Defines the database driver Java class name.
`db_ciphers`	Defines the database ciphers to be used when using Oracle database connections in SSL mode.
`db_host`	Defines the IP name or IP address of the database server.
`db_name`	Defines the name of the database or schema.
`db_oracle_type`	Defines the Oracle database type. The valid values are `sid` and `service`.
`db_password`	Defines the password for the database user. Two formats for this password can be used: clear text password and base64-encoded encrypted password. The base64-encoded encrypted password is generated by an MFT program and cannot be set by editing the file.
`db_port`	Defines the port that the database is listening on.

Parameter	Description
db_ssl	Defines whether the database connections use SSL/TLS. The valid values are as follows: • true: uses TLS/SSL for database connections. • false: uses clear database connections.
db_type	Defines the type of the database you use. The valid values are: mysql, oracle,postgresql, and mssql.
db_url	Allows you to override the URL that MFT will normally generate. This parameter is optional. When used, it will cause the installer to ignore these parameters: db_host, db_port, and db_name.
db_user	Defines the user name that has access to the defined database.
db_schema	Defines the schema name of the database.

Server Port Settings

Parameter	Description
httpport	Defines the HTTP port number.
httpsport	Defines the HTTPS port number.
shutdownport	Defines the server shutdown port number.

Keystore Settings

Parameter	Description
keystore	Defines the name of the keystore file used by the HTTPS

Parameter	Description
	connector. **Note:** The keystore file must be located in the same directory as the other MFT installation files.
`keystorealias`	Defines the keystore key alias used by the HTTPS connector.
`keystorepassword`	Defines the encrypted keystore password. Two formats for this password can be used: clear text password and base64 encoded encrypted password. The base64-encoded encrypted password is generated by an MFT program and cannot be set by editing the file.

Miscellaneous Settings

Parameter	Description
`admininstall`	Defines whether the admin service will be installed for this MFT application.
`adminpassword`	Defines the encrypted password for the admin user.
`adminuser`	Defines the admin user that is used to connect to the MFT server to validate that it is operational.
`allow_root`	Defines whether the MFT application can be installed by a root user. The valid values are `true` and `false`.
`context`	Defines the context for the MFT server.
`hostname`	Defines the host name for the MFT application. This is an optional parameter. When not defined, the host name of the computer where TIBCO MFT Command Center is being installed is used.

Parameter	Description
	When defined, this parameter overrides the host name.
`logs_directory`	Defines the directory in which the MFT audit, message, trace, and webAdmin files are written.
`server_ciphers`	Defines the cipher strength that is supported. The default value is `all_secure` which includes the CBC ciphers. The valid values are `all_secure`, `most_secure`, and `all`.
`win_service`	Defines whether the MFT application should be run as a Windows service. The valid values are `true` and `false`.
`tib_activation`	Specifies the URL of the activation server or the path to a local license file. For a URL, use the connection URL found in the `/usr/libexec/tiblm/README.txt` file on the activation server. For a file path, provide the full path to the directory containing the license file. The directory must be on the same host as the MFT instance. To activate the MFT instance using a local file, set the value to the directory where the license file is stored as shown in the following example: `<arg name="tib_activation" value="C:\temp\mft_licenses"/>`

Installing TIBCO MFT Command Center using Express Installation

Express installation is meant to simplify the container installation, but it can be used with the standard on-premises installation. The MFT distribution file includes a pre-installed TIBCO MFT Command Center. After accepting the EULA, the pre-installed TIBCO MFT Command Center is installed in the server directory. You must set the environment variables to define the connectivity to the MFT database and to the HTTPS Keystore. You can use the `clouddbconfig` utility in the `cloud/dbconfig` directory to create the environment variables and encrypt the database password. You can use the `keystorepwd` utility in the distribution\util\dbsettings directory to update the keystore password in the server.xml. On startup, TIBCO MFT Command Center or TIBCO MFT Internet Server, checks the database for updates. Updates are applied if required, and the database version is set to the latest installed or hotfix version.

The distribution file includes a self-signed keystore: `./keystore/keystore.jks`

The Common Name (CN) for the private key is `localhost`

Restriction:

- Database drivers are not included. You must supply the JDBC driver.
- JMS library jar files are not included. If JMS is used, you must supply the JMS jar files driver.
- The context is hard-coded to `cfcc`. Express installation is not supported if you want to use a different context.
- To change the default http/https ports, you must manually change the following file:
  `<MFT-Install>/server/conf/server/xml`

Note: If installing on Windows 10 or Windows 11 operating systems, express installation is performed only when the user is running the installation with Administrator privileges.

The configured http/https ports depend on whether you have installed TIBCO MFT Internet Server or TIBCO MFT Command Center is installed:

Product	HTTP Port	HTTPS Port
Command Center	8080	8443
Internet Server	7080	7443

Procedure

1. Download and extract the TIBCO MFT Command Center installation package to an installation directory on your computer.
2. Install a current version of Java JDK. We suggest using Java 17 or Java 21, since these are LTS (Long Term Support) versions of Java.
3. Run the following scripts to setup the installed version of TIBCO MFT Command Center.

   UNIX: `./expressinstall.sh`

   Windows: `expressinstall.bat`

   > **Note:** Accept the EULA to continue the installation.

   When the `expressinstall` script completes, the preinstalled TIBCO MFT Command Center is installed in the `<MFT-Install>/server` directory.
4. Set the `JAVA_HOME` environment variable. We suggest adding this to the profile that starts TIBCO MFT Command Center.

   UNIX: `export JAVA_HOME=/java/jdk-21`

   Windows: `set JAVA_HOME=/java/jdk-21`
5. Copy the JDBC jar files to the `./server/lib` and `./cloud/dbconfig/lib` directories.
6. Copy a keystore to replace the default keystore. Note that the supplied `keystore.jks` is a self-signed keystore with the default java keystore password.

   If you are using a different keystore that is not using the default keystore password, run the following script to change the keystore password in the `server/conf/server/xml` file:

```
cd distribution/util/dbsettings ; ./keystorepwd.sh
```

Follow the instructions in the script. Note that you require the fully qualified path to the `server.xml` file. Typically this is: `<MFT-Install>/server/conf`.

Optionally, you can set the environment variable `COM_TIBCO_MFT_CE_KEYSTORE_PWD` to define the keystore password. For more information about setting environment variables, see Environment Variables.

7. If you are using JMS, copy the JMS jar files to the `./server/webapps/cfcc/WEB-INF/lib` directory.
8. Set the environment variables to point to the correct database. You must set the following environment variables:

```
export COM_TIBCO_MFT_CE_DB_TYPE
export COM_TIBCO_MFT_CE_DB_USER
export COM_TIBCO_MFT_CE_DB_PWD
export COM_TIBCO_MFT_CE_DB_URL
export COM_TIBCO_MFT_CE_DB_DRIVER
export COM_TIBCO_MFT_CE_KEYSTORE_PWD
```

To display the environment variables, use the following script:

```
cd cloud/dbconfig ; ./clouddbconfig.sh
```

- Follow the instructions to **Add Database Server settings**.
- Go to **Manage Database Server settings**.
- Select **DB Config** and select **Display Cloud Database Environment variables**.

  The environment variables are displayed.

Optionally, you can use the following script to display the encrypted password. You can then set the displayed password by the environment variable `COM_TIBCO_MFT_CE_DB_PWD`:

```
cd cloud/dbconfig ; ./clouddbconfig.sh encrypt abc123
```

9. Start TIBCO MFT Command Center.

   UNIX: `server/bin/startup.sh`

Windows: `cd server/bin & startup.bat`

Use the following command when TIBCO MFT Internet Server is installed as a service:

```
cd server/bin
service install
net start MFT_Command_Center
```

Note: Start the TIBCO MFT Command Center as an administrator when running on Windows 10 or Windows 11.

You can run the following sample UNIX script to perform steps 4 to 9 of the procedure. Change the bold italicized fields to match your environment.

Example

```
# Step 4: Export JAVA_HOME
export JAVA_HOME=/java/jdk-21

# Step 5: Copy the JDBC jar files
cp /data/jdbc/mysql/mysql-connector-java-8.0.27.jar ./server/lib


# Step 6: Copy the keystore (if you using a different keystore file)
cp /data/keystore/mykeystore.jks ./keystore/keystore.jks

# Step 7: Copy the JMS jar files (if JMS is required)
cp /data/JMS/* ./server/webapps/cfcc/WEB-INF/lib

# Step 8: export the database environment variables
export COM_TIBCO_MFT_CE_DB_TYPE=mysql
export COM_TIBCO_MFT_CE_DB_USER=cfcc
export COM_TIBCO_MFT_CE_DB_PWD=/X5Zwxbjh+HJMy8eTVJfaVQpia/N1o=
export COM_TIBCO_MFT_CE_DB_URL="
jdbc:mysql://your.db.server:3306/mftdb?characterEncoding=UTF8&useSSL=fal
se&serverTimezone=UTC"
export COM_TIBCO_MFT_CE_DB_DRIVER=com.mysql.cj.jdbc.Driver
export COM_TIBCO_MFT_CE_KEYSTORE_PWD=CLR:yourkeystorepwd
# Step 9: Start MFT Server
server/bin/startup.sh
```

Environment Variables

Enter the following environment variables to override the existing environment variables or to extend the capabilities of MFT:

Environment Variable	Description
COM_TIBCO_MFT_CE_DB_TYPE	Defines the type of database used. Valid values are `oracle, mysql, mssql, postgresql` Example: `export COM_TIBCO_MFT_CE_DB_TYPE=mysql`
COM_TIBCO_MFT_CE_DB_USER	Defines the database user when creating JDBC connections to the database. Example: `export COM_TIBCO_MFT_CE_DB_USER=mftdbuser` **Note:** This parameter overrides the value defined in the web.xml DBUser parameter. This environment variable only takes effect when all of the `COM_TIBCO_MFT_CE_DB` parameters are defined.
COM_TIBCO_MFT_CE_DB_PWD	Defines the database password used when creating JDBC connections to the database. Enter the password value in the following three ways: • Encrypted Password `export COM_TIBCO_MFT_CE_DB_ PWD=YFqc/bvasl/TNwDxypF8PHsnlp4=`

Environment Variable	Description
	`` export COM_TIBCO_MFT_CE_DB_ PWD=PWD:pz1clz83dAcdrJVBUYUu2P+vLyY= ``` **Note:** Use the `clouddbconfig.sh encrypt` utility to encrypt a password. This utility is located in `<MFT-Install>/cloud/dbconfig` • Base64 encoded password: ``` export COM_TIBCO_MFT_CE_DB_ PWD=B64:eW91cmRicGFzc3dvcmQ= ``` • Clear Text Password: ``` export COM_TIBCO_MFT_CE_DB_ PWD=CLR:yourdbpassword ``` **Note:** This parameter always overrides the value defined in the web.xml `DBPass` parameter. As a security mechanism the DB password can be extracted from a vault product without being defined in a disk file.
COM_TIBCO_MFT_CE_DB_URL	Defines the database URL used when creating JDBC connections to the database. Example: ``` export jdbc:mysql://localhost:3306/yourdb? characterEncoding=UTF8&useSSL= false&serverTimezone=UTC ```

Environment Variable	Description
	Note: This parameter overrides the value defined in the web.xml `DBConn` parameter. It only takes effect when all of the `COM_TIBCO_MFT_CE_DB` parameters are defined.
COM_TIBCO_MFT_CE_DB_DRIVER	Defines the JDBC Driver class name used when creating JDBC connections to the database. Example: `export COM_TIBCO_MFT_CE_DB_DRIVER=com.mysql.cj.jdbc.Driver` **Note:** This parameter overrides the value defined in the web.xml `DBDriver` parameter. It only takes effect when all of the `COM_TIBCO_MFT_CE_DB` parameters are defined.
COM_TIBCO_MFT_CE_KEYSTORE_PWD	Defines the keystore password used when creating the HTTPS connector. Enter the password value in the following three ways: • Encrypted Password: `export COM_TIBCO_MFT_CE_KEYSTORE_PWD=1Gwv9di9cneSnXSqQl0pOT5L+CI=` **Note:** Use the `clouddbconfig.sh encrypt` utility to encrypt a password. This utility is located in `<MFT-Install>/cloud/dbconfig` • Base64 encoded password: `export COM_TIBCO_MFT_CE_KEYSTORE_`

Environment Variable	Description
	`` PWD=B64:Y2hhbmdlaXQ= ``` • Clear Text Password: ``` export COM_TIBCO_MFT_CE_KEYSTORE_ PWD=CLR:changeit ``` This parameter always overrides the value defined in the server.xml `keystorePass` parameter. As a security mechanism the keystore password can be extracted from a vault product without being defined in a disk file.
COM_TIBCO_MFT_ENCRYPT_KEY	Defines the encryption key used to encrypt the created or updated passwords. The decryption routine can detect when a password is encrypted using the environment variable. The length of this environment variable is limited only by the length permitted by the operating system or shell. We suggest using uppercase and lowercase characters, numbers, and special characters for this parameter. **Note:** This does not apply to user passwords which are not encrypted in the database. They are stored as hashes. Example: ``` COM_TIBCO_MFT_ENCRYPT_ KEY=kowY61kTy..... ```

Environment Variable	Description
	Note: • This parameter overrides the encryption key used to encrypt passwords stored in the database. • Encryption of passwords using the encryption key defined by this environment variable only occurs when the following conditions are met: ◦ All Internet Server and Command Center instances have been upgraded to version 8.5.0 or higher. ◦ All Internet Server and Command Center instances have set this environment variable to the same value. For more information about this environment variable, see "Define Encryption Key Environment Variable" section in TIBCO MFT Command Center Security guide.
COM_TIBCO_MFT_HOSTTYPE	Defines whether MFT configuration records that are created when Internet Server or Command Center initializes, are for cloud instances or for permanent instances. The following values are valid for this parameter: • Cloud `export COM_TIBCO_MFT_HOSTTYPE=Cloud` The configuration records are for cloud instances and are deleted when a cloud instance stops and is inactive for 10 minutes. • Permanent

Environment Variable	Description
	`` export COM_TIBCO_MFT_ HOSTTYPE=Permanent ``` The configuration records are for permanent instances. If the config records are not defined, permanent config records are created, or else the existing config records are used.
JAVAMEM_MB_MIN	Defining this parameter overrides the minimum heap size `-Xms` defined in the **setenv.sh** and **setenv.bat** files. The default value is `-Xms512m`. Example: ``` export JAVAMEM_MB_MIN=1024m ```
JAVAMEM_MB_MAX	Defining this parameter overrides the maximum heap size `-Xmx` defined in the **setenv.sh** and **setenv.bat** files. The default value is `-Xmx4096m`. Example: ``` export JAVAMEM_MB_MAX=8192m ```
TIB_ACTIVATION	The URL of the activation server or the full path to the directory where the local license file is located. For URL value, enter the connection URL found in the `/usr/libexec/tib-lm/README.txt` file that is used on the activation server.

Installing Connection Manager Server

Connection Manager is an MFT feature that allows connections from the DMZ to be initiated through from the Internal Network to the DMZ. Connection Manager must be installed and configured when the DMZ firewall does not allow connection to the internal network to be initiated by applications running in the DMZ. There are two components of Connection Manager:

1. CMS (Connection Manager Server)

The CMS is installed in the internal network and accepts connection requests over a control connection and initiates the TCP connection to the external network.

2. CMA (Connection Manager Agent)

The CMA is installed in the DMZ, typically on the same machine where the Internet Server is run. CMA uses the SOCKS protocol to intercept connection requests from the Internet Server. It then sends a connection request to CMS over the control connection established by CMS, and waits for CMS to initiate a TCP connection back to CMA.

CMS (Connection Manager Server) must be installed in the internal network behind the firewall. CMS works with CMA (Connection Manager Agent), which is typically found in DMZ installations where firewall rules do not allow connections to be initiated from the DMZ to the internal network. The CMS installation is not supported by the Silent Installer.

Before you begin

- You must be the system administrator of the operating system to complete the CMS installation.
- A Java JDK must have been installed, if you have not already set the *JAVA_HOME* variable as required for the TIBCO MFT Command Center installation, set this variable along with the PATH statement. See Configuring Java on Windows or UNIX for more information.

Note: CMS installation is not part of the TIBCO MFT Command Center installation. CMA can either be installed as part of the Internet installation or it can be installed separately. See Appendix C: Connection Manager in the User Guide for more information.

Upgrading CMS

There are two ways to upgrade CMS.

Procedure

1. Install the updated CMS in the same directory where the CMS prior version was installed.
   CMS detects that CMS software needs to be upgraded. CMS upgrades the application server and CMS code as needed.
2. Install the updated CMS in a new installation directory.

   The CMS installer prompts you for an upgrade CMS from another installation. If you want to upgrade from another installation directory, you are prompted for the current CMS installation directory. CMS copies the configuration files from the current CMS directory and uses these for the new CMS installation.

   Note: When installing CMS in a new directory, make sure to change any automated startup scripts or services to point to the new installation directory.

CMS Installation Modes

You can install Connection Manager Server in one of the following ways:

- Performing a console installation and responding to prompts. For more information, see CMS Console Installation.
- Performing a silent installation. This requires a one-time setup. After that, you can install CMS using a single command without any console interaction. For more

information, see Installing CMS using Silent Installation.

CMS Console Installation

CMS is installed separately from TIBCO MFT Command Center. Although it can be installed in the same directory as Command Center, we suggest installing this in a separate directory.

Procedure

1. Copy the `CMSInstall.jar` file from the *`MFTCC_Install`* directory to the directory on the machine where you want to run CMS. If you are installing CMS on the same machine where a TIBCO MFT Command Center is running, we suggest copying these files to a directory at the same level as the TIBCO MFT Command Center install directory.
2. Extract the file using the `jar -xvf CMSInstall.jar` command.

