TIBCO® Managed File Transfer Command Center

Managed File Transfer Overview

Version 8.7.0 | October 2025

Contents

TIBCO® Managed File Transfer Components

TIBCO® Managed File Transfer (MFT) includes the following major functional components that facilitate the secure transfer of data through a network.

- **MFT Command Center**

  It is used to configure and manage MFT Internet Server and MFT Platform Servers.

- **MFT Internet Server**

  It is used to perform file transfers, generally through the internet with open protocols such as SFTP, FTP, HTTPS, AS2, and the proprietary Platform Server protocol.

- **Connection Manager**

  It allows MFT Internet Server running in the DMZ to open all ports from the internal network to the external network.

- **MFT Platform Server**

  It is used to perform file transfers, generally in the internal network. MFT Platform Servers are developed for each platform and use a proprietary protocol.

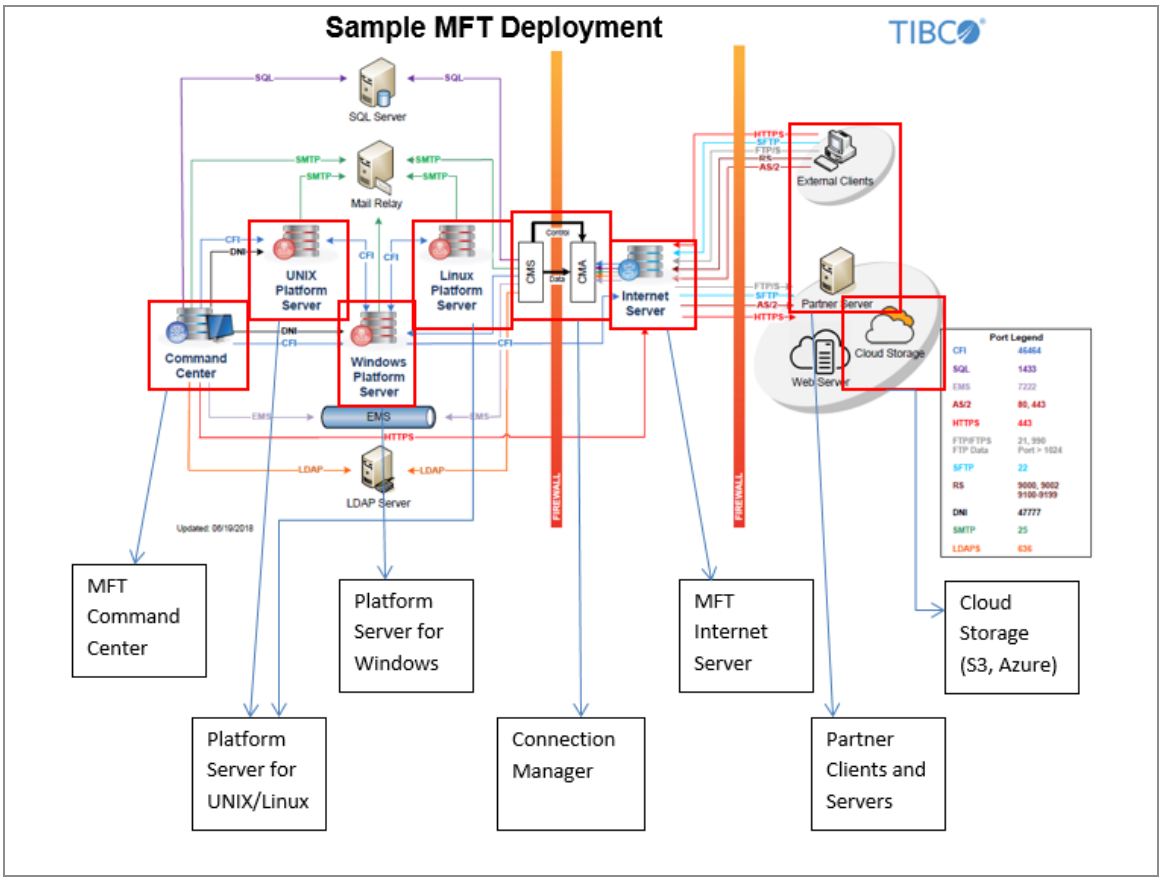
- **pDNI**

  It is an event-driven service that performs file transfers when a file is created or modified.

Each functional component is either a TIBCO product or a part of a TIBCO product. For more information about each component, see Detailed Explanation of TIBCO MFT Components.

TIBCO MFT Components

TIBCO MFT components are illustrated in the following diagram of a typical MFT deployment.



Each of these TIBCO MFT components are discussed in detail in the following sections.

- MFT Command Center
- MFT Internet Server
- Connection Manager
- MFT Platform Server
- pDNI

MFT Command Center

MFT Command Center is a TIBCO® Managed File Transfer Command Center product.

MFT Command Center is the management component for MFT Internet Server and MFT Platform Servers. You can use the administrative component in MFT Internet Server to configure all of the parameters (users, servers, transfers) that executes file transfers. MFT Command Center extends these capabilities to provide additional functionality.

MFT Command Center runs in the internal network; it should not execute in the DMZ because you can configure and execute Internet Server transfers and Platform Server transfers.

Note: For any recent changes in MFT Command Center, see this section in the latest version of *TIBCO® Managed File Transfer Command Center - Managed File Transfer Overview*.

Supported Functionality in MFT Command Center

By default, MFT Command Center includes support for the following functions:

1. High Availability (HA). The MFT Scheduler and JMS interface work in a HA Active/Active environment.
2. LDAP or Database authentication
3. OIDC and SAML SSO (Single Sign On) support
4. Multi Factor Authentication (MFA)
5. Certificate and/or password authentication
6. APIs
    - REST calls to perform many configuration capabilities.

- Command line interface to perform many configuration capabilities.

7. Logging and reporting of all administration changes (includes before and after images of the changes).

MFT Command Center Extends Capabilities of MFT Internet Server

MFT Command Center extends the capabilities of MFT Internet Server to perform the following functions:

1. Alerts: With alerts, you can configure actions based on events or non-events. There are three types of alerts:
   - Transfer Events: Alerts can be triggered by a transfer.
   - Non-Transfer Events: Alerts can be triggered by a transfer not executing.
   - Logon: Alerts can be triggered by a user logon.
2. Connection Manager Nodes: Connection Manager allows MFT Internet Server instances running in the DMZ to open all connections from the internal network to the external network. For more information, see Connection Manager.
3. JMS Interface: The JMS interface allows you to initiate Internet Server or Platform Server transfers. Also provides a mechanism for MFT Internet Server to write data to, or read data from, JMS queues.
4. View Active Internet Server Transfers: MFT Command Center allows you to view transfers executing on all Internet Server instances in the MFT cluster.
5. Logging of all Admin Changes: This includes before and after images of the changes.

MFT Command Center Extends the Capabilities of MFT Platform Servers

MFT Command Center also extends the capabilities of MFT Platform Servers to perform the following functions:

1. Collection Service: It provides a centralized location for collecting and reporting on MFT Platform Server transfers. Also allows you to execute alerts when an MFT Platform Server transfer is collected.
2. Execute Platform Transfers: It allows platform transfers to be initiated from a centralized location.
3. Configure MFT Platform Servers: It allows you to configure the following MFT Platform Server components:
    - Node definitions
    - Profiles and responder profiles
4. View Active Platform Server Transfers: MFT Command Center allows you to view transfers executing on defined MFT Platform Server instances. Note that this capability requires MFT Platform Server V8.1 or higher.

Other MFT Command Center Capabilities

MFT Command Center also includes the following management capabilities:

1. Reporting: It allows you to execute reports on transfers, users, alerts, and AS2 transfers.
2. Status Server: It displays the status of target servers that MFT Internet Server communicates with.
3. Scheduler: It allows you to schedule the following actions:
    - Perform Platform Server transfers
    - Perform Internet Server transfers
    - Send emails
    - Execute commands
    - Execute Java class
    - Perform maintenance functions
    - Notify users of expiring keys
    - Purge database tables

- Purge log files

4. DNI Daemons: DNI Daemons allow you to manage pDNI templates from a centralized location. For more information, see Event Driven Processing in MFT Platform Servers.

MFT Internet Server

MFT Internet Server is the TIBCO® Managed File Transfer Internet Serverproduct.

MFT Internet Server is the file transfer component. MFT Internet Server supports many open protocols, and it also supports the Platform Server protocol. MFT Internet Server has an administrative component that allows you to configure all of the parameters (users, servers, transfers) to allow file transfers to execute. For additional capabilities, you can install MFT Command Center.

MFT Internet Server can be installed in the DMZ or in the internal network; when executing in the DMZ, you must disable the administrative capability because it allows you to configure Internet Server transfers.

Think of MFT Internet Server as a protocol converter. Here is an example:

- **File Upload**

  An SFTP client connects to MFT Internet Server to upload a file. The MFT Internet Server is configured to send the file to a target MFT Platform Server:

  SFTP --> Internet Server --> Platform Server for Linux

  As MFT Internet Server receives packets from the SFTP client, it converts the packets to the Platform Server protocol and sends the packets to the MFT Platform Server. All of this is done in a streaming mode. Packets are not written to a disk file; the file can be sent directly to the location where the data is processed.

- **File Download**

  An HTTP client connects to MFT Internet Server to download a file. The MFT Internet Server is configured to receive the file from a target FTPS Server:

  HTTPS <-- Internet Server <-- FTPS Server

  As MFT Internet Server receives packets from the FTPS Server, it converts the packets to the HTTPS protocol and sends the packets to the HTTPS client. All of this is done in a streaming mode. Packets are not written to a disk file; the file can be sent directly from the location where the file is stored.

Note: For any recent changes in MFT Internet Server, see also the latest version of *TIBCO® Managed File Transfer Internet Server - Managed File Transfer Overview*.

Supported Functionality in MFT Internet Server

By default, MFT Internet Server supports the following functions:

1. High Availability (HA). Because MFT configuration information is stored in the database, MFT Internet Server runs HA Active/Active.
2. High volume. Each Internet Server can run hundreds of concurrent transfers.
3. LDAP or Database authentication
4. OIDC and SAML support
5. Multi Factor Authentication (MFA)
6. Certificate/Key and/or password authentication
7. APIs
    - REST calls to perform many configuration capabilities.
    - REST calls to perform file transfers.
    - Command Line interface to perform many configuration capabilities.
8. Logging of all admin changes. Includes before and after images of the changes.
9. Mailbox capability. Allows you to upload a file to an MFT repository and send an email to the end user. When the end user clicks on a link in the email, they can download the file from the MFT repository (after logging in and being authenticated).
10. File sharing capability. Users can share directories with other users.
11. Four eyes capability.
12. Postprocessing actions.

Supported Protocols in MFT Internet Server

MFT Internet Server supports the following client protocols:

1. FTP, FTPS
2. SFTP
3. HTTP/HTTPS(through a browser client or an API)
4. AS2
5. Platform Server Protocol
6. OFTP2

MFT Internet Server supports protocols for connecting to the following target servers:

1. FTP, FTPS
2. SFTP
3. Platform Server (Proprietary Protocol)
4. HTTP/HTTPS Server
5. AS2
6. Amazon S3 Buckets
7. Custom Server
8. Azure File Share, Block Blob, or ADLS Gen2
9. JMS Server
10. Local Accessible Storage (I.e. NAS or NFS)
11. HDFS
12. Google Cloud Storage or BigQuery
13. OFTP2 Servers
14. Microsoft SharePoint Servers
15. Mailbox Servers
16. Email Servers

MFT Internet Server Security Capabilities

MFT Internet Server ensures the security of file transfers and the file transfer data by implementing the following capabilities:

1. PGP Encryption/Decryption: MFT can PGP-encrypt or PGP-decrypt data in a streaming mode. PGP provides the following capabilities:
    a. It provides non-repudiation. MFT can identify the signature of the client that encrypted and signed the data.
    b. PGP provides an extra level of encryption. Clear text FTP transfers can send encrypted data that can only be decrypted by a PGP client with the correct private key.
    c. This adds a second level of security to secure protocols. For example, you can PGP encrypt data sent in an encrypted SSH connection. This provides two high levels of encryption.
    d. PGP can also automatically compress and decompress data.