   The following files are shown, if you install CMS on one of the supported UNIX systems. The default permissions are also displayed:

   ```
   cfccEncrypt.jar                -rw-r--r--
   cmsinstall.bat                -rw-r--r--
   CMSInstall.jar                -rw-r--r--
   cmsinstall.sh                 -rw-r--r--
   cmsserver.jar                 -rw-r--r--
   connmgr.jar                   -rw-r--r--
   installer.jar                 -rw-r--r--
   log4j-1.2-api-2.24.1.jar      -rw-r--r--
   log4j2.xml                    -rw-r--r--
   log4j-api-2.24.1.jar          -rw-r--r--
   log4j-core-2.24.1.jar         -rw-r--r--
   server.jar                    -rw-r--r--
   ```

3. If CMS is running on UNIX and the `cmsinstall.sh` file attributes do not contain the executable flag, run the following command:

   ```
   chmod 555 cmsinstall.sh
   ```

4. Run the following command to start the CMS automated installation:
   - On Windows: `cminstall.bat`

- On UNIX: `./cmsinstall.sh`

The following information is displayed. In this example, the default values are accepted.

```
MFT Connection Manager Server

The MFT Connection Manager consists of two components:
: MFT Connection Manager Agent(CMA):  Distributed with MFT Internet
Server
: MFT Connection Manager Server(CMS): Distributed with MFT Command
Center

This program will guide you through the CMS installation.
The CMS is deployed in the internal network and coordinates the
creation of
sessions to the CMA running in the DMZ.

Press <ENTER> to continue with the Connection Manager Server
installation.
```

Procedure

1. This step installs the CMS in the current running directory.

```
CMS Step 1  Install CMS Server

CMS will be installed in directory:
  C:\MFT730CMS\cmsserver
.....
```

2. This step defines the ports that CMS uses in the environment.

   By default, CMS uses HTTPS port 48443 and shutdown port 48005.

```
CMS Step 2 Configure CMS Server ports
CMS will use the following Server Ports:
: HTTPS Port.....: 48443
: Shutdown Port.....: 48005
Note: The default values will work in most environments.
To Accept these ports hit <Enter>. Otherwise type N and press
<ENTER>:[y]
Testing Server ports:
Testing https port 48443: Successful!
```

```
Testing shutdown port 48005: Successful!
Server Port tests successful. Press <Enter> to Continue
```

3. This step sets up the default ports to be used for the communication between CMS and CMA.

```
CMS Step 3 Configure Connection Manager Agent (CMA)
Now we will define the Connection Manager Agents (CMA).
Enter the CMA Host Name or IP Address.........[]:10.97.142.89
CMA will use the following Command and Data ports.
CMA Command IP Port...............: 48000
CMA Data IP Port..................: 48001
Note: The default values will work in most environments.
To Accept these ports ............:[y]
```

4. This step configures the password used by TIBCO MFT Command Center to configure the CMS.

   By default, the password is set to `changeit`. You can change the password later through TIBCO MFT Command Center if needed.

```
CMS Step 4 Configure the password used by Command Center to
configure CMS.
Command Center requires a password to configure CMS
Enter the password used by Command Center to configure CMS.....
[changeit]:
Default password:[changeit] will be used
Enter 'y' to confirm, Enter 'n' to re-enter password...[y]
```

5. This step starts the CMS application server.

   CMS attempts to connect to CMA every 30 seconds until a connection is successful.

```
CMS Step 5 Starting the CMS Server
The Connection Manager Server is starting
Testing Connection Manager Server connection:
Try 0 to contact to the application server
Try 1 to contact to the application server
Connection Manager test successful
CMS will connect to the CMA agent every 30 seconds until a
connection is
```

```
successful: (10.97.142.89:48000 )
```

6. This step displays some final instructions to help you log in and configure the CMS through TIBCO MFT Command Center.

```
CMS Step 6 CMS Installation Complete
Now you can install the Connection Manager Agent(CMA) on host
10.97.142.89
The CMA is distributed with the MFT Internet Server. CMA can be
installed and
configured during the MFT Internet Server installation in the DMZ.
During the
MFT Internet Server CMA installation you will be prompted for the
following
information:
: When prompted for "CMA Command IP Port" enter: 48000
: When prompted for "CMA Data IP Port" enter: 48001
Most of these configuration options can be changed through the
Command Center
admin pages. To configure this CMA through Command Center, do the
following:
: Management ==> Connection Manager ==> Add Connection Manager Node
: Set a unique name for this CM Node
: Set the IP Address to the IP Address or Host Name of this machine
: Set the IP Port to HTTPS Port 48443
: Set the Password in the Server credentials to the password you
just entered
Connection Manager Server installation completed successfully!
Installation details are in the cmsinstall.log file.
```

Note: For more information about configuring the connection, see *TIBCO Managed File Transfer Command Center User Guide*.

Installing CMS using Silent Installation

The silent installation process consists of the following steps:

1. Collecting the configuration information required for installation by using a one-time setup.
2. Using the configuration information to perform a silent installation.

Procedure

1. Copy the `CMSInstall.jar` file in the `MFTCC-Install` directory to the directory on the machine where you want to run the CMS setup.
2. Extract the file using the `jar –xvf CMSInstall.jar` command.

   The directory should have the following files:

   ```
   cfccEncrypt.jar
   cmsinstall.bat
   cmsinstall.sh
   cmsserver.jar
   connmgr.jar
   installer.jar
   log4j-1.2-api-2.24.1.jar
   log4j2.xml
   log4j-api-2.24.1.jar
   log4j-core-2.24.1.jar
   server.jar
   ```

3. If CMS is installed on UNIX, run the following command to set the executable flag:

   ```
   chmod 555 cmsinstall.sh
   ```

4. Run the following command to configure the silent information:

   For Windows: `cmsinstall.bat setup`

   For UNIX: `./cmsinstall.sh setup`

5. Users are prompted to enter the required information for CMS to run. The information needed is the same as the normal installation.
   - HTTPS port: CMS HTTPS port TIBCO MFT Command Centeruses to configure CMS.
   - Shutdown port: CMS shutdown port.
   - CMA IP address: CMA IP address to which this CMS connects.
   - CMA command port: CMS connects to this port to build a command channel.
   - CMA data port: CMS connects to this port for a data channel.
   - Host IP address or subnet that can manage this CMS.

- Password that TIBCO MFT Command Center uses to manage this CMS.

6. If you are starting CMS in a different directory, copy all the files in this directory (.jar, .bat, .sh, .xml, .properties) to the destination directory where CMS is installed.
7. Run the following command to start the silent installation:

   For Windows: `cmsinstall.bat silent`

   For UNIX: `./cmsinstall.sh silent`

Starting the CMS Service Automatically

By default, CMS is not configured to automatically start on startup. You can configure CMS to start automatically at startup.

To start CMS automatically on Windows or UNIX, follow the instructions in Appendix D: Starting the Connection Manager Server Automatically

Removing the CMS Service

To remove the CMS service on Windows or UNIX, follow the instructions in Appendix D: Starting the Connection Manager Server Automatically.

Upgrading TIBCO MFT Command Center

You can upgrade TIBCO MFT Command Center from a previous version.

If you are installing Command Centeron a machine that already has a version of Internet Server or Command Center installed, you must install Command Centerin a new empty directory and NOT over an existing old MFT installation. This does not apply when installing Service Packs as hotfixes.

Some steps in the upgrading process vary depending on the version of TIBCO MFT Command Center you have installed presently.

Both TIBCO MFT Command Center and TIBCO MFT Internet Server can be installed on the same server sharing a database as long as different ports are used. By default, TIBCO MFT Command Center uses port 8080 for HTTP and port 8443 for HTTPS. TIBCO MFT Internet Server uses port 7080 for HTTP and port 7443 for HTTPS.

You must always back up the database before you upgrade TIBCO MFT Command Center.

Upgrading Java JDK

When upgrading the Java JDK that is being used by TIBCO MFT Command Center, you have to update a few items of Java JDK before TIBCO MFT Command Center starts to use the new Java JDK.

Procedure

1. Follow either one of the procedures based on whether you are running TIBCO MFT Command Center on Windows or on UNIX.
    - If you are running TIBCO MFT Command Center on Windows as a service:
    a. Stop the TIBCO MFT Command Center service. Alternatively, use the shutdown script to stop the TIBCO MFT Command Center process.
    b. Navigate to the *MFTCC_Install*`\server\bin` directory and issue the `service remove` command.

c. Update the *JAVA_HOME* environment variable on the system pointing to the new JDK directory. And then, run the `java -version` command to verify the version.

d. If TIBCO MFT Command Center is installed on a Windows system, navigate to the *`MFTCC_Install`*`\server\bin` directory and issue the `service install` command to install TIBCO MFT Command Center to run as a service.

e. Start TIBCO MFT Command Center.

- If you are running TIBCO MFT Command Center on UNIX or on Windows when not started as a service:

a. Stop the Service using the `shutdown.sh` command.

b. Update the JAVA_HOME environment variable on the system to point to the new JDK directory.

c. Update the PATH to include the new Java JDK bin directory. Run the java –`version` command to verify the version.

d. Start TIBCO MFT Command Center.

Configuring BouncyCastle FIPS 140-2

This section describes the procedure to enable FIPS mode and configure BouncyCastle FIPS 140-2. See the following topics, before you configure Bouncy Castle FIPS 140-2:

- General FIPS Guidelines
- BouncyCastle Configuration Challenges
- BouncyCastle FIPS Restrictions
- FIPS Utility
- FIPS Script Format

Note: When running in BouncyCastle FIPS mode, only Elliptic Curve Diffie-Hellman ciphers are supported. In FIPS mode, RSA keys are not allowed to be used for encrypting/decrypting and then used again for signing/verifying. SSL/TLS is particularly prone to this problem, especially if keys are shared by the client and the servers. Therefore, only Elliptic Curve Diffie-Hellman ciphers are allowed for FTPS, Platform Server SSL, Platform Server Tunnel and HTTPS client and server requests. While Elliptic Curve ciphers are supported in recent client and server software, some old software does not support Elliptic Curve ciphers. You must ensure that all client and server software supports Elliptic Curve ciphers. See the Release Notes for the minimum versions required to use Elliptical Curve ciphers when using TIBCO® Managed File Transfer Platform Server for Windows or TIBCO® Managed File Transfer Platform Server for UNIX.