    When transferring sensitive data or data that contains financial transactions, we strongly suggest using double levels of encryption: SFTP and PGP.
2. Key/Certificate and/or Password authentication: Key/Certificate authentication provides the highest level of authentication security. The client key or certificate associated with the private key must be uploaded to MFT and associated with a user before it can be used. Hence, only users with the client system key and the system key password can connect to MFT. Key/Certificate is supported for the following protocols:
    a. Platform Server protocol
    b. HTTPS
    c. FTPS
    d. SFTP

    Client connections to MFT servers support key/certificate authentication.

    MFT connections to target servers also support key/certificate authentication.
3. Rights assignments: MFT provides granular rights to allow specific admin or transfer functionality. No access is allowed if you do not have the required rights.
4. Password lockout functionality: MFT can be configured to lockout users after a pre-

defined number of invalid logon attempts.

5. File transfer access: No access is allowed by default. `TransferRight` must be assigned to a user before any transfers can be performed. Additionally, transfer definitions must be defined for a user before any transfers can be performed.
6. User configuration: A user can be configured so that the user can upload files without getting access to see any files or directory lists.
7. Virtual aliases: The actual location of the files and directories is abstracted from the end user through the use of virtual aliases. For example, the following definitions can be made for a user:
    - Tax data can be located on a target UNIX Platform Server.
    - Payroll can be located on a target UNIX SFTP Server.
    - Invoices can be located on a customer's FTPS Server.
8. File uploads and downloads: Data can be pulled (download) from a target server or pushed (upload) to a target server. This allows MFT to initiate all file transfers for and from a target customer. MFT also allows the customer to initiate upload or download transfers.

MFT Internet Server Postprocessing Actions

MFT Internet Server provides the following support for postprocessing actions:

- Postprocessing is defined in the transfer definition.
- Up to four postprocessing actions can be defined.
- All postprocessing actions are executed on MFT Internet Server or on a target server.
- Each postprocessing action can be executed on success or failure.
- The following postprocessing commands are supported:
    - Execute command
    - CALLJCL (MFT Platform Server for z/OS only)
    - CALLPGM (MFT Platform Server for z/OS only)
    - SUBMIT (MFT Platform Server for z/OS only)
- Up to 256 bytes of data can be passed to the postprocessing action. Symbolic

parameters can be used to pass transfer-related information to the Postprocessing data field.

- When the target server is a Platform Server, the post processing actions are passed directly to the target Platform Server.
- When the target server is a Local Server, the postprocessing actions are executed on the Internet Server.
- When the target server is an FTP/FTPS Server, the postprocessing actions are limited to `Delete` or `Rename` commands.
- When the target server is an SFTP Server, the postprocessing actions are limited to `Delete` or `Rename` commands. In addition to `Delete` or `Rename`, there is a special format of the PPA that can execute commands on a target SFTP Server.

Connection Manager

Connection Manager is a part of the TIBCO® Managed File Transfer Command Center product.

Connection Manager is used together with MFT Internet Server when Internet Server is running in the DMZ. Many installations do not allow TCP connections to be opened from the DMZ network to the internal network. Connection Manager solves this problem by opening all connections from the internal network to the external network.

Components of Connection Manager

There are two components to Connection Manager:

1. Connection Manager Server: Runs in the internal network. Establishes control connections to the Connection Manager Agent. Accepts a connection request from the Agent over the control connection and creates TCP Connections to the Agent.
2. Connection Manager Agent: Runs in the DMZ. Requests connections over the control connections and accepts connection request from the Connection Manager Server.

Connection Manager Agent is shipped with MFT Internet Server. Connection Manager Server is distributed with MFT Command Center. MFT Command Center is required to configure the Connection Manager Agent and Server.

Connection Manager Installation

When MFT Internet Server is installed in the DMZ, it typically requires connections to servers in the internal network. This can include the following servers:

- LDAP Server
- JMS Server
- Platform Servers

- SMTP Servers
- SSH Servers
- FTP/FTPS Servers

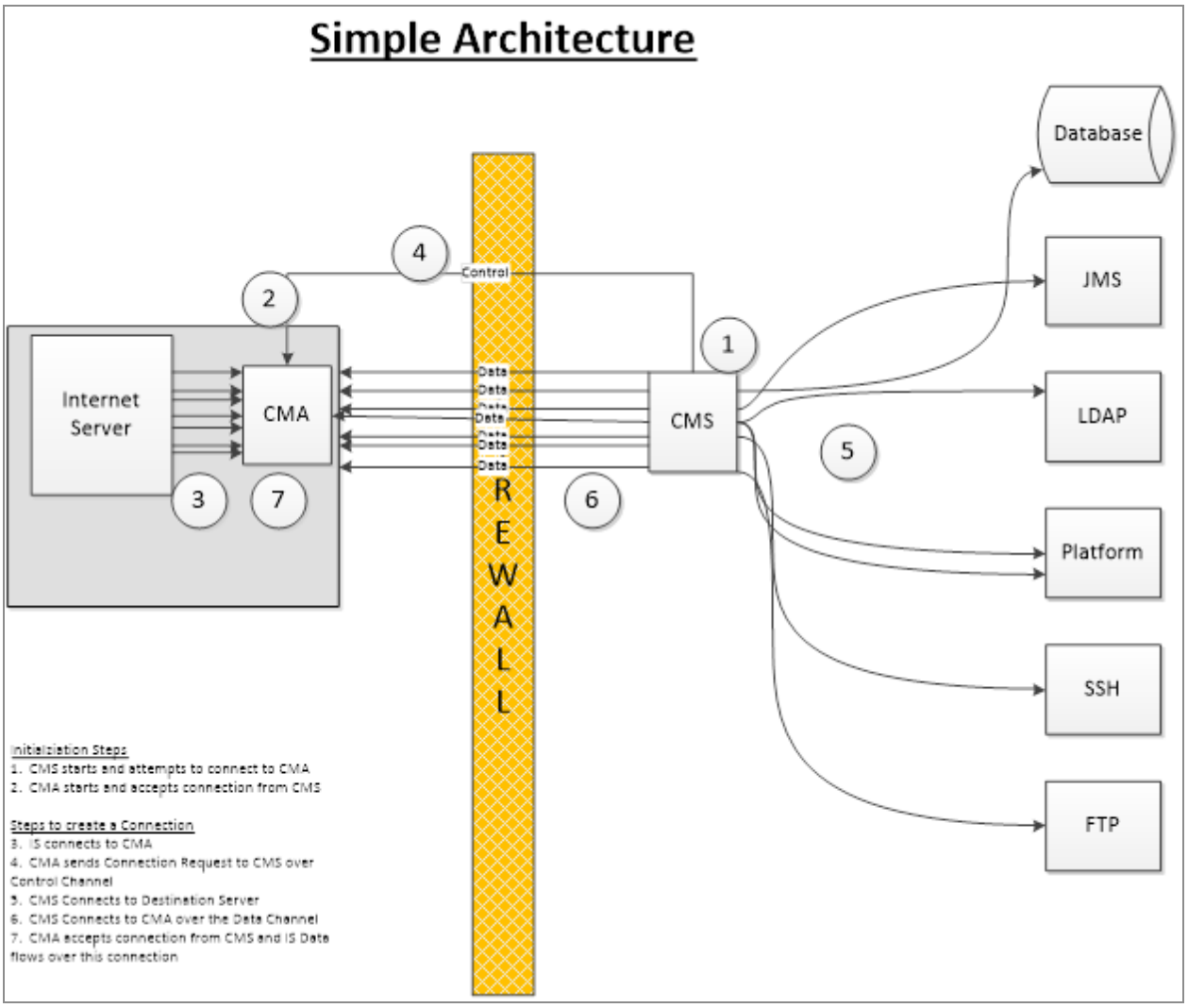
Without the Connection Manager, MFT Internet Server must be able to open TCP connections from the DMZ to the internal network. With the Connection Manager, you can open all TCP connections from the internal network to the DMZ. Firewalls frequently need exceptions to allow connections to be opened from the DMZ to the internal network.

The following examples illustrate the installation of Connection Manager by using different methods:

- Simple Architecture
- Complex Architecture
- Two Tier DMZ Architecture

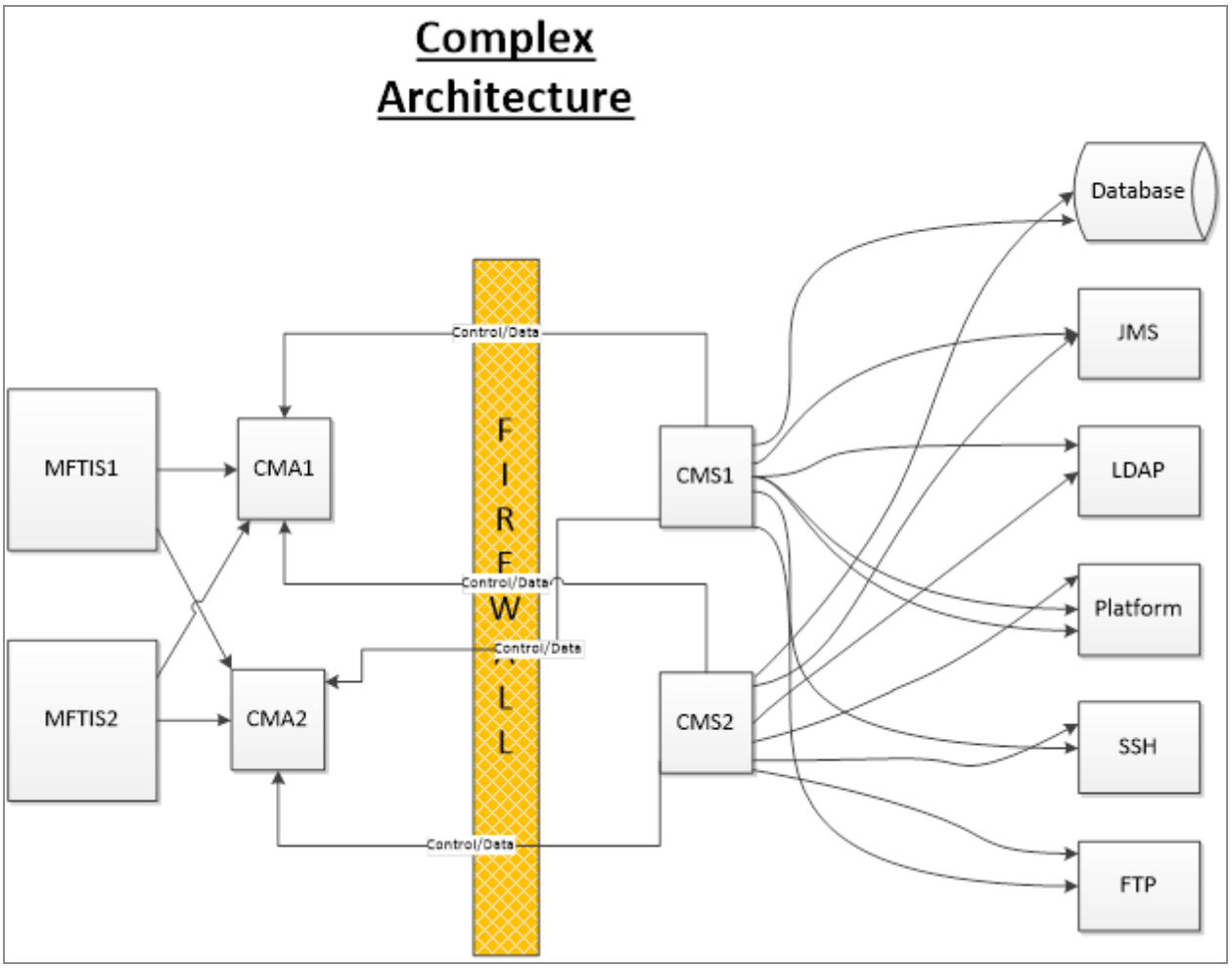
Simple Architecture

The following example describes a simple Connection Manager installation. Internet Server needs to access multiple servers in the internet network (Database, JMS, LDAP, and so on).



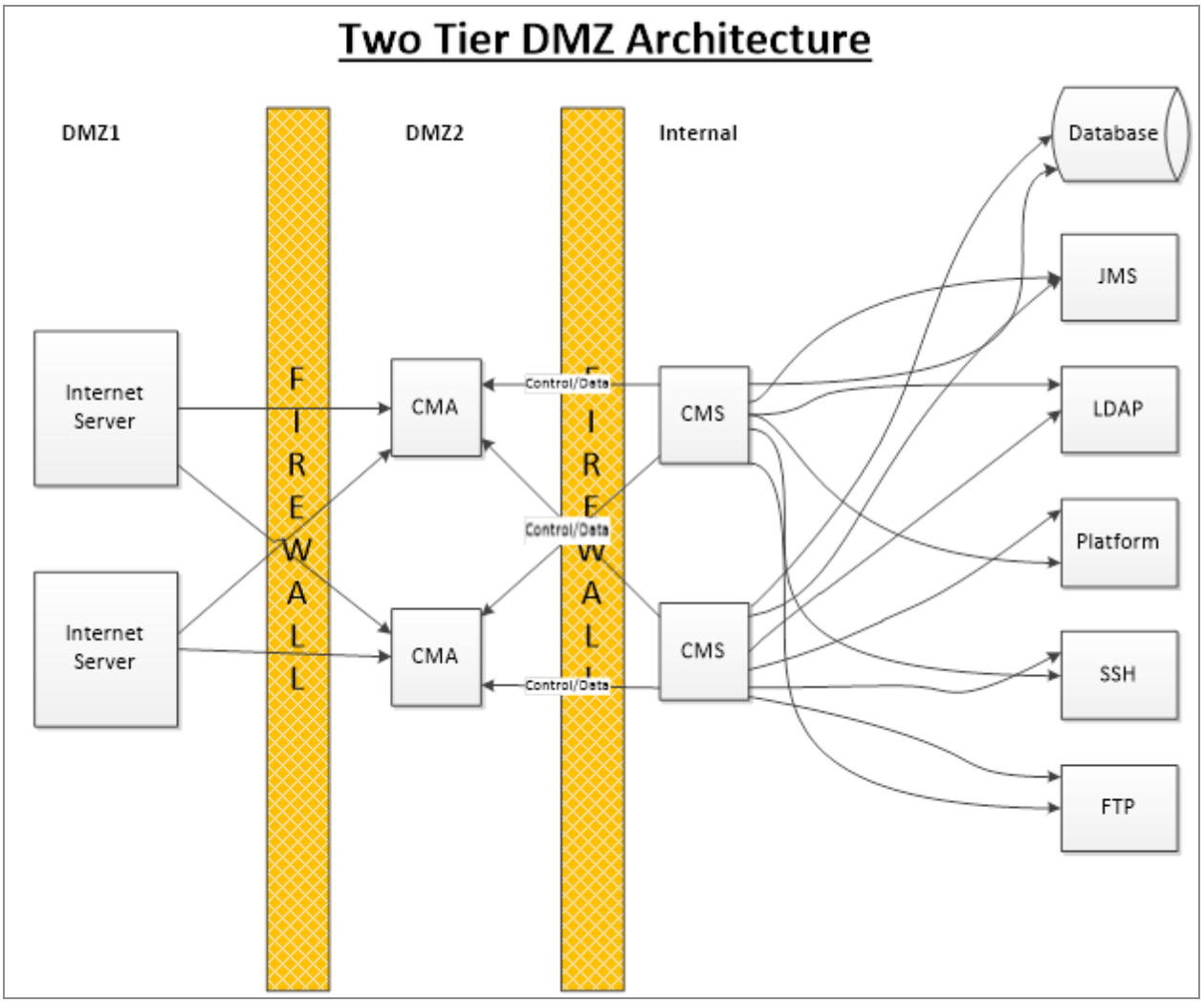
Complex Architecture

The following example describes a complex Connection Manager installation. The Internet Server machines are connected to multiple Connection Manager Agents (CMA). The Connection Manager Agents have connections to multiple Connection Manager Servers (CMS).



Two Tier DMZ Architecture

The following example describes a two tier Connection Manager installation. For this to work, the Internet Server in DMZ1 must be able to initiate connections to the Connection Manager Agents in DMZ2. Connections between DMZ2 and the internal network are initiated by the Connection Manager Server.


Two Tier DMZ Architecture
DMZ1
DMZ2
Internal
Internet Server
Internet Server
FIREWALL
CMA
CMA
Control/Data
Control/Data
Control/Data
FIREWALL
CMS
CMS
Database
JMS
LDAP
Platform
SSH
FTP

MFT Platform Server

MFT Platform Server is available as TIBCO® Managed File Transfer Platform Server for UNIX, TIBCO® Managed File Transfer Platform Server for z/Linux, TIBCO® Managed File Transfer Platform Server for z/OS, TIBCO® Managed File Transfer Platform Server for Windows, and TIBCO® Managed File Transfer Platform Server for IBM i products.

MFT Platform Servers are peer-to-peer file transfer servers that typically execute in the internal network. MFT Platform Servers are built specifically for each platform. These servers are meant for high-volume transfers so they are efficient and fast. The following hardware and software platforms are supported:

Hardware

- Solaris SPARC
- Solaris Intel

Software

- Z/OS
- Windows
- IBM i
- z/Linux
- Linux
- AIX

MFT Platform Server Features

MFT Platform Servers have the following features:

- Proprietary protocol to transfer files.
- Initiating a file send or a file receive to a target Platform Server.
- Initiating a directory send or directory receive from a target Platform Server.

- Initiating a file send or file receive to a target Internet Server.
- Initiating a directory send or directory receive from a target Internet Server.
- Support for the following levels of data encryption:
    - No data encryption
    - AES 256 encryption
    - TLS session to negotiate symmetric keys for AES 256 encryption
    - Encapsulate all data through TLS tunnel
- Authenticating and passing credentials in multiple ways.
    - User profiles: Users can transfer files without knowing the target system credentials.
    - Responder profiles: Users can transfer files without providing any credentials to the target server.
- With TLS or Tunnel Mode, you can configure the software to accept only specific certificates.
- When running on z/Linux, Linux, or UNIX, the Platform Server Daemon can run as root or non-root.
- When running on Windows, requests can be validated against the responder profiles or Active Directory.
- When running on z/Linux, Linux, or UNIX requests can be validated against the responder profiles or the password or shadow password files. PAM authentication is also supported.
- Command Center can manage some Platform Server functionality, such as the following functions:
    - View completed Platform Server transfers
    - View and update node definitions
    - View and update profile and responder profile definitions
    - Initiate a transfer
- Performing postprocessing actions when a transfer completes. This is discussed in detail later in this document.

- Running a command on a target system.
- Sending an email when a transfer completes, either successfully or unsuccessfully.
- Perform authorization using the security of the UNIX or Windows platform.

MFT Platform Server Preprocessing and Postprocessing Actions

Preprocessing actions define the action that can be taken before a transfer starts. Based on the return code from the preprocessing action, transfer can continue, terminate, or be retried at the next retry interval. Preprocessing requires Platform Server V8.1 or higher. Preprocessing is not supported on Internet Server.

Postprocessing actions define the action that should be taken when a transfer completes, either successfully or unsuccessfully. Postprocessing actions are supported on all Platform Servers. Limited postprocessing actions are also supported on Internet Servers as well.

- Up to four preprocessing and postprocessing actions can be defined.
- Preprocessing and postprocessing are defined on the command line or in the transfer process statements.
- Each preprocessing or postprocessing action can be executed in the initiator or the responder.
- Each postprocessing action can be executed on success or failure.

The following preprocessing and postprocessing commands are supported:

- Execute command
- CALLJCL (MFT Platform Server for z/OS only)
- CALLPGM (MFT Platform Server for z/OS only)
- SUBMIT (MFT Platform Server for z/OS only)

Up to 256 bytes of data can be passed to the preprocessing or postprocessing action. Symbolic parameters can be used to pass transfer related information to the Preprocessing or Postprocessing data field.

Event Driven Processing in MFT Platform Servers

MFT Platform Servers have support for event-driven processing through the Directory Named Initiation (DNI) feature. Here is an overview of DNI capabilities integrated with MFT Platform Server.

Note: The pDNI (perl DNI) capabilities are described in more detail in the "pDNI" section.

- **MFT Platform Server for z/OS DNI**

  MFT Platform Server for z/OS supports initiating transfers based on the creation of a file. Since z/OS does not include a date-modified timestamp on its files, MFT Platform Server for z/OS cannot initiate transfers based on a file being modified. MFT Platform Server for z/OS DNI can also detect data in an MQ queue and initiate data transfer from the MQ queue to a target system.

- **MFT Platform Server for Windows**

  MFT Platform Server for Windows includes an integrated DNI capability that can be managed by the MFT Platform Server for Windows Administrator.

  MFT Platform Server for Windows also supports the pDNI that is described in the "pDNI" section.

- **MFT Platform Server for z/Linux and UNIX**

  MFT Platform Server for z/Linux and UNIX supports the pDNI that is described in the "pDNI" section.

pDNI

pDNI is a Perl based event-driven processing that works with TIBCO® Managed File Transfer Platform Server for UNIX, TIBCO® Managed File Transfer Platform Server for z/Linux, and TIBCO® Managed File Transfer Platform Server for Windowsproducts.

TIBCO Perl Directory Named Initiation (DNI) Installation and Operations Guide is contained within the `dni.tar` file.

The dni.tar file is distributed with the following products in a zip or tar format:

- MFT Internet Server and Command Center:

  `<MFT-Install>/distribution/dni`

- MFT Platform Server for Windows:

  `<TIBCO__HOME>/MFT Platform Server`

- MFT Platform Server for UNIX:

  `<$CFROOT>/dni`

pDNI Features

pDNI has the following features:

1. Monitors directories to detect files created or modified.
2. Waits for a predefined interval before sending and receiving a file to make sure that a file has not been modified.
3. Support for the following functionalities:
   a. DNI Send: Sends files to a target location.
   b. DNI Receive: Send files from a target location.
   c. FTP Receive: Receives files from an FTP Server (specifically the MFT Internet Server FTP Service).
4. Monitors subdirectories.

5. Supports High Availability in an Active/Passive mode.
6. Supports file and directory REGEX to limit the files transferred.
7. Supports directory scanning based on day of the week and time of the day.
8. Supports pre-transfer commands to limit the files that are transferred.
9. DNI Send and DNI Receive can execute any command; the default is to execute a cfsend/cfrecv(UNIX) or ftmscmd(Windows). But it allows you to execute any script or command.
10. Can execute up to 15 transfers at the same time.
11. Many DNI templates can run at the same time on a server. The only limit is the size of the machine where pDNI is executing.

pDNI can be configured manually on each machine. In addition, pDNI can be configured from a centralized Command Center. pDNI supports a daemon that allows the following functionality:

1. Add, update, or delete pDNI templates.
2. Start pDNI templates.
3. Stop pDNI templates.
4. View Template log files.

MFT High Availability and Failover

This section is intended for users requiring redundancy and scalability in TIBCO MFT production environments. It covers the following TIBCO products:

- Web server-based products for file transfer:
    - TIBCO® Managed File Transfer Internet Server
- Web server-based products for administration:
    - TIBCO® Managed File Transfer Command Center
- Platform Server:
    - TIBCO® Managed File Transfer Platform Server for UNIX
    - TIBCO® Managed File Transfer Platform Server for Windows
- Event-driven processing:
    - pDNI

Common High Availability Requirements for MFT Command Center and Internet Server

To create a highly available MFT environment for Command Center and Internet Server, you require a load balancer that is configured to support an Active/Passive or Active/Active configuration. This section covers aspects of load balancing MFT.