Procedure

1. Executing the fips Scripts.
2. Testing BouncyCastle FIPS Mode.
3. Enabling BouncyCastle FIPS Mode.
4. Disabling BouncyCastle FIPS Mode.
5. Update to java.security File When Enabling or Disabling FIPS Mode.
6. Manual Processing (After Enabling or Disabling FIPS Mode).

General FIPS Guidelines

Whenever possible, select **Use Default** to define a PGP private key or a protocol system key. When switching between FIPS and non-FIPS mode, you must only change the default key to the FIPS or non-FIPS key type.

- Use the "fips test" function to create a report that shows whether PGP and protocol keys are supported in FIPS mode.
- Prior to converting to FIPS mode, you must make sure that only FIPS supported keys are used. For PGP, this means that you must move away from El Gamal keys and use RSA keys instead. For both PGP and protocol keys, you must use keys of at least 2048 bits and only use the supported algorithms.
- Prior to enabling FIPS mode, you must make sure that every entry identified as "FAIL" in the "fips test" report has been converted to a FIPS supported public or private key. We suggest disabling keys that do not support FIPS and are no longer used. After a period of time, we suggest deleting these keys.

Note: 1024-bit system keys are not supported when running in FIPS mode and cannot be created when running in FIPS mode. If you create a 1024-bit system key while running in non- FIPS mode and attempt to use this key in FIPS mode, transfers fail. FIPS mode allows you to create 2048 or 3072-bit keys. Creating 4096-bit keys is not supported in FIPS mode, but 4096-bit keys can be used in FIPS mode.

BouncyCastle FIPS Configuration Challenges

BouncyCastle FIPS Certified Cryptography routines can be used with MFT when using Oracle Java, OpenJDK, and IBM Java. The BC FIPS jar has been designed and implemented to meet FIPS 140-2 Level 1 requirements.

Enabling FIPS mode is complex and time consuming. It requires a great deal of planning and testing before being used in a production environment. FIPS mode affects the following MFT components:

- Accessing the MFT Server through HTTPS
    - Internet Server Transfers

    - Internet Server and Command Center admin pages
- AS2 Client and Server
    - AS2 System Keys
- Key authentication for incoming requests
    - HTTPS
    - Platform Server
    - SSH
    - FTPS
- Key authentication for outgoing requests
    - HTTPS
    - Platform Server
    - SSH
    - FTPS
- Connecting to partner servers
    - HTTPS
    - Platform Server
    - SSH
    - FTPS
- PGP processing
    - Encrypting and decrypting data
    - Signing data and verifying signatures

One thing that makes it difficult to implement FIPS is that you may not have control over all of the keys that are used in your system. If a customer is using a PGP private key that is not FIPS compliant (that is, an El Gamal Key), the customer must change to a key that fully supports FIPS; otherwise, transfers will fail when FIPS mode is enabled. Likewise, if you connect to a partner's SSH Server that uses a 1024-bit key, connections to the server will fail when running in FIPS mode.

Hence, you must make local changes to support FIPS mode and your transfer partners must also make changes to support FIPS mode. This must be done before enabling FIPS mode.

BouncyCastle FIPS Restrictions

When using Bouncy Castle FIPS, the following restrictions exist:

1. BouncyCastle FIPS cannot be used on MFT versions prior to V8.2.
2. BouncyCastle FIPS supports only keystores in a BCFKS format when stored on the file system.
3. BouncyCastle FIPS does not support public or private key sizes less than 2048 bits.
4. The following PGP protocols are not supported:
    - El Gamal, CAST5, MD2, MD5, RipeMD
5. SSL/TLS ciphers are more limited.

FIPS Utility

MFT supplies a utility that enables you to easily enable or disable FIPS mode because BouncyCastle FIPS mode is more restrictive and more difficult to implement.

To implement BouncyCastle FIPS mode, you must first install MFT in a non-FIPS mode. On UNIX, the utility is a `fips.sh` script while on Windows, the utility is a `fips.bat` script. The scripts are distributed in the root directory of the MFT distribution.

The FIPS utility has three functions:

1. Test: Create a report on FIPS compatibility.
2. Enable: Turn on FIPS mode.
3. Disable: Turn off FIPS mode.

FIPS Script Format

The FIPS script can show the functions that will or will not work when in FIPS mode. Additionally, it can convert AS2 system keys to a format supported in FIPS mode.

```
fips { Test | Enable | Disable } {silent}
```

or

```
./fips.sh { Test | Enable | Disable } {silent}
```

Examples:

```
fips test
```

```
fips enable
```

```
fips enable silent
```

```
fips disable
```

```
fips disable silent
```

Keep in mind that the FIPS `enable` or `disable` functions do not do all the work required to enable or disable FIPS mode. The TIBCO MFT administrator also makes sure that only FIPS supported functions are used prior to enabling FIPS mode.

Executing the FIPS Scripts

The fips scripts use the parameters defined in the *`MFT-Install`*`/server/conf/server.xml` and the *`MFT-Install`*`/server/webapps/cfcc/WEB-INF/web.xml` to review your system for FIPS compatibility and to enable or disable FIPS mode.

You can run the fips script on a different MFT Server installation. For example, you can run a fips test on V8.0.2 or V8.1.0 to check the compatibility with FIPS. If you want to run the script on an existing MFT Server installation, follow this procedure.

Before you begin

You must install MFT 8.2.0 or higher prior to executing the fips script.

Procedure

1. Copy the `fips.sh` or `fips.bat` file to the installed system depending on whether the system is UNIX or Windows.

2. Edit file `fips.sh` or `fips.bat`. For example, `fips.bat` on Windows.
3. Make either of the following changes based on your requirement.

    - On Windows: In the `fips.bat` file, change the following line (at around line 7) from `set MFT_HOME="%CD%"` to `set MFT_HOME=c:\`*`YourMFTInstallDirectory`*.
    - On UNIX: In the `fips.sh` file, change the following line (at around line 19) from `MFT_HOME=$PWD` to `MFT_HOME=/opt/`*`YourMFTInstallDirectory`*.

    Important: *`YourMFTinstallDirectory`* must point to the MFT installation directory where the fips script is copied from. The script uses this directory to setup the java classpath.

4. Run the fips script. For example:
    - On Windows: `fips test` or `fips enable` or `fips disable`
    - On UNIX: `./fips.sh test` or `./fips.sh enable` or `./fips.sh` disable

Testing BouncyCastle FIPS Mode

The FIPS test checks for key sizes, keystore types, encryption algorithms and hashing/message digest algorithms. The "test" function produces a report called `fips-report.txt` in the current working directory. The report shows the components that fail FIPS testing and will therefore not work in FIPS approved mode.

Procedure

1. Enter one of the following commands:
    - On Windows: `fips test`
    - On UNIX: `./fips.sh test`

    The file `fips-report.txt` is created and shows the components that are FIPS compatible and the components that are not FIPS compatible. The FIPS test reports on the following capabilities:

    - Keystore status: tests the server HTTPS private key
    - PGP system keys

- PGP user/server public keys
- AS2 system keys
- Protocol user/server public keys
- Cipher suites
    - HTTPS cipher suites
    - SSL cipher suites

The **PGP System Keys**, **PGP User/Server Public Keys**, **AS2 System Keys**, and **Protocol User/Server Public Keys** display a status column. This column consists of these values:

Value	Description
Pass	The protocol and key size conforms to FIPS guidelines and works in FIPS mode.
Fail	The protocol and key size do not conform to FIPS guidelines and do not work in FIPS mode.
Action	The protocol and key size conforms to FIPS guidelines and works in FIPS mode. You must configure the software to use the FIPS supported key after running the FIPS Enable script.

Enabling BouncyCastle FIPS Mode

Procedure

1. Enter one of the following commands:
    - On Unix: `./fips.sh enable` or `./fips enable silent`

      or
    - On Windows:`./fips enable` or `fips enable silent`

   After you enable FIPS mode for an MFT Server, the following actions occur:
    - Adds BouncyCastle security providers.

- Converts the `server.xml` to a bcfks file and updates the `server.xml`.
- Updates the TLS ciphers in the `server.xml` to ciphers supported by FIPS mode.
- Converts all AS2 private keys in the database to a format compatible with FIPS mode.
- Adds the necessary environment parameters to the MFT startup script.

Note: When you use Oracle database and FIPS, the database user password must have a minimum of 14 characters.

Disabling BouncyCastle FIPS Mode

Procedure

1. Enter one of the following commands:
   - On Windows: `fips disable` or `fips disable silent`
   - On Unix: `./fips.sh disable` or `./fips disable silent`

   The following functions are performed when you disable FIPS mode for an MFT Server:
   - Removes BouncyCastle security providers.
   - Converts the `server.xml` to a .jks file and updates the `server.xml`.
   - Updates the TLS ciphers in the `server.xml` to ciphers supported by non-FIPS mode.
   - Removes the FIPS environment parameters in the MFT startup script.

Manual Processing (After Enabling or Disabling FIPS Mode)

Manually process the following keys as required:

Procedure

1. Manual Processing for AS2 System Keys.
2. Manual Processing for PGP Private Keys.

Manual Processing for AS2 System Keys

AS2 system keys are converted to a format that FIPS supports. The FIPS script creates copies of the protocol key and appends "[FIPS]" to the AS2 private key description.