Requirements

- Load Balancer
- Highly Available Database Cluster Environment: Ensure the database, which holds all application configuration data, is always available.

Load Balancer Prerequisites

- Session Awareness: When a client connects to an MFT server through a load balancer, all subsequent requests from the client during that session must be handled by the same MFT server. For HTTPS connections, configure the load balancer to support HTTP Sticky Sessions. For the FTP protocol, both control and data channels must be directed to the same MFT server. Use the Source IP address persistence.
- Source NATTING: Ensure that when a client receives a reply from the server, the reply comes from the load balancer address and not directly from the server.
- Protocol support for HTTPS, AS2, SFTP, FTP/FTPS, OFTP2, and Platform Server.
- Load balancer support for HTTPS, AS2, SFTP, FTP/FTPS, OFTP2, and Platform Server protocols. Certain protocols, such as FTPS, are not supported on all load balancing devices.
- The Platform Server protocol is a single connection TCP-based protocol and can be supported by most load balancers.

The Load Balancer's vendor must provide technical support for load balancing devices. The following are the general guidelines for creating a virtual pool on the load balancer:

1. Configure a virtual IP address on the load balancer.
2. Configure a virtual pool of servers on the load balancer.
3. Add each MFT Internet Server as a member of the virtual pool.
4. For Active/Passive Load Balancing:
    a. Configure the priority of the primary MFT server so that all connections route to the Active server unless the Active server is unavailable.
    b. Configure the remaining servers in the pool as Passive. These servers should only accept connections if the primary server is unavailable.
5. For Active/Active Load Balancing:
    a. Configure each server in the pool with the same priority. Client requests are routed to the server with the fewest active connections.
    b. Passive servers can still be used in an Active/Active environment for redundancy and failover capability. Configure these servers with a lower priority so they only accept connections if the primary Active servers are unavailable.

TIBCO MFT Internet Server and TIBCO MFT Command Center Configuration

Note: TIBCO MFT Internet Server and TIBCO MFT Command Center store all configuration data in a central database. You can add or remove instances from the MFT environment without losing any application configuration data.

Extracting Originating IP Address for HTTPS and SFTP Connections

When using a Load Balancer, the IP address received by MFT Internet Server and MFT Command Center is the IP address of the Load Balancer. To get the originating Client IP address for HTTPS and SFTP protocols, perform the following steps:

Note: The Load Balancer must be configured to save the originating IP Address and pass it to the MFT HTTPS and SFTP servers. Not all Load balancers support this configuration for HTTPS and SFTP.

HTTPS

Configure the Load Balancer to send the originating client IP address to MFT in the X-Forwarded-For HTTP header. Ensure that MFT is configured to accept this header from a Load Balancer.

There are two ways to configure MFT to accept the X-Forwarded-For HTTP header:

1. Update the `web.xml` parameter `LoadBalancerIPAddressList` to include the IP addresses of all Load Balancers that can connect to the MFT server. You can configure multiple IP addresses separating them by a comma. If the Load Balancer IP address is dynamic, set the parameter to `*All` to accept the `X-Forwarded-For` HTTP header from any IP address. Use this when running in a managed cloud environment where the Load Balancer IP Address can change.
2. Update the `server.xml` file to enable the `RemoteIP` valve. The `internalProxies` parameter of this valve defines the Load Balancer IP addresses. Use the `|` character as a delimiter when configuring multiple IP addresses. If the `internalProxies` parameter is not defined, the default IP addresses are `127.0.0.0/8|169.254.0.0/16|10.0.0.0/8|192.168.0.0/16`. This parameter is not

used by the MFT application; it is used by internal Tomcat services, such as the Rate Limit filter and the `localHostAccess` file.

SFTP

Configure the load balancer to send the originating client IP Address to MFT during the SSH session negotiation. Ensure that MFT is configured to accept this property from the load balancer.

To configure MFT to accept the originating SSH Client IP address from the load balancer, update the `web.xml` parameter `LoadBalancerIPAddressList` to include the IP Addresses of all load balancers that can connect to the MFT server. You can configure multiple IP Addresses separating them by a comma. If the Load Balancer IP address is dynamic, set this parameter to `*All` to accept the SSH originating IP address from any Load Balancer IP address. Use this when running in a managed cloud environment where the Load Balancer IP Address can change.

Note: The `web.xmlLoadBalancerIPAddressList` parameter is used for both HTTPS and SSH/SFTP connections. Setting it for HTTPS connections sets it for SSH/SFTP connections. Similarly, setting the parameter for SSH/SFTP connection also sets it for the HTTPS connections.

Settings Stored on Each Server

In case a new instance must be deployed, the following items are local to each Internet Server and should be backed up :

- Certificate keystores must match the external DNS name used by the Load Balancer if HTTPS encryption is not offloaded to the load balancer.
- Any files used for customizing the web application, such as custom logos, email templates, `web.xml` updates, and redirector files, should be identical on every server in the environment.

Third-Party Components

High Availability environments must be configured to avoid any single point of failure. As MFT uses third-party components, the following components must have High Availability:

- Database: The database used by MFT is critical and must be installed on a database

cluster, such as SQL Server or Oracle RAC cluster.

- Email Server: MFT products use third-party email servers to send SMTP emails.
- LDAP: MFT can connect to various LDAP servers for authentication.
- JMS: MFT can send alerts or whole file payloads over JMS.

TIBCO MFT Command Center High Availability

The following MFT Command Center services require special considerations to enable High Availability (HA):

- Scheduler Service
- Status Server Service
- Collection Service
- JMS Service
- Platform Server Service

Configuring the Scheduler Service for High Availability

The Scheduler Service can execute in Active/Active High Availability. Multiple MFT Command Center instances can enable and start the Scheduler Service. Ensure that when the Scheduler Service is started on multiple Command Center instances, the instances use a time-sync service to synchronize their times. If times are not synchronized, jobs may execute multiple times. When multiple MFT Command Center instances have enabled the Scheduler Service, scheduler jobs can execute on any Command Center instance; you cannot direct a request to a specific instance.

Note: When a Scheduler job executes an Internet Server Transfer, the transfer routes to an Internet Server instance based on the server definition with Server type **Internet Server**. To support High Availability, set the host name of the Internet Server server definition to a Load Balancer IP name or IP address. This allows requests to route to any Internet Server instances configured by the Load Balancer.

Configuring the Status Server Service for High Availability

The Status Server Service can execute in Active/Passive High Availability. On enabling the Status Server Service, you can select multiple Command Center instances where the Status Service should execute. The Status Service executes on all enabled Command Center instances, but only one Command Center instance actively performs Status Service polling. The other instances remain in standby mode until the active Status Service stops running.

Configuring the Collection Service for High Availability

The Collection Service can execute in Active/Passive High Availability. On enabling the Collection Service, you can select multiple Command Center instances where the Collection Service should execute. The Collection Service executes on all enabled Command Center instances, but only one Command Center instance actively collects Platform Server transfers. The other instances remain in standby mode until the active Collection Server instance stops running.

Configuring JMS for High Availability

The JMS can execute in Active/Active High Availability. There are two ways that the JMS can be enabled for High Availability:

- You can enable the JMS Service on multiple Command Center instances. Each Command Center instance can read and process data from the JMS Queue. If one Command Center instance stops executing, the other instances continue reading and processing data from the JMS Queue.
- You can define the **JMS Server URL** to point to multiple JMS instances. Refer to the JMS Server documentation for the correct URL format to connect to different JMS Servers.

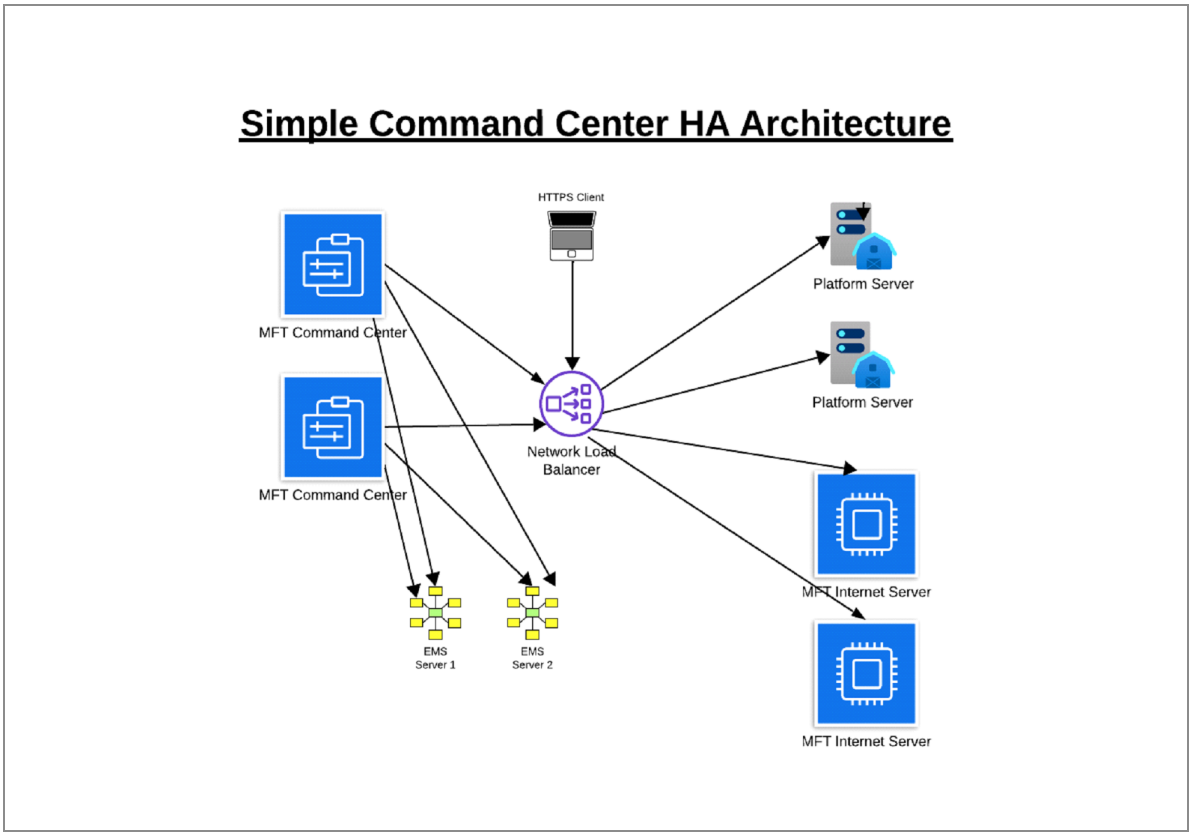
> **Note:** The JMS Server URL can be configured either globally or be overridden for specific Command Center or Internet Server instances.

Note: When a JMS job executes an Internet Server Transfer, the transfer routes to an Internet Server instance based on a server definition with a Server type of **Internet Server**. To support High Availability, set the host name of the **Internet Server** server definition to the Load Balancer IP name or IP address. This allows requests to route to any Internet Server instances configured by the Load Balancer.

Configuring the Platform Server Service for High Availability

The Platform Server Service can execute in Active/Active High Availability. The Platform Server Service allows you to execute JMS jobs from a Platform Server client. To implement High Availability, activate the Platform Server service on multiple Command Center instances. However, clients should connect to the Command Center Platform Server port through a Load Balancer.

Note: The Platform Server Service is not frequently used and generally does not require High Availability.



- Browsers connect to the Load Balancer to connect to either Internet Server or Command Center instances.
- All Command Center and Internet Servers must connect to the same database.
- Command Center connects to the Load Balancer to connect to an Internet Server instance when initiating a JMS or Scheduler transfer.
- Command Center connects to the Load Balancer to connect to a Platform Server instance when initiating transfers, collecting Audit records, or configuring Platform Servers.
- Command Center can be configured to connect to multiple JMS instances for writing data to a JMS Queue or Topic. This is also used when retrieving messages from JMS Queues. Either Command Center can retrieve JMS requests to perform Platform Server or Internet Server transfers.

Note: When multiple Command Center instances activate the MFT Scheduler, synchronize the Command Center instance date and time with a time server to prevent scheduled jobs from executing twice.

TIBCO MFT Internet Server High Availability

To configure the MFT Internet Server for High Availability, consider the following additional functions:

Local Server Definitions

When a server definition is configured to write to local disk storage, such as NAS or NFS, ensure that the local storage is accessible by all Internet Server instances. Otherwise, transfer clients may get different results when executing on different Internet Server instances.

Mailbox, FileShare, and Four Eyes Repository Servers

When enabling Mailbox, FileShare, or Four Eyes, ensure that the repository server is accessible by all Internet Server instances. The following server types are allowed for the Mailbox, FileShare, or Four Eyes repository:

- Local: Ensure that the storage is accessible by all Internet Server instances. For example, use NAS storage.
- Platform Server: Route all requests to a Platform Server, allowing all instances to connect to a single Platform Server to store Mailbox, FileShare, or Four Eyes files.

  > **Note:** You can connect to a Platform Server running in High Availability mode. See Platform Server High Availability for more information.

- Cloud Storage: Use Google Cloud Storage, Azure File, or Amazon S3 storage. Cloud storage is typically used in the cloud but can also be used on-premises.

The following MFT Internet Server services may require special considerations to enable High Availability:

- AS2 Service
- FTP/FTPS Service
- Platform Server Service
- SSH Service
- OFTP2 Service

Configure the AS2 Service for High Availability

The AS2 service uses the Tomcat server HTTP/HTTPS service for incoming connections. Configure the load balancer HTTPS settings to work for incoming AS2 transfers.

Configure the FTP/FTPS Service for High Availability

FTP and FTPS connections require two connections for listing and transferring files: Control connection and Data connection. Data connections must be made to the same Internet Server instance as the Control connection. Configure the firewall to support FTP/FTPS connections as both the FTP and FTP Services can support non-standard ports. On the **System Configuration** page, go to **FTP Settings** tab, configure the parameters `Limit Local Ports`, `Starting Port`, and `Number of Ports to Use` to define the ports used when Internet Server listens to a port for incoming data connections. Unless overridden on the individual Server instance, these values are used for all Internet Server instances. Configure the Load Balancer to pass data connections to the same Internet Server instance where the control connection was established.

Configure the Platform Server Service for High Availability

Platform Server uses a single data connection for each file transfer. In the event of a restart, the client can connect to a different Internet Server instance to restart the transfer. No special Load Balancer rules are required for incoming Platform Server connections.

Configure the SSH Service for High Availability

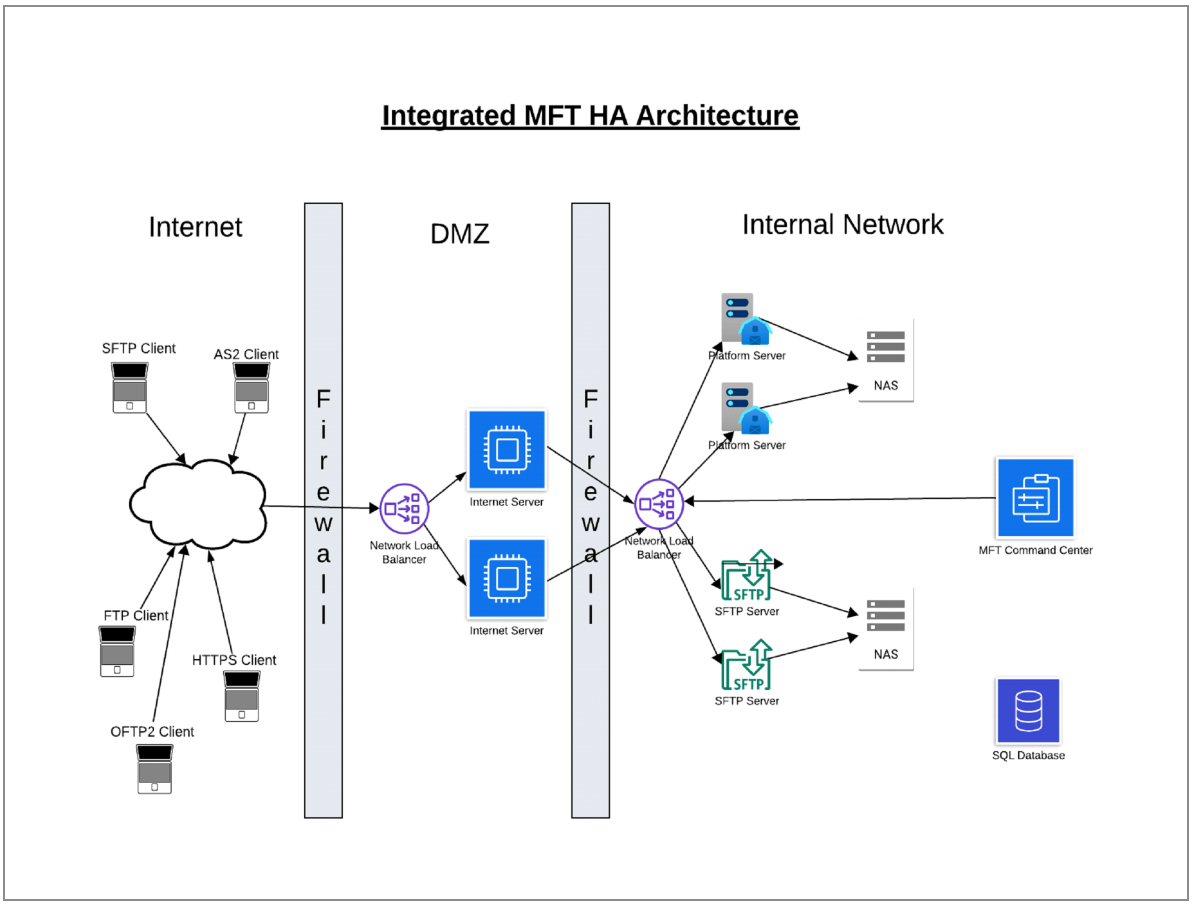
SSH uses a single data connection for each file transfer. In the event of a restart, the client can connect to a different Internet Server instance to restart the transfer. No special Load Balancer rules are required for incoming SSH connections.

Configure the OFTP2 Service for High Availability

OFTP2 uses a single data connection for each file transfer. In the event of a restart, the client can connect to a different Internet Server instance to restart the transfer. No special Load Balancer rules are required for incoming OFTP2 connections.

Integrated MFT Internet Server and Platform Server High Availability Diagram

The diagram below illustrates an integrated High Availability architecture for Internet Servers, Platform Servers, and SSH Servers.



- Clients (SFTP, HTTPS, FTP/FTPS, AS2, and OFTP2) connect to a DMZ Load Balancer through the internet.
- Two or more MFT Internet Server instances are installed in the DMZ.
- The Load Balancer periodically pings the Internet Server instances to verify they are active and directs transfer clients to an active Internet Server.
- The Internet Server authenticates and authorizes the user, then connects to a Load Balancer. The Load Balancer directs transfer clients to an active Platform Server or an SFTP Server.
- The file to be transferred is streamed to a Platform Server or an SFTP Server. The Internet Server can connect to a Load Balancer if multiple target Platform Servers or SFTP Servers are running in High Availability mode.
    - Platform Server: Two Platform Servers are configured for High Availability. The files are then stored on a NAS server.
    - SFTP Server: Multiple SFTP Linux instances can be configured to write data to a

NAS server.

Note: All Internet Server and Command Center instances require a network connection to the database. When using LDAP Authentication, all Internet Server and Command Center instances require a network connection to the LDAP Server.

Configuring Connection Manager for High Availability

Use Connection Manager with MFT Internet Server when the Internet Server runs in the DMZ. Many installations restrict opening TCP connections from the DMZ network to the internal network. Using the Connection Manager solves this and opens all connections from the internal network to the external network.

Note:
For detailed information about Connection Manager, see the following documentation:

- TIBCO® Managed File Transfer Internet Server User Guide
- TIBCO® Managed File Transfer Internet Server Managed File Transfer Overview
- TIBCO® Managed File Transfer Command Center User Guide
- TIBCO® Managed File Transfer Command Center Managed File Transfer Overview

There are two components of Connection Manager:

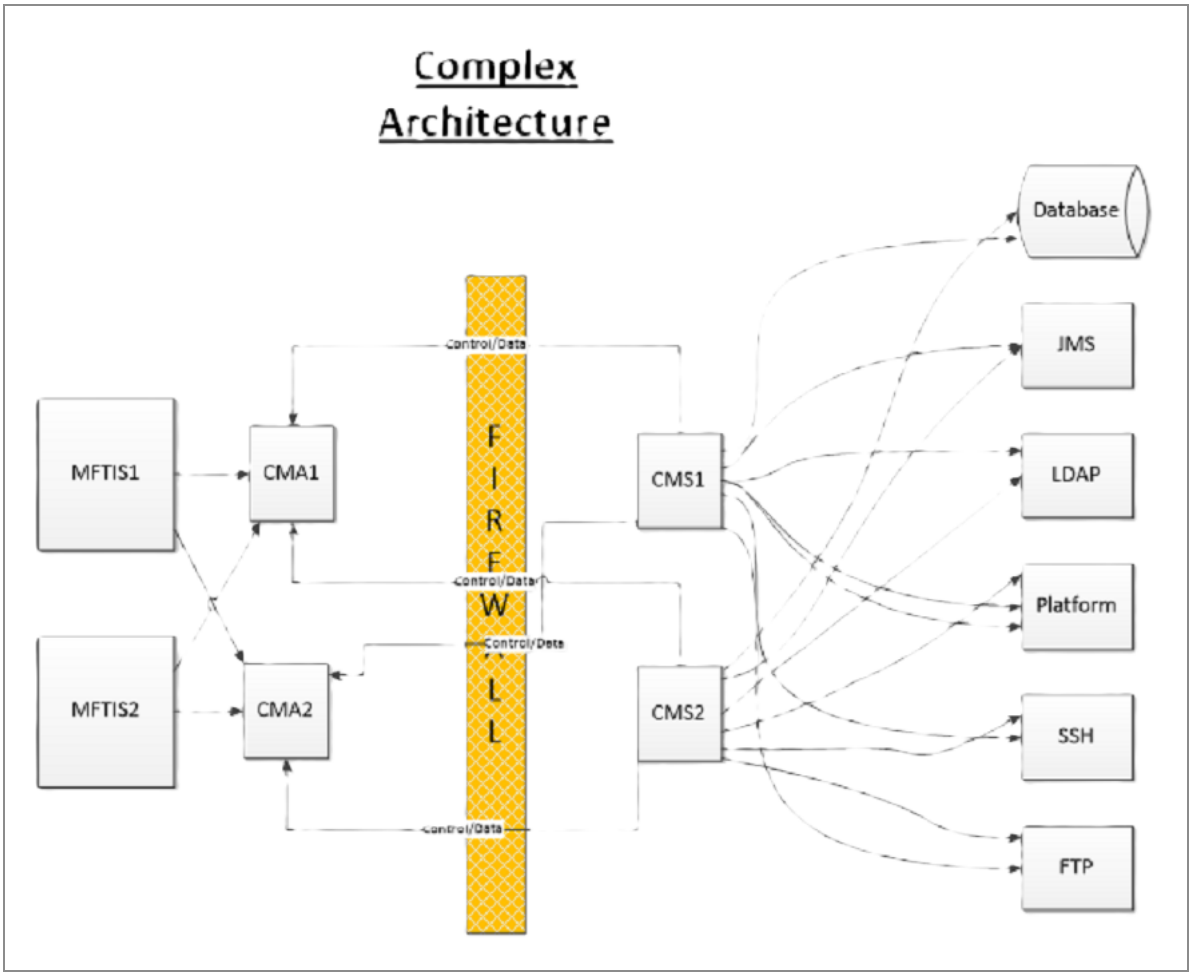
- Connection Manager Server: Runs in the internal network. Establishes control connections to the Connection Manager Agent. Accepts connection requests from the Agent over the control connection and creates TCP connections to the Connection Manager Agent. Connection Manager Server is distributed with MFT Command Center.
- Connection Manager Agent: Accepts connection requests from Internet Server, sends

connection requests over the control connection to the Connection Manager Server, and accepts connection requests from the Connection Manager Server. Connection Manager Agent is included with the MFT Internet Server.