Depending on how your transfer services are defined, you may need to do some manual work when enabling or disabling FIPS mode. This may be required because of the following reasons:

- FIPS mode does not support the old AS2 system key format.
- Non-FIPS mode supports the converted AS2 keys.

Procedure

1. Define the AS2 system key as default key by selecting **Use Default**

   MFT selects the default key and finds the correct key to use. You do not need to do anything else, unless a key was deleted after executing the FIPS script.

2. Review the AS2 system key for compatibility with the FIPS mode.
   - For the AS2 Server, the server definition defines the AS2 system key. You must update all of the server definitions to make sure that the proper key type is used. To update server definitions, perform the following actions on the admin page:
     - Go to **Servers > Manage Servers**.
     - In the Selection criteria box, change Server Type to AS2 and click **Search**. A list of AS2 Servers is displayed in the results table.
     - Click **Server name** for each server and open the **AS2 Options** box. Make sure that the **Encryption System Key** and **Signing System Key** values are correct for the FIPS mode you use.

Manual Processing for PGP Private Keys

The fips script test function shows the public or private keys that do not work when executing in FIPS mode. Hence, you must review the definitions to make sure that you are not using a key that is not supported.

When running in FIPS mode, PGP private keys fail under the following circumstances:

- When using one of the following algorithms: El Gamal, CAST5, MD2, MD5, RipeMD
- When the key size is less than 2048 bits.

PGP private keys are defined for transfers and servers.

Procedure

1. You can define the PGP private key to be used in one of the following ways:
    - Use a Default Private Key. When this is selected, MFT selects the default key. You must ensure that the default key is a supported key.
    - Specify a PGP private key to use. You must review all transfer definitions that use PGP to make sure that a supported PGP private Key is used. You must also review all server definitions that use PGP to make sure that a supported PGP private key is used. Navigate to the following page to configure the PGP private key to use for these services.
        - Navigate to **Servers > Manage Servers**, or
        - Navigate to **Transfers > Manage Transfers**.

Update to java.security File When Enabling or Disabling FIPS Mode

When enabling or disabling FIPS mode, the fips script updates the following file: *java-home*`/jre/lib/security/java.security`

The following line in this file is changed:

```
ssl.KeyManagerFactory.algorithm=SunX509
```

or

```
ssl.KeyManagerFactory.algorithm=IbmX509
```

gets changed to

```
ssl.KeyManagerFactory.algorithm=PKIX
```

> **Note:** Because the fips script updates the `java.security` file, it is important that MFT is the only application that uses this Java installation. If necessary, we suggest installing a Java version that only MFT uses.

Prior to enabling or disabling FIPS, the script checks if it has access to this file. If it does, the script continues. If it does not have access to this file, the script terminates with the following message:

```
You do not have access to C:\Program Files\Java\jdk-21\jre\lib\security
```

Changing the Default Logos

You can customize some logos within TIBCO MFT Command Center.

You can customize the following logos:

- The following logo is used by TIBCO MFT Command Center Administrator and is displayed in the upper-left corner of your browser. It is named as `mft-cc-logo.png` with a size of 204X35, and is located in the *`MFTCC_Install`*`\server\webapps\cfcc\view\images` directory.

- The following logo is used by TIBCO MFT Command Center Administrator login. It is named as `corporate_logo1.png` with a size of 460X99, and is located in the *`MFTCC_Install`*`\server\webapps\cfcc\login\images` directory.

Procedure

1. Go to the directory where the logo is located.
2. Rename the logo by adding `.old` after the file extension. Example: `logo.png.old`.
3. Copy your new logo into the directory and ensure that the file name, type, and size match the original file in the directory.
4. Refresh your browser.

Changing the Default Navigation Bar Color

You can customize the color of the navigation bar in TIBCO MFT Command Center.

To customize the color of the navigation bar, add the required CSS statements to the `<MFT Install>/server/webapps/cfcc/view/css/customize.css` file in the section titled `MAIN NAVIGATION STYLES`.

You can use the following example to change the color of the navigation bar to green:

```
.pl-globalnav__navbar
{ background-color: green !important; }
.pl-globalnav-navbar__brand
{ background-color: green !important; }
.pl-globalnav-navbar__action:hover
{ border-bottom: 4px solid lightgreen !important; }
.pl-globalnav-navbar__action.is-selected
{ border-bottom: 4px solid lightgreen !important; }
```

Adding Login Disclaimer to the Login Page

To add a disclaimer to the login page, perform the following steps:

Procedure

1. Open the `server/webapps/cfcc/login/disclaimer.jsp` file.
2. Add the required content in the following sample code:

```
<div class="custom-disclaimer">

<!-- put your content here -->
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam
pretium, sapien et laoreet rhoncus, eros erat congue quam, posuere
placerat lacus orci scelerisque magna. Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Donec varius odio odio. Sed nisi ex,
lobortis id interdum non, suscipit ut dolor. Aliquam sed euismod
lorem, non pellentesque erat. Vivamus urna mauris, imperdiet at
luctus et, hendrerit et diam. Fusce porttitor tellus nisi, non
condimentum odio dignissim nec. Etiam volutpat vulputate eros, a
efficitur metus. Proin quis dui id orci lobortis feugiat vel nec
enim.
</p>
<p>
Maecenas aliquet ornare nisi, et aliquet justo pretium sit amet.
Praesent hendrerit varius ligula, in efficitur nulla faucibus eget.
Vestibulum tempus ut sapien sagittis accumsan. Praesent pretium
lorem gravida tellus sagittis, eu mattis tellus feugiat. Mauris
ultrices pellentesque rhoncus. Donec egestas ligula urna, vel
pulvinar est ullamcorper eu. In sit amet facilisis ipsum. Cras
tristique risus sed orci mollis accumsan. Pellentesque habitant
morbi tristique senectus et netus et malesuada fames ac turpis
egestas. Pellentesque mi libero, elementum a odio sed, fringilla
eleifend sem. Maecenas facilisis, ipsum vitae laoreet rutrum, nisl
nisi ullamcorper metus, id cursus ipsum lectus faucibus arcu.
Aliquam ut feugiat dui. Nunc ut massa ut mauris volutpat porta.
Maecenas a imperdiet est.
</p>

</div>
```

3. Open the `server/webapps/cfcc/login/styles/login.css` file.
4. Edit the following sample and add the style to the custom-disclaimer selector:

```
.custom-disclaimer{
 position: absolute;
 top: 10px;
 right: 10px;
 width: 30%;
 background-color: white;
 border-radius: 5px;
 padding: 10px;
}
```

Uninstalling TIBCO MFT Command Center

Note: If you have CMS installed, you have to first uninstall CMS. See Uninstalling Connection Manager Server for information about removing the CMS service.

If TIBCO MFT Command Center is installed as a Windows service, see Removing the TIBCO MFT Command Center Service on Windows for details about removing the TIBCO MFT Command Center service.

To remove TIBCO MFT Command Center from Windows or UNIX, stop the TIBCO MFT Internet Server and delete the *MFT_Install* directory.

Uninstalling Connection Manager Server

You can remove CMS if you want.

If CMS has been installed as a Windows service, see the Removing the CMS Service topic for more information.

To remove CMS from Windows or UNIX, delete the *CMS_Install* directory.

Appendix A: Installation Worksheet

This worksheet serves as the location where you can collect information that is used throughout the installation and configuration ofTIBCO MFT Command Center.

You can use this worksheet to gather information before the installation of TIBCO MFT Command Center. You might also use the default values provided by the installation program.

Web Server Information

1. Have you downloaded and installed the Java JDK: _____
2. Is the *JAVA_HOME* variable set: _____

Database Information

1. What is the DNS or IP address and port number for the Internet Server database: _____
2. What database administrator ID and password should be used: _____
3. What is the database table or schema name: _____
4. Ensure that you have downloaded the JDBC driver database.

Java Keystore Information

Note: Using a trusted Certificate Authority (CA) certificate is strongly recommended. If you do not have a CA certificate, the TIBCO MFT Command Center installer creates a self-signed certificate keystore. You can manually add a new CA certificate keystore after the installation is complete.

1. What is the path and file name of your java keystore: _____
2. What is your keystore password: _____

3. What is the alias for the private key: _____

TIBCO MFT Command Center Application Information

1. What is the DNS or IP address of the server where the TIBCO MFT Command Center application is being installed? _____
2. What context root do you want to use (default option is cfcc): _____ _____
3. In what directory should log files be kept (defaults option is the `install` directory): _ _____

LDAP Information

Note: This information is optional because you might not be using LDAP for authentication.

1. LDAP server type: _____
2. DNS or IP address of the LDAP server: _____
3. What is the LDAP port number: _____
4. What is the LDAP administrator DN: _____ _____
5. What is the password for the user DN: _____

License Activation

Note: This information is necessary because this product requires an activation license to be applied.

1. If you are using an activation server, what is the connection URL: _____ _____
2. Or, if you are using a local license, what is the full path to the directory where the license file is stored: _____

Appendix B: Certificate Update Guideline

TIBCO MFT Command Center uses the following type of certificate:

- HTTPS certificate: used for communicating with TIBCO MFT Command Center using HTTPS (HTTP over SSL).

Updating HTTPS Certificate

To obtain a new HTTPS certificate from CA, a certificate request must be issued.

Note:
- The commands listed here are only examples and do not include all the options that the **keytool** program offers. Careful consideration must be taken when generating your key pair for your environment. Consult with your web Administrator.
- Each certificate requires a separate keystore.
- CA might have specific options required for creating an HTTPS certificate. Review the instructions provided by the CA before generating the certificate request.