Note: MFT Command Center is required to configure the Connection Manager Agent and Server.

Creating a Highly Available Connection Manager Architecture

The following figure shows a sample Connection Manager architecture containing multiple Connection Manager Agents (CMA) and multiple Connection Manager Servers (CMS). As all connections between the CMA and CMS are initiated by the CMS to the CMA, do not place a load balancer between the CMA and CMS.



To configure high availability, adhere to the following rules:

- Create two or more CMS instances in the internal network, executing on different computers.
- Create two or more CMA instances in the DMZ, executing on different computers.
- Create two or more Internet Server instances in the DMZ, executing on different computers.

> **Note:** CMA and Internet Server can execute on the same or different computers in the DMZ. CMS and Command Center can execute on the same or different computers in the internal network.

- Configure CMS1 and CMS2 to connect to CMA1 and CMA2.
- Configure CMA1 and CMA2 to accept connections from CMS1 and CMS2.
- Configure CMA1 and CMA2 to accept connection requests from MFT Internet Server 1 and MFT Internet Server 2.
- Configure MFT Internet Server 1 and MFT Internet Server 2 to connect to CMA1 and CMA2.

Connection Manager operates in active or passive mode. Requests are sent to the first available component. If the connection to that component fails or is unavailable, the Connection Manager attempts to send the request to the next component.

High Availability for Platform Server for UNIX

Platform Server for UNIX version 8.1.0 and above supports an integrated High Availability feature. When you install multiple Platform Servers for UNIX instances behind a load balancer,

- Multiple Platform Servers appear to the Platform Server client as a single Platform Server.
- All transfers use a single audit log.
- All transfers use a single transaction number file, ensuring unique transaction IDs across all Platform Servers in the High Availability cluster.
- Configuration files (Nodes, Profiles, and Responder Profiles) can be shared among Platform Server instances by placing these files in a common directory.

- The `config.txt` file must be in the `$CFROOT/config` directory and is typically not shared. If the `config.txt` file is the same for all Platform Server instances, move the `config.txt` file to a common location and create a soft link to the `config.txt` file in the `$CFROOT/config` directory. Any changes to the `config.txt` file, such as tracing, applies to all Platform Server instances.
- Command Center can configure Platform Servers in the High Availability Cluster. If the Platform Servers share configuration files, Command Center can configure Platform Server Nodes, Profiles, and Responder Profiles for all Platform Servers in the High Availability Cluster.
- Command Center can collect all Platform Transfer audit records. Individual Collection requests can be processed on any instance in the Platform Servers in the High Availability Cluster.
- The MFT BW plug-in **Wait for Platform Transfer Completion** can execute on any instance in the Platform Servers in the High Availability Cluster.
- You can have more than two Platform Server instances in the High Availability cluster.

  Note: Platform Server responder (server) requests work in a High Availability Active/Active mode. Initiator (client) requests are initiated by commands on individual UNIX and Linux servers and are not highly available. However, they share `Log.txt` (audit), config, and transaction number files.

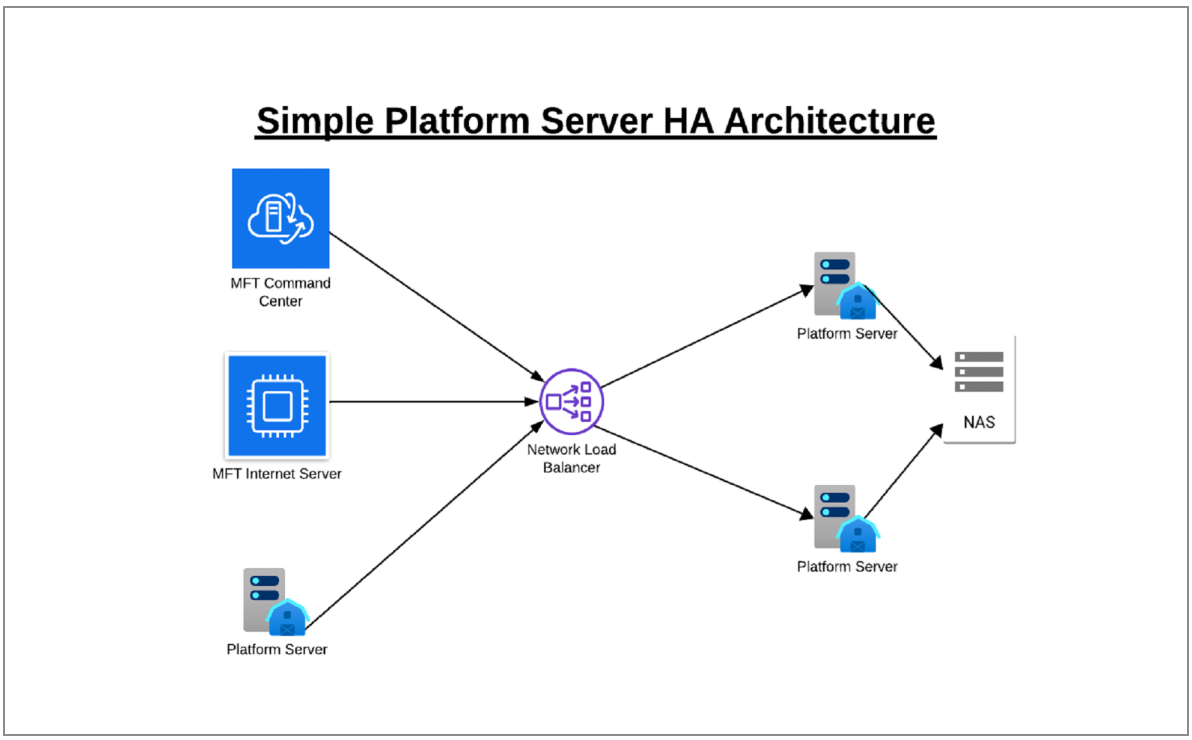
Note: Multiple Platform Servers for UNIX instances can be in a High Availability Cluster. However, Platform Server for UNIX and Platform Server for Windows cannot be in the same High Availability cluster.

HA can be used under the following circumstances:

- When critical files must be transferred and 100% uptime is required.
- For transfers from other Platform Servers.
- For transfers from external clients through Internet Server.
- As a repository for files being transferred by the Internet Server Mailbox, FileShare and Four Eyes features.

Diagram of a Simple Platform Server for UNIX HA Environment

The following image illustrates a simple High Availability architecture for Platform Server for UNIX.



- Two or more Platform Servers are installed in HA mode behind a Load Balancer. To Clients, these Platform Servers appear as a single Platform Server.
- MFT Command Center configurations and Audit requests route to any Platform Server behind the Load Balancer.
- Platform Server initiators (clients) can be any system that supports the Platform Server protocol. File Transfer Clients are typically MFT Internet Server or Platform Servers (Windows, UNIX, z/OS, or IBM i).
- Platform Servers running in HA mode must access common storage, typically NAS, or NFS storage.
- File transfer files are written to the common storage so that transfer restart requests can be processed by any machine in the HA cluster.
- Common log, transaction number, and restart files are shared among all machines in

the HA cluster.

- Common configuration files (Nodes, Profiles, and Responder Profiles) can be shared among all machines in the HA cluster.

Load Balancer Requirements for Platform Server for UNIX

Connection Awareness

- Platform Server uses a single connection for each file transfer or administration request. Once a TCP connection is established, the Load Balancer must direct all TCP requests for that connection to the same target server.
- Platform Server is a proprietary server. When the Load Balancer **pings** the Platform Server to verify it is active, the Load Balancer should not send any data packets. The Load Balancer should only Open and Close a TCP connection to the Platform Server without sending data packets.

Source NATTING

- When a client connects to a target Platform Source through a Load Balancer, the IP address of the incoming request is the IP address of the Load Balancer, not the IP address of the initiating Platform Server. As a result, the Target Platform Server uses the same Node definition for all incoming requests. This could affect how responder profiles are configured and used.
- The Platform Server protocol is a single-connection TCP-based protocol and can be supported by most load balancers.

Installing Platform Server for UNIX in a High Availability Environment

For information on installing Platform Server for UNIX in a high availability (HA) environment, see the *TIBCO® Managed File Transfer Platform Server for UNIX Installation* Guide.

High Availability Requirements

- Multiple Platform Servers for UNIX instances are installed and configured for High Availability.
- A common storage server, such as a NAS server, is accessible from all Platform Server for UNIX instances in the HA cluster. This storage is used for the following purposes:
  - Saving Configuration files (node, profile, and responder profile)
  - Saving Audit records (`Log.txt` files)
  - Saving the Transaction Number file to ensure that all Transaction Numbers are unique
  - Reading and writing files for file transfers
- A load balancer is installed for incoming and possibly outgoing Platform Server requests.
- The load balancer is configured to ping the Platform Servers to verify they are active. The **Ping** request should establish a TCP connection without sending TCP data.
- Network connectivity between the Platform Server instances and the load balancer.
- Network connectivity between each Platform Server instance and the CyberMgr Daemons on the primary and secondary Platform Server instances.

Architecture for Platform Server for UNIX

A new Daemon CyberMgr, facilitates High Availability for Platform Server for UNIX. CyberMgr runs on each Platform Server instance and provides the following capabilities:

- Writing audit records (`Log.txt` records).
- Retrieving and updating the Transaction number file.
- Writing message and Audit log files.
- Retrieving Active Transfers across different Platform Server instances.

Note: The CyberMgr daemon must be running for successful transfers. If a transfer is executing and CyberMgr is not active, the transfer fails with an error.

When running in HA mode, configure one Platform Server CyberMgr as the Primary CyberMgr and another as the Secondary CyberMgr. Each Platform Server instance must start a CyberMgr daemon, even if it is not the primary or secondary instance.

The local CyberMgr on each Platform Server instance is used for:

- Writing message and Audit log files

The primary, and possibly the secondary, CyberMgr is responsible for:

- Writing audit records (`Log.txt` records)
- Retrieving and updating the Transaction number file.
- Retrieving Active Transfers across different Platform Server instances.

When a transfer process needs to call CyberMgr to process a request (for example, write an Audit Record), the transfer process connects to the Primary CyberMgr. If the connection to the Primary CyberMgr is successful, the Primary CyberMgr processes the request. Otherwise, when the connection fails, the process connects to the Secondary CyberMgr to process the request. If connections to both the Primary and Secondary CyberMgr daemons fail, the transfer fails.

Installing and Configuring Platform Server for UNIX for High Availability

There are two ways to install or configure Platform Server for UNIX High Availability:

1. During the product installation.
2. Using the **hainstall** script in `$CFROOT/bin`.

These two methods are discussed in the following sections.

Enable High Availability During Platform Server for UNIX Installation

See the *TIBCO MFT Platform Server for UNIX Installation* Guide for detailed instructions. There are two ways to install Platform Server for UNIX.

Note: If you are reinstalling Platform Server for UNIX, issue the `cfstop` command to stop CyberMgr and all CyberResp processes before starting the installation.

1. Console Installation
2. Silent Installation

Console Installation

Follow the installation instructions. When prompted with the message `Would you like to configure High Availability mode now? (Y/N)?`, reply `Y` or `y`.