Procedure

1. Run the following sample command to generate a Java keystore and key pair where the certificate will be considered valid for 365 days:

```
keytool -genkey -v -alias cfcc -keyalg RSA -keysize 2048 -keypass
changeit -keystore MFTCC_Install\keystore\newkeystore.jks -
storepass
changeit -validity 365
```

   In the sample command, the *keypass* and *storepass* values are the same. These two values must match each other. It is good practice to use the same keystore and

storepass values that are used to create the original keystore. This way you will not have to update the keystore password in the product configuration files.

The `keytool` utility will display messages requesting more information about the certificate request. When the `keytool` utility prompts `What is your first and last name`. You must enter the DNS name that is used to access TIBCO MFT Command Center. For example, you can enter mft.yourcompany.com as the DNS name. This DNS name is used as the Common Name (CN) in the certificate. HTTPS requires CN to match the DNS name used to access the HTTPS server.

2. Generate a certificate request.

   You can use the following sample command:

   ```
   keytool -v -certreq -alias cfcc -file MFTCC_
   Install\keystore\cfcc.csr -
   keypass changeit -keystore MFTCC_Install\keystore\newkeystore.jks -
   storepass changeit
   ```

3. Submit the certificate request file created in the previous sample command to CA.
4. Install the CA certificate into the TIBCO MFT Command Center keystore by performing the following steps:
   a. Save the certificate returned by the CA to a file *Cert_File*.
   b. Run the following `keytool` command to import the certificate:

      ```
      keytool -v -import -alias cfcc -trustcacerts -file Cert_File -
      keystore
       Keystore_File_Name
      ```

      Note: Some CAs now issue an intermediate certificate along with the main certificate. If this is true for your CA, import certificates using unique aliases to the keystore created in Step 1. This step is required to prevent the client from receiving a certificate warning.

5. Go to the *`MFTCC_Install`*`\server\conf` directory and change the keystore path in the `server.xml` file to update the MFT server to use the new keystore.
   a. Look for the connector associated with the HTTPS port.

b. Update the `keystoreFile` parameter to point to the new keystore.

c. If the password is changed, update the `keystorePass` parameter with the new keystore password.

> **Note:** You can rename your old keystore file, for example, `org.keystore.jks`. And then rename the new keystore to have the old file name in the same location. This way no changes are needed to the `server.xml` file which is located in the *`MFTCC_Install`*`\server\conf` directory, and then you can go to Step 6.

6. Restart TIBCO MFT Command Center.
7. Verify that the MFT server is listening on the defined port.
8. Perform a file transfer to verify that TIBCO MFT Command Center is functioning correctly.

Appendix C: Starting the TIBCO MFT Command Center Service Automatically

By default, the application server is not configured to automatically start on startup. You can set up an automatic start for the TIBCO MFT Command Center embedded application server on UNIX and Windows systems.

Starting the TIBCO MFT Command Center Service on Windows Automatically

You can set up an automatic start for the TIBCO MFT Command Center embedded application server on Windows systems.

Procedure

1. Check whether the *JAVA_HOME* system environment variable is configured on your server.

   You can perform the following steps to set the variable:

   a. Open your System Properties window and click the **Advanced** tab.

   b. Click **Environment Variables**.

   c. In the Environment Variables window, search for the *JAVA_HOME* variable in the System variables panel.

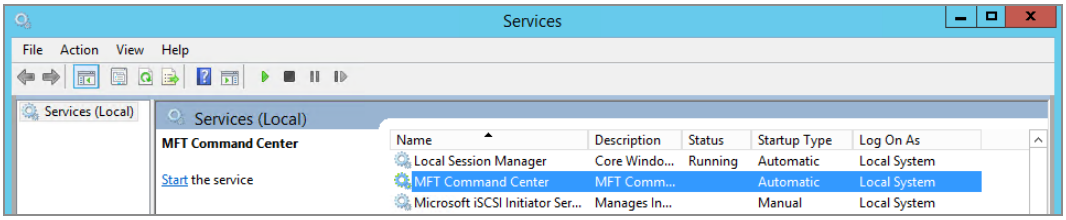
   d. Set the *JAVA_HOME* variable to make it points to your Java JDK file. For example, `C:\Program Files\Java\jdk-21`.

      If you cannot find the *JAVA_HOME* variable in the list, you must add the *JAVA_HOME* variable pointing to your Java JDK file.

> **Note:** If you create or change a variable, you must open a new Windows shell in order for MFT to recognize the new environment variable.

2. Navigate to the *MFTcc_Install*`\server\bin` directory and stop your present TIBCO MFT Command Center application using the `shutdown` command.
3. Run the following install command from the same directory: `service install`.

   After running the script, open the Services window and see the TIBCO MFT Command Center service listed as follows:



> **Note:** The TIBCO MFT Command Center service is installed by default using the **Manual** startup option.

4. Restart Windows.
5. Open the Services window, right-click TIBCO MFT Command Center and click **Properties**.
6. Set **Startup Type** to Automatic and click **OK**.

   TIBCO MFT Command Center will start automatically at the next startup of Windows.

Starting the TIBCO MFT Command Center Service on UNIX Automatically

You can use a number of methods to start the TIBCO MFT Command Center on different UNIX/Linux operating systems automatically.
This example is for **systemd** service for the Red Hat Linux Enterprise operating system; however, it has been tested on other UNIX and Linux distributions. The instructions for setting automatic start on Linux are as follows:

Procedure

1. Add the *JAVA_HOME* variable to the `setenv.sh` file which is located in the *`MFTCC_Install`*`/server/bin` directory.

```
#!/bin/sh
if [ -z "$JAVAMEM_MB_MIN" ]; then
  JAVAMEM_MB_MIN=512m
fi
if [ -z "$JAVAMEM_MB_MAX" ]; then
  JAVAMEM_MB_MAX=4096m
fi

CATALINA_OPTS="-Xms$JAVAMEM_MB_MIN -Xmx$JAVAMEM_MB_MAX"
JAVA_HOME=/java/jdk21
JAVA_OPTS="-Duser.language=en -Duser.country=US -
Dfile.encoding=UTF-8 -Dssh.maxWindowSpace=4096000 -
Djdk.tls.ephemeralDHKeySize=2048 -
Dcom.tibco.tibcrypt.httpshandler=false -
Dorg.apache.catalina.session.StandardSession.LAST_ACCESS_AT_S
TART=false -
Dorg.apache.catalina.session.StandardSession.ACTIVITY_CHECK=t
rue -Dcom.ibm.jsse2.overrideDefaultTLS=true -
Dsun.net.http.allowRestrictedHeaders=true -
Dmaverick.disableDirectoryCheck=true -XX:
+HeapDumpOnOutOfMemoryError"
.........................................
```

2. Create a SystemD script

> **Note:**
> - Run the following commands as a root or sudo user.
> - Absolute paths must be provided for variables in angle brackets (<>).

To create a unit file, create the `/etc/systemd/system/mftcc.service` file and include the following text:

```
[Unit]
Description=MFT Command Center Service
[Service]
Type=oneshot
ExecStart=<MFTCC>/server/bin/startup.sh
RemainAfterExit=true
StandardOutput=journal
ExecStop=<MFTCC>/server/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

Run the following command to enable the service to start automatically after a reboot:

```
systemctl enable mftcc.service
```

Run the following command to start the MFT Command Center service:

```
systemctl start mftcc.service
```

Run the following command to check if the process is started:

```
systemctl status mftcc.service
```

Note: The next time Linux is restarted, the Command Center service starts automatically.

Removing the TIBCO MFT Command Center Service on Windows

You can remove the automatic start feature of TIBCO MFT Command Center on Windows.

Procedure

1. Stop the TIBCO MFT Command Center service.
2. Navigate to the *MFTCC_Install*\server\bin directory and run the following command:

```
service remove
```

Result

The following message is displayed:

```
The service 'MFT_Command_Center' has been removed.
```

Removing the TIBCO MFT Command Center Service on Linux

You can remove the automatic start feature of TIBCO MFT Command Center on Linux:

Note: Ensure that the procedure is run by a root or sudo user.

Procedure

1. Run the following command to stop the Command Center Service:

```
systemctl stop mftcc.service
```

2. Run the following command to disable the Command Center Service:

```
systemctl disable mftcc.service
```

3. Delete the following Command Center unit file:

```
/etc/systemd/system/mftcc.service
```

Appendix D: Starting the Connection Manager Server Automatically

By default, Connection Manager Server (CMS) is not configured to start automatically on startup. You can set up an automatic start for the Connection Manager Server application server on UNIX and Windows systems.

Starting the Connection Manager Server Service On Windows Automatically

You can set up the Connection Manager Server application server to auto-start on Windows systems.

Procedure

1. Check whether the *JAVA_HOME* system environment variable is configured on your server.

   Perform the following steps to set the variable:

   a. Open the **System Properties** window and click the **Advanced** tab.

   b. Click **Environment Variables**.

   c. In the **Environment Variables** window, go to the **System variables** panel and search for the *JAVA_HOME* variable.

   d. Configure the `JAVA_HOME` variable in such a way that it points to your Java JDK file. For example, `C:\Program Files\Java\jdk-21`.

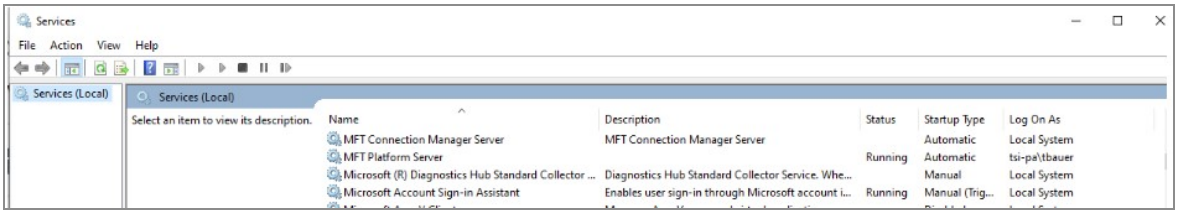
      If you cannot find the *JAVA_HOME* variable in the list, you must add the *JAVA_HOME* variable pointing to your Java JDK file.

> **Note:** If you create an environment variable, you must open a new Windows shell for MFT to recognize the new environment variable.

2. Navigate to the *CMS_Install*`\server\bin` directory and stop your present CMS application using the `shutdown` command.
3. Run the following install command:

```
service install
```

4. After running the script, open the **Services** window and the MFT Connection Manager Service is listed as follows:



> **Note:**
> The Connection Manager Server service is installed by default using the **Automatic startup** option. After installing the service, click **MFT Connection Manager Service** and start the service. When you restart Windows, the CMS service starts automatically.

Starting the TIBCO MFT Connection Manager Server Service On UNIX Automatically

You can use a number of methods to automatically start a Connection Manager Server on different UNIX/Linux operating systems.

This example is for the **systemd** service for the Red Hat Linux Enterprise operating system. However, it has been tested on other UNIX and Linux distributions. Perform these steps to set up auto-start on Linux:

Procedure

1. Add the `JAVA_HOME` variable to the setenv.sh file located in the *CMS_Install/cmsserver/bin* directory.

```
#!/bin/sh
if [ -z "$JAVAMEM_MB_MIN" ]; then
  JAVAMEM_MB_MIN=512m
fi
if [ -z "$JAVAMEM_MB_MAX" ]; then
  JAVAMEM_MB_MAX=4096m
fi

CATALINA_OPTS="-Xms$JAVAMEM_MB_MIN -Xmx$JAVAMEM_MB_MAX"
JAVA_HOME=/java/jdk21
JAVA_OPTS="-Duser.language=en -Duser.country=US -
Dfile.encoding=UTF-8 -Dssh.maxWindowSpace=4096000 -
Djdk.tls.ephemeralDHKeySize=2048 -
Dcom.tibco.tibcrypt.httpshandler=false -
Dorg.apache.catalina.session.StandardSession.LAST_ACCESS_AT_S
TART=false -
Dorg.apache.catalina.session.StandardSession.ACTIVITY_CHECK=t
rue -Dcom.ibm.jsse2.overrideDefaultTLS=true -
Dsun.net.http.allowRestrictedHeaders=true -
Dmaverick.disableDirectoryCheck=true -XX:
+HeapDumpOnOutOfMemoryError"
.........................................
```

2. Create a SystemD script

> **Note:**
> - Run the following commands as a root or sudo user.
> - Absolute paths must be provided for variables in angle brackets (<>).

To create a unit file, first create the `/etc/systemd/system/mftcms.service` file and then include the following text:

```
[Unit]
Description=MFT Connection Manager Server Service
[Service]
Type=oneshot
ExecStart=<MFTCMS>/server/bin/startup.sh
RemainAfterExit=true
StandardOutput=journal
ExecStop=<MFTCMS>/server/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

Run the following command to enable the service to start automatically after a reboot:

```
systemctl enable mftcms.service
```

Run the following command to start the MFT CMS service:

```
systemctl start mftcms.service
```

Run the following command to check if the process is started:

```
systemctl status mftcms.service
```

Note: The next time Linux is restarted, the CMS service starts automatically.

Removing the TIBCO MFT Connection Manager Server Service On Windows

You can remove the automatic start feature of the Connection Manager Server on Windows.

Procedure

1. Stop the Connection Manager Server service.
2. Navigate to the `CMS_Install\server\bin` directory and run the following command:

```
service remove
```

Result

The following message is displayed:

```
The service 'MFT_Connection_Manager_Server' has been removed.
```

Removing the TIBCO MFT Connection Manager Server Service On Linux

You can remove the automatic start feature of Connection Manager Server on Linux.

Note: Ensure that the procedure is run by a root or sudo user.

Procedure

1. Run the following command to stop the CMS Service:

```
systemctl stop mftcms.service
```

2. Run the following command to disable the CMS Service:

```
systemctl disable mftcms.service
```

3. Delete the following CMS unit file:

```
/etc/systemd/system/mftcms.service
```

Appendix E: Setting HTTP SSL Ciphers

By default, MFT uses configure TLS services (HTTPS, FTPS, OFTP2, and Platform Server) to use the most secure ciphers. Only TLSv1.2 is supported by default.

By default, ciphers are set to the TLS protocol using 128-bit encryption or higher. You can edit the `server.xml` file which is located in the *`MFTCC_Install`*`\server\conf` directory to set certain SSL ciphers.

A default HTTP connector is defined in this file, as shown in the following example:

```
<Connector SSLEnabled="true" URIEncoding="UTF-8" acceptCount="128"
allowHostHeaderMismatch="false" ciphers="TLS_AES_256_GCM_SHA384,TLS_AES_
128_GCM_SHA256,TLS_CHACHA20_POLY1305_SHA256,TLS_ECDHE_ECDSA_WITH_AES_
256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_
WITH_CHACHA20_POLY1305_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_
ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_
SHA256,TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDH_RSA_WITH_AES_256_
GCM_SHA384,TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDH_RSA_WITH_AES_
128_GCM_SHA256,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_128_GCM_
SHA256"
. . . . . .
 sslEnabledProtocols="TLSv1.2, TLSv1.3" sslProtocol="TLS"
tcpNoDelay="true"
. . . . . .
>
```

The following example forces client connections to maintain cipher strengths of 128 bits or higher. The ciphers in this example are from Oracle Java 8 update 40.

```
ciphers="TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_
256_CBC_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_ECDH_ECDSA_WITH_AES_
256_CBC_SHA384,TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384,TLS_DHE_RSA_WITH_
AES_256_CBC_SHA256,TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,TLS_ECDHE_ECDSA_
WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_
AES_256_CBC_SHA,TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDH_RSA_WITH_
AES_256_CBC_SHA,TLS_DHE_RSA_WITH_AES_256_CBC_SHA,TLS_DHE_DSS_WITH_AES_
256_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_
AES_128_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDH_ECDSA_WITH_
AES_128_CBC_SHA256,TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,TLS_DHE_RSA_
```

```
WITH_AES_128_CBC_SHA256,TLS_DHE_DSS_WITH_AES_128_CBC_SHA256,TLS_ECDHE_
ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_
WITH_AES_128_CBC_SHA,TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDH_RSA_
WITH_AES_128_CBC_SHA,TLS_DHE_RSA_WITH_AES_128_CBC_SHA,TLS_DHE_DSS_WITH_
AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_
WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_
WITH_AES_256_GCM_SHA384,TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDH_
RSA_WITH_AES_256_GCM_SHA384,TLS_DHE_RSA_WITH_AES_256_GCM_SHA384,TLS_DHE_
DSS_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_
RSA_WITH_AES_128_GCM_SHA256,TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,TLS_
ECDH_RSA_WITH_AES_128_GCM_SHA256,TLS_DHE_RSA_WITH_AES_128_GCM_
SHA256,TLS_DHE_DSS_WITH_AES_128_GCM_SHA256"
```

The following example will force client connections to maintain cipher strengths of 256 bit or higher. The ciphers in this example are from Oracle Java 8 update 40.

```
ciphers="TLS_RSA_WITH_AES_256_CBC_SHA256"
```

Note: Only certain browsers will support 256-bit cipher strength.

In these examples, the ciphers are limited in default connector to show how to change the ciphers. Limiting the cipher to one is not realistic, this is only for demonstration purposes:

```
<Connector SSLEnabled="true" URIEncoding="UTF-8" acceptCount="128"
ciphers="TLS_RSA_WITH_AES_256_CBC_SHA256" clientAuth="false"
compression="off" connectionLinger="-1" connectionTimeout="60000"
disableUploadTimeout="true" enableLookups="true"
keystoreFile="C:\MFTIS\keystore\keystore.jks"
keystorePass="PWD:encryptedkeystorepassword" keystoreType="JKS"
maxKeepAliveRequests="100" maxThreads="150" port="443"
protocol="org.apache.coyote.http11.Http11Protocol" proxyPort="0"
redirectPort="-1" scheme="https" secure="true" server="MFTServer"
socket.txBufSize="131072" sslEnabledProtocols="TLSv1.2"
sslProtocol="TLS" tcpNoDelay="true"
trustManagerClassName="com.proginet.sift.tomcat.ssldap.TrustAllMgr"/>
```

After you have saved your changes, you must restart the application server.

Appendix F: Customizing Translation Tables

TIBCO MFT Command Center is shipped with four ASCII to EBCDIC conversion tables to convert ASCII characters to EBCDIC characters and vice versa.

By default, the `Comtblg.dat` file which is located in the *`MFTCC_Install`*`\server\webapps\`*`context`*`\translate` directory is used by the system. The following table lists the conversion tables:

Conversion	Description
`Comtblg.classic`	The `comtblg.dat` file shipped with prior versions (before version 7.2).
`Comtblg.cp037`	Extended ASCII table that is based on IBM Code page 037.
`Comtblg.cp1047`	Extended ASCII table that is based on IBM Code page 1047.
`Comtblg.dat`	ASCII/EBCIDIC table used by MFTPS at run time. By default, it is a copy of the `Comtblg.cp037` file.