You are prompted for the following information:

- Location of the **Shared HA Folder** accessible by all Platform Server instances in the HA cluster
- Host name (IP Name or IP Address) and port of the Primary RPC Service
- Host name (IP Name or IP Address) and port of the Secondary RPC Service
- Whether common Config files are used. If common config files are used:
    - Define the directory where the common config files are located. This directory must be in the **Shared HA Folder** defined earlier
    - Indicate if you want to copy the config files to this directory
    - If you want to copy the source files, specify the location of the source config files
    - Indicate whether to replace or create the config files in the target folder

Silent Installation

Update the `silent.cfg` file used during silent installation with the following parameters:

Parameter	Value	Description
HAMode	No	HA Yes or No
HASharedDirectory	/Shared/HADirectory	HA Shared Directory

Parameter	Value	Description
HAPrimaryRPCIPName	primary.mftrpc.ip	HA Primary RPC IP Name
HASecondaryRPCIPName	secondary.mftrpc.ip	HA Secondary RPC IP Name
HAPrimaryRPCPort	46678	HA Default Primary RPC Port
HASecondaryRPCPort	46678	HA Default Secondary RPC Port
HAUseMyConfigFiles	Create	HA Create or Replace or No

Note: When HAMode is enabled, the `ConfigDirectory` parameter in the `config.txt` file is set to `HASharedDirectory/config`.

Enable High Availability Using the HAinstall Command

To get help information for the `hainstall` parameters, run the following command:

```
./hainstall.sh -h
```

You can execute this command with all parameters on the command line or be prompted for each parameter.

The following command is a sample `hainstall.sh` command:

```
./hainstall.sh -dshared /commonfolder/PSU -hprimary server1.acme.com -
hsecondary server2.acme.com -cfiles c
```

This command sets the following values.

Parameter	Value
Common HA Folder	/commonfolder/PSU
Primary HA Server	server1.acme.com
Secondary HA Server	server2.acme.com

Note: Copy but do not replace the `config` files. The config files are copied from the `$CFROOT/config` directory to the directory defined by the `dshared` parameter.

High Availability for Platform Server for Windows

Platform Server for Windows version 8.1.0 and above supports an integrated High Availability (HA) feature. On installing multiple Platform Servers for Windows instances behind a load balancer,

- Multiple Platform Servers appear to Platform Server clients as a single Platform Server.
- All transfers use a single audit log.
- All transfers use a single Transaction Number file, ensuring unique Transaction IDs across all Platform Servers in the HA cluster.
- Configuration files (Nodes, Profiles, and Responder Profiles) can be shared among Platform Server instances by placing these files in a common directory.
- Configuration data for Platform Server for Windows is stored in three places:
    - Windows Registry: Specific to each Platform Server instance.
    - PQF File: Shared across multiple Platform Server instances when running in HA mode.
    - Shared configuration files (nodes, profiles, and responder profiles).
- Command Center can configure Platform Servers in the HA cluster. If the Platform Servers in the HA Cluster share configuration files, Command Center can configure Platform Server Nodes, Profiles, and Responder Profiles for all Platform Servers in the HA cluster.
- Command Center can collect all Platform Transfer audit records. Individual Collection requests can be processed on any instance in the Platform Servers in the HA cluster.
- Execute the MFT BW plug-in **Wait for Platform Transfer Completion** on any instance

in the Platform Servers in the HA cluster.

- Support more than two Platform Server instances in the HA cluster.

Note: Multiple Platform Servers for Windows instances can be in an HA cluster. However, Platform Server for Windows and Platform Server for UNIX cannot be in the same HA cluster.

HA Mode for Initiator and Responder Requests

Responder Requests

Platform Server responder (server) requests work in an HA Active/Active mode. Transfers route to any Platform Server for Windows instances behind the load balancer.

Initiator Requests

The Platform Server Initiator requests can be initiated in two ways:

- Executed through the `ftmscmd`, `cfsend`, or `cfrecv` commands and not submitted to the Platform Server service.

  For example,

  - Executing `ftmscmd`, `cfsend`, or `cfrecv` commands through the Windows command prompt without specifying the submitted server parameter.
  - Executing pDNI transfers on a Windows instance.

  These transfers are not executed in HA mode and run on the machine where the command is executed.

- Transfers submitted to the Platform Server Windows Service.

  For example,

  - Transfers initiated by `ftmscmd`, `cfsend`, or `cfrecv` commands through the Windows command prompt that specify the submitted server parameter.
  - Transfers initiated by the Platform Server Administrator GUI application.
  - Transfers initiated by Platform Server for Windows DNI processing.

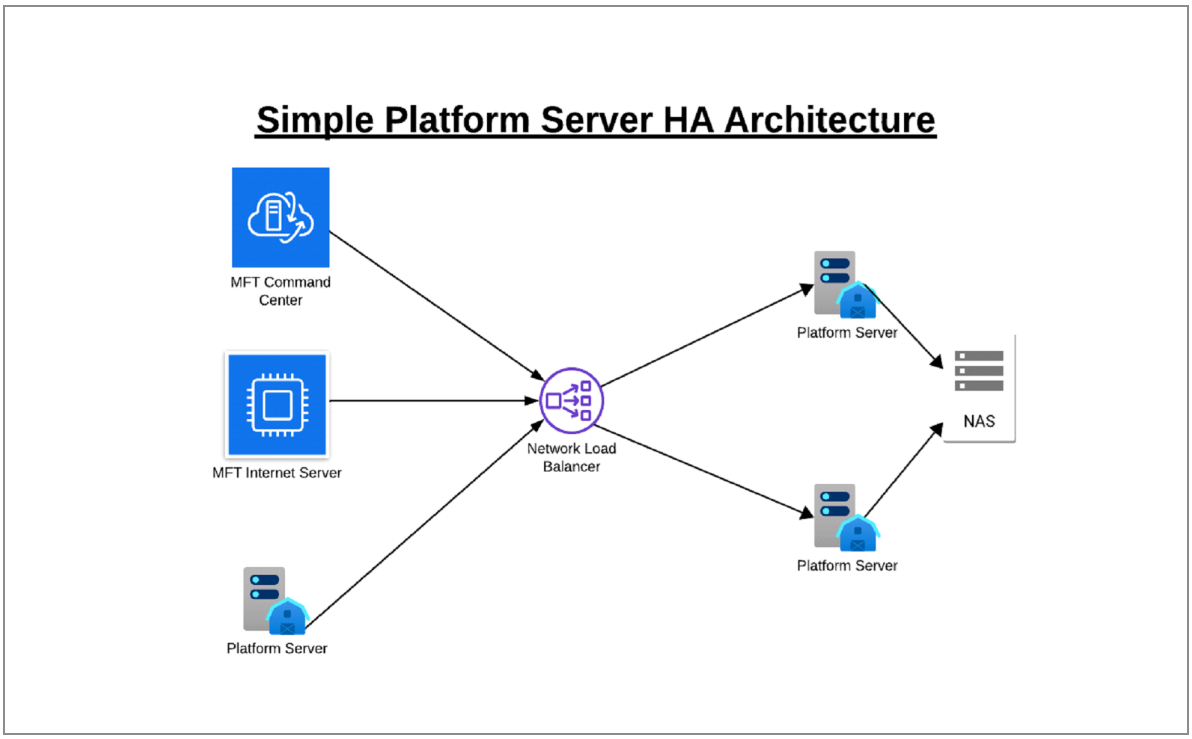
  These transfers are executed in HA mode.

HA can be used under the following circumstances:

- When critical files need to be transferred and 100% uptime is required.
    - Transfers from other Platform Servers.
    - Transfers from external clients through Internet Server.
- As a repository for files being transferred by the Internet Server Mailbox, FileShare and Four Eyes features.

Diagram of a Simple Platform Server for Windows HA Environment

The following image illustrates a simple High Availability architecture for Platform Server for Windows.



- Two or more Platform Servers are installed in HA mode behind a Load Balancer. To Clients, these Platform Servers appear as a single Platform Server.
- MFT Command Center configurations and Audit requests route to any Platform Server behind the load balancer.
- Platform Server initiators (Clients) can be any system that supports the Platform

Server protocol. File Transfer Clients are typically MFT Internet Server or Platform Servers (Windows, UNIX, or z/OS).

- Platform Servers running in HA mode must access common storage, typically NAS, or UNC storage.
    - Platform Servers should write file transfer files to the common storage so that transfer restart requests can be processed by any machine in the HA cluster.
    - Platform Servers should share common Log, Transaction number, and PQF files among all machines in the HA cluster.
    - Platform Servers should share common configuration files (Nodes, Profiles, and Responder Profiles) among all machines in the HA cluster.
    - DNI Templates and Initiation Directories must access the scan directories and transfer files.

Load Balancer Requirements for Platform Server for Windows

Connection Awareness

Platform Server uses a single connection for each file transfer or administration request. Once a TCP connection is established, the Load Balancer must direct all TCP requests for that connection to the same target Server.

Platform Server is a proprietary server. When the Load Balancer **pings** the Platform Server to verify if it is active, the Load Balancer should not send any data packets. The Load Balancer should only open and close a TCP connection to the Platform Server without sending data packets.

Source NATTING

When a client connects to a target Platform Source through a Load Balancer, the IP address of the incoming request is the IP address of the Load Balancer, and not the IP address of the initiating Platform Server. As a result, the Target Platform Server uses the same Node definition for all incoming requests. This could affect how the responder profiles are configured and used.

The Platform Server protocol is a single-connection TCP-based protocol. It can be supported by most Load Balancers.

Installing Platform Server for Windows in a High Availability Environment

To enable Platform Server for Windows High Availability (HA) mode, see the "Appendix A: High Availability" section in the *TIBCO® Managed File Transfer Platform Server for Windows User Guide*. This document provides a brief summary of the steps needed to enable and disable HA.

Enabling High Availability Mode

To convert to HA mode, perform the following steps:

1. Stop all active Platform Server for Windows applications running on the machine.
2. Stop the **TIBCO MFT Platform Server** service.
3. Open a new command prompt as an administrator and run the following command:

   ```
   hasetup.exe install "\\shared.mycompany.com\mftps\haRoot"
   ```

4. You are prompted for optional HA parameters. For information on HA parameters, see the *TIBCO® Managed File Transfer Platform Server for Windows User Guide*.
5. After running the `hasetup.exe` app and adding a server to the HA cluster, restart the **TIBCO MFT Platform Server** service.
6. In the MFT Platform Server Administrator app, a new entry on the Server Property page, **HA Setup**, with the `haRoot` folder path is visible. This confirms that the server is running in HA mode. The first server started in HA mode becomes the Primary HA Server.

Disabling High Availability Mode

To convert to Normal mode from HA mode, complete the following steps:

1. Stop all active Platform Server for Windows applications running on the machine.

2. Stop the **TIBCO MFT Platform Server** service.
3. Open a new command prompt as an administrator and run the following command:

```
hasetup.exe uninstall "\\shared.mycompany.com\mftps\haRoot"
```

4. You are prompted for an optional HA. For information on HA parameters, see the *TIBCO® Managed File Transfer Platform Server for Windows User Guide*.
5. After running the `hasetup.exe` app and removing a server from the HA cluster, restart the **TIBCO MFT Platform Server** service.
6. In the MFT Platform Server Administrator app, you can no longer see the **HA Setup** entry on the **Server Property** page. This confirms that the server is not running in HA mode.

High Availability for pDNI

pDNI is the event-driven component of MFT that allows you to detect files within a directory and transfer each file to a target MFT Platform Server or MFT Internet Server. MFT Command Center can manage and configure pDNI, or it can be managed and configured directly on the instance where pDNI is installed. pDNI is supported on Windows, Linux, z/Linux, and UNIX servers.

pDNI supports High Availability (HA) in an Active/Passive mode. pDNI uses a lock file to determine if a DNI Template is active on another system. If another system is using the DNI template, you must wait for a scan interval before DNI checks the lock again. The default time to check for an active system is 60 seconds (configurable). If the other system is inactive for 60 seconds, a lock file is created and it becomes the active DNI. If the other system starts, DNI detects that this system is active and turns into the passive system.