Note: The `Comtblg.dat` file is used by the system. If one of the other conversion tables needs to be used or a customized table is created, rename the existing `Comtblg.dat` file and copy the new table to the `Comtblg.dat` file. The default file used for conversion must be named `Comtblg.dat`.

Occasionally, the default translation table is not exactly what is needed. In these situations, an administrator can define a new translation table to be used by the TIBCO MFT Command Center installation.

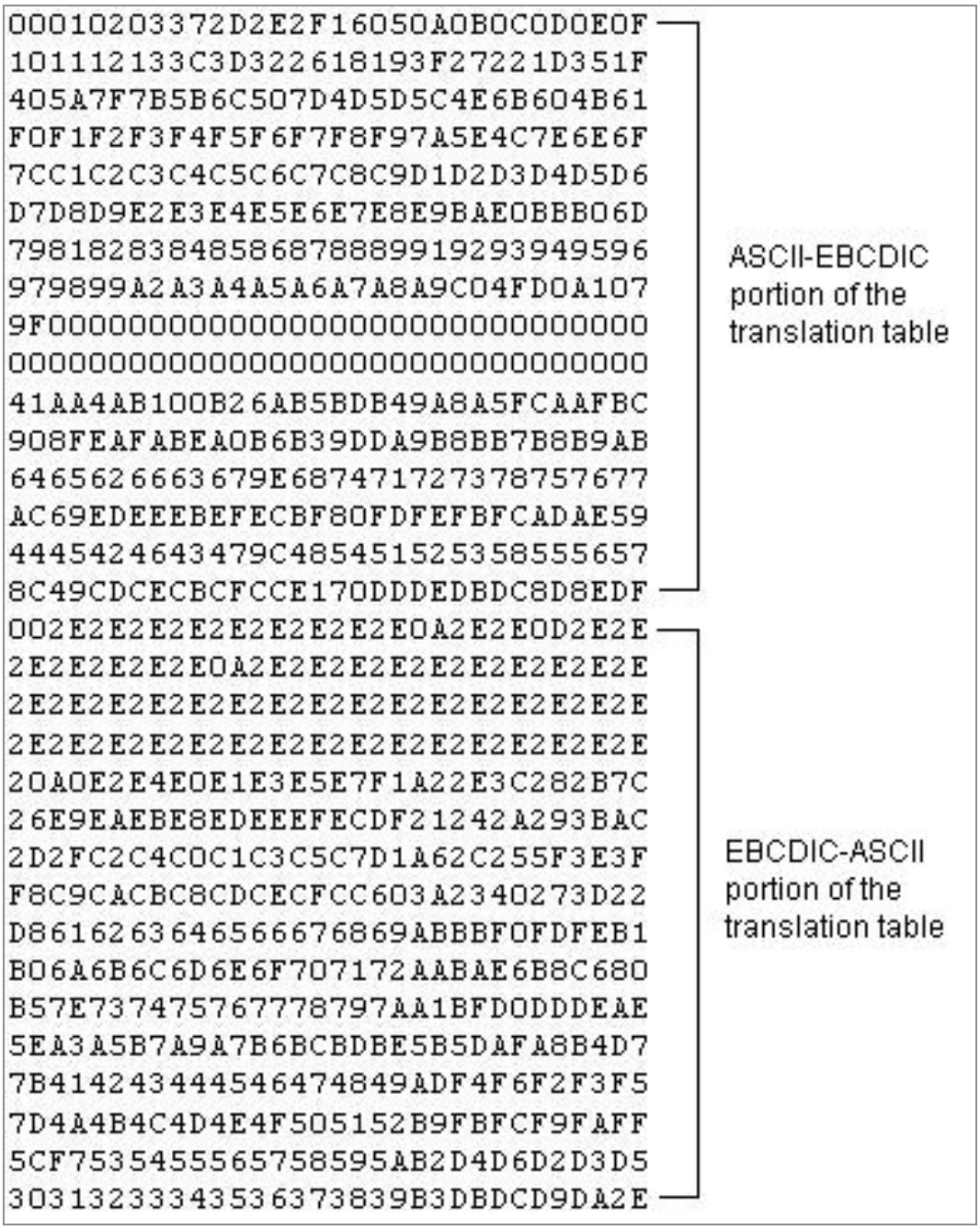
The following example demonstrates how to alter the text JSY contained in a file to read CAT on the remote z/OS system:

Procedure

1. Create a customized translation table.

a. Navigate to the *MFTCC_Install*/server/webapps/*context*/translate directory and copy the Comtblog.cp037 file to an empty directory on the TIBCO MFT Command Center web server, and then rename it as Comtblg.dat.

   The Comtblg.dat file contains the following table, which converts data from ASCII to EBCDIC character and from EBCDIC to ASCII character.



The following figure shows the table being placed in an Excel spreadsheet for demonstration purpose only:

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

b. To convert from an ASCII system (Windows) to an EBCDIC system (z/OS), look up the EBCDIC character for each ASCII character and replace it with the EBCDIC character that you want.

   For example, the ASCII value for J is 4A; if you want to translate J to 4A, you have to go the chart above and locate the 4th column and slide your finger to the right until you are in the A column; the EBCDIC value for J is D1. To translate J to C, replace D1 with C3, which is the EBCDIC value for C. Follow the same process to translate S to A and Y to T.

2. Replace the existing `Comtblg.dat` file.

   a. Navigate to the *`MFTcc_Install`*`/server/webapps/`*`context`*`/translate` directory, rename the existing `Comtblg.dat` file as `org.Comtblg.dat`.

   b. Copy the new `Comtblg.dat` file that is customized step 1 to this folder.

      This file is now your default conversion table used by the system.

Appendix G: Using the TIBCO Hawk Microagent

TIBCO MFT Command Center supports TIBCO Hawk microagents that communicate to TIBCO Hawk through TIBCO Rendezvous. This allows the TIBCO Hawk console to monitor TIBCO MFT Command Center instances. After you have completed the MFT installation and before the MFT Hawk interface can be configured and used, you must perform the following tasks:

1. Set the `MFT_CLASSPATH` environment variable to include the path to the TIBCO Hawk and TIBCO Rendezvous lib directories.
2. Set the `PATH` or `LD_LIBRARY_PATH` environment variable to include the path to the TIBCO Rendezvous C runtime libraries.

Note: You must complete these steps for each TIBCO MFT Command Center instance that you want to monitor.

Setting the MFT_CLASSPATH environment variable

You must set the `MFT_CLASSPATH` environment variable to the TIBCO Hawk and TIBCO Rendezvous lib directories.

For example:

Windows

If TIBCO Rendezvous is installed at `C:\tibco\tibrv\8.6` and TIBCO Hawk is installed at `C:\tibco\hawk\6.2`

```
SET MFT_CLASSPATH=C:\tibco\tibrv\8.6\lib\*;C:\tibco\hawk\6.2\lib\*
```

UNIX, Linux, and zLinux

If TIBCO Rendezvous is installed at `/tibco/tibrv/8.6` and TIBCO Hawk is installed at `/tibco/hawk/8.6`

```
export MFT_CLASSPATH=/tibco/tibrv/8.6/lib/*:/tibco/hawk/6.2/lib/*
```

Note: The asterisk (*) includes all jar files in this directory.
Windows uses the semicolon (;) as a path separator and UNIX and Linux use the colon (:) as a path separator.

Setting the PATH or LD_LIBRARY_PATH environment variable

You must set the `PATH` or `LD_LIBRARY_PATH` environment variable to the TIBCO Hawk and TIBCO Rendezvous lib directories.

For example:

Windows

If TIBCO Rendezvous is installed at `C:\tibco\tibrv\8.6`

```
SET PATH=C:\tibco\tibrv\8.6\bin\*;%PATH%
```

UNIX, Linux, and zLinux

If TIBCO Rendezvous is installed at `/tibco/tibrv/8.6`

```
export LD_LIBRARY_PATH=/tibco/tibrv/8.6/lib
```

Note: Do not include the asterisk (*). Windows uses the `PATH` environment variable and UNIX, Linux uses the `LD_LIBRARY_PATH` environment variable. TIBCO Rendezvous on Windows stores the libraries in the bin directory, but on UNIX, Linux, and zLinux they are located in the lib directory.

Setting the java.library.path system property

You must edit the java.library.path system property to include the Rendezvous C runtime libraries directory.

- For Unix/Linxus/zLinux:

  Update the `setenv.sh` file in the server/bin directory.

- For Windows running as a service:

  Update the `service.bat` file in the server/bin directory.

- For Windows not running as a service:

  Update the `setenv.bat` file in the server/bin directory.

Following is an exert from the `setenv.bat` file that has been updated:

```
-Djava.library.path="%CATALINA_BASE%/webapps/cfcc/WEB-INF/license-
files/;D:/tibco/tibrv/8.6/bin/"
```

Note: You need both the license directory and the Rendezvous directories here. On Windows use a semi-colon character to separate the directories. On UNIX/Linux/zLinux use a colon character.

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 Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the TIBCO® Managed File Transfer Command Center Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, 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

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

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

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: https://scripts.sil.org/OFL

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG 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 a specific version of Cloud SG software on a specific operating system platform.