The pDNI script is self-documenting. To get more information about pDNI, enter the following command in the directory where pDNI is installed:

```
perl dni help ha
```

To get more information about the parameters that define pDNI High Availability, enter the following command in the directory where pDNI is installed:

```
perl dni help template
```

Press enter until you see the HA (High Availability) parameters.

pDNI HA is enabled on a template-by-template basis. You can enable pDNI HA on one template.

Restrictions on Using pDNI HA

Following are the restrictions on using pDNI HA are provided below.

- Post Action Leave is not supported.
- Both Active and Passive systems require read, write, and control access to the directory defined by the `HADirectory` parameter.
- Both Active and Passive systems require read, write, and control access to the directory where the files are read (DNI Send) or written (DNI Receive).
- The best practice if for both the systems to use the same template.

pDNI High Availability Template Parameters

The following table describes the pDNI template parameters associated with high availability (HA).

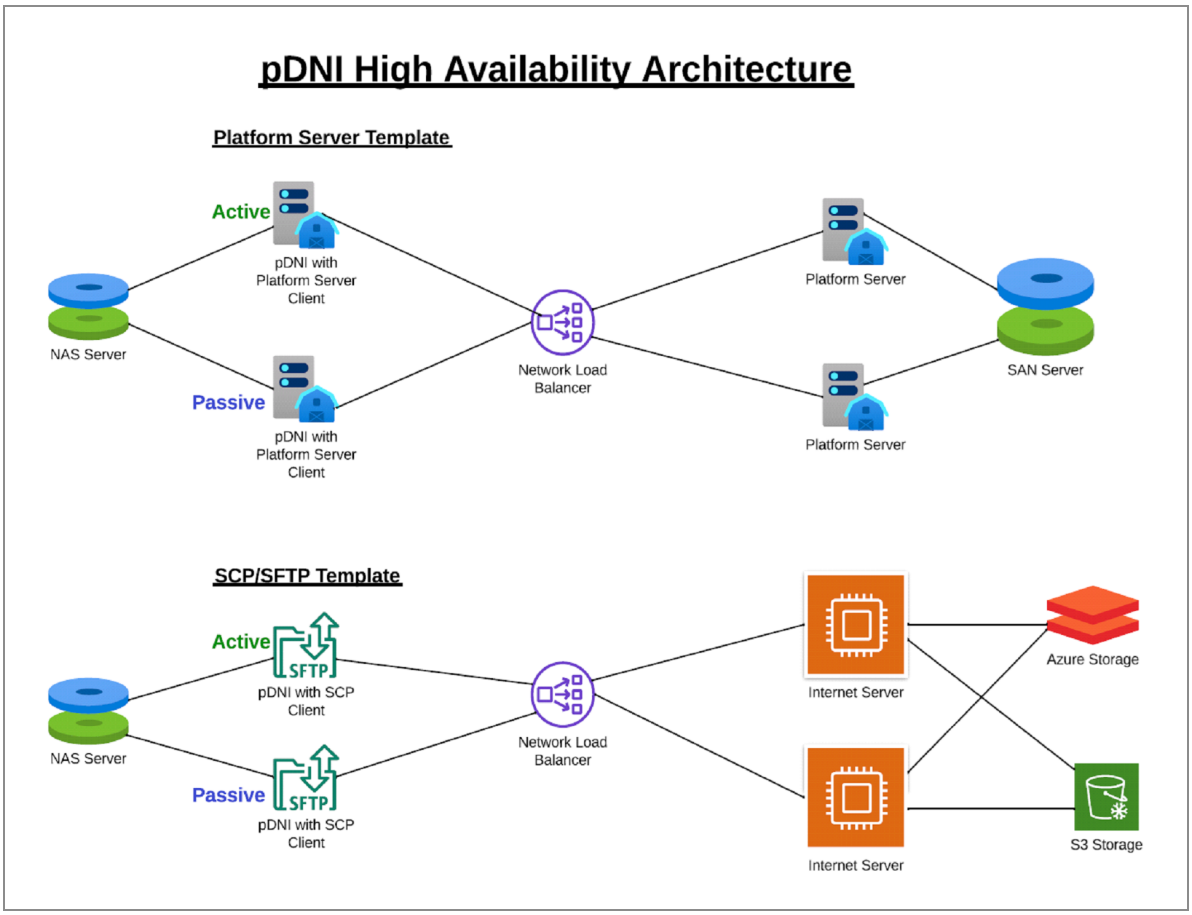
Parameter Name	Parameter Description
HADirectory	Defines the directory to which both the Active and Passive systems have full read, write, and control access. pDNI uses this directory to read, write, and delete lock files. pDNI uses this directory for two purposes: 1. Detect the pDNI active system that locked this template. 2. Create a lock on the pDNI template. **Note:** If this parameter is not defined or is commented out, HA is disabled for this template.
HADebug	Defines whether HA debugging is enabled. There are two valid values:

Parameter Name	Parameter Description
	• Yes (HA debugging is turned on) • No (HA debugging is turned off) Unless directed to enable debugging by the TIBCO Technical Support, set this parameter to **No**.
HAScanInterval	Defines how many seconds pDNI waits between lock file scans to detect if the template is locked. Valid values are 1 to 30 seconds. The default value is 10 seconds.
HAScanTime	Defines how many seconds pDNI waits before determining that a template is not active. If the contents of the lock file have not changed in this interval, this pDNI instance creates its own lock file and shifts from **Passive** to **Active** state. Valid values are 10 to 120 seconds. The default value is 60 seconds.
HAUpdateInterval	Defines how many seconds pDNI waits between lock file updates when running in active mode. Valid values are 1 to 20 seconds. The default value is 5 seconds.

Diagram of pDNI High Availability

The following image shows two the templates that use pDNI Active/Passive High Availability:

1. Platform Server Template
2. SCP/SFTP Template



Common features

- Both templates are configured for pDNI HA.
- Both templates have been started. The first template started becomes the active template. The second template started becomes the passive template.
- Both templates have NAS (or other common storage) with read and write access.
- Both templates point to a Load Balancer that redirects them to HA Platform Servers or HA Internet Server.

Platform Server Template

- Connects to the Load Balancer that redirects them to another Platform Server.
- Required when you want to download files from a partner Platform Server or Internet Server.
- Can connect to Internet Server to upload or download files from Internet Server.

SCP/SFTP Template

- Connects to the Load Balancer that redirects them to an Internet Server.
- Can be used when a Platform Server is not installed on this machine.
- Can upload files to Internet Server, but cannot download files from Internet Server.

MFT Disaster Recovery

This section describes various Disaster Recovery (DR) scenarios for TIBCO MFT Command Center and TIBCO MFT Internet Server. It does not cover DR for TIBCO MFT Platform Servers and pDNI event-driven processing, as these are typically integrated with applications. These Disaster Recovery scenarios should be considered when designing application Disaster Recovery.

The following Disaster Recovery topics are covered in this section:

- Network and IP Address considerations for Disaster Recovery
- Database considerations for Disaster Recovery
- Internet Server and Command Center instances at the Disaster Recovery site
- Other applications at the Disaster Recovery site
- Disaster Recovery sites:
    - At a different location
    - On the cloud
- Types of Disaster Recovery:
    - Always Active
    - Passive: Hot standby
    - Passive: Started on demand

Requirements

Ensure that MFT Internet Server and Command Center instances at the DR site are at the same release and hotfix level as the servers at the primary site. While MFT Internet Server and Command Center are compatible with older versions, you may lose fixes or be vulnerable to Common Vulnerabilities and Exposures (CVEs) resolved in the current MFT releases.

Disclaimer

The Disaster Recovery (DR) design for every company is different. There is no one solution to handle DR. This document describes different types of DR and the steps needed to implement different DR solutions. Use it as a guide when designing a File Transfer DR Recovery plan.

Disaster Recovery Mode: Always Active

In this mode, MFT Internet Servers and Command Center instances at the DR site run simultaneously with the instances at the primary site. Routers and Load Balancers can forward traffic to the on-premises site under normal circumstances. If the primary site is unreachable, traffic routes to the DR site. Optionally, traffic can be routed to the Primary and DR sites at all times. Transfers can execute on either the primary (on-premises) servers or the DR servers.

This mode requires Transaction Replication between the database servers. For more information, see Database Considerations.

Disaster Recovery Mode: Passive: Hot Standby

In this mode, MFT Internet Servers and Command Center instances at the DR site run simultaneously with the instances at the primary site. Routers and Load Balancers forward traffic to the on-premises site under normal circumstances. If the primary site is unreachable, traffic routes to the DR site. When using Hot standby mode, transfers cannot run simultaneously on both the primary and DR sites. Traffic should be routed to one site or the other, but not both.

This mode requires Mirroring or Transaction Replication between the database servers. For more information, see Database Considerations.

Disaster Recovery Mode: Passive: Started on Demand

In this mode, MFT Internet Servers and Command Center instances at the DR site do not run until the system is in DR mode. MFT Internet Servers and Command Center instances are installed but not started. When a disaster occurs and MFT must be moved to the DR site, the following must occur:

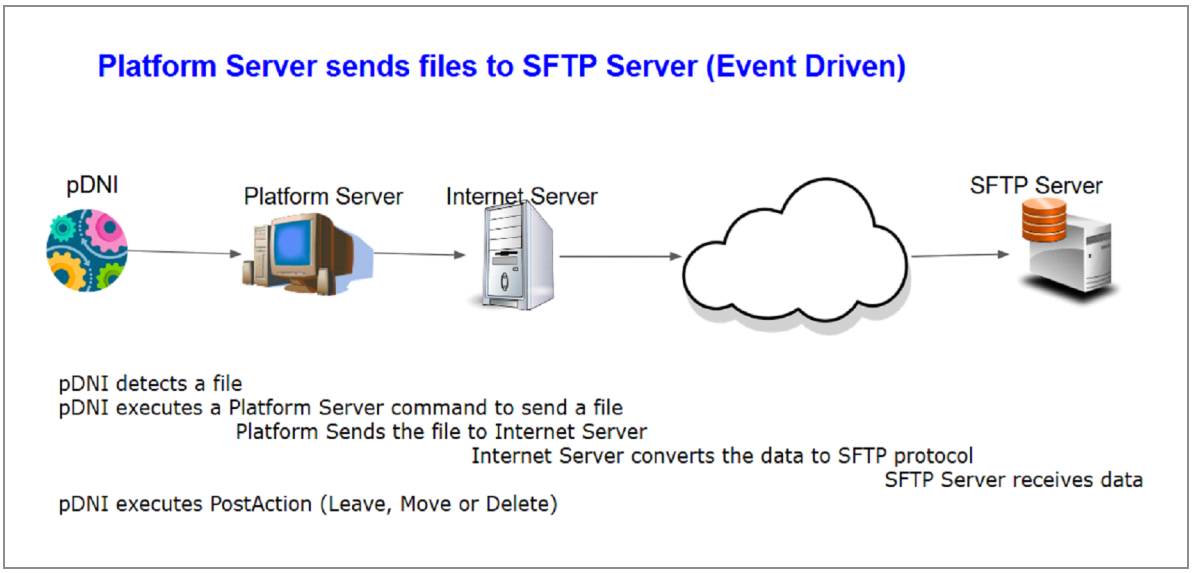
- The database backup must be restored to the DR database.
- Internet Servers and Command Center instances must be started.

- Load Balancers and DNS Servers must route traffic to the DR sites.

This mode typically uses the Backup/Restore option to copy the database from the primary site to the DR site. For more information, see Database Considerations.

How the MFT Architecture Simplifies Disaster Recovery Procedures

MFT acts as a protocol converter between a client and a server. MFT accepts data from a Transfer Client, converts the data from the Transfer Client protocol to the Transfer Server protocol, and streams the data to the Transfer Server without staging it. The following image shows how MFT event-driven processing can send data to a target SFTP server.



- MFT pDNI Event-Driven Processing detects a file that needs to be transferred.
- MFT pDNI issues an MFT Platform Server command to send the file to MFT Internet Server.
- MFT Internet Server converts the Platform Server protocol to the SFTP protocol.
- MFT Internet Server sends the data to the target SFTP server.
- Data is never staged. It is sent directly to the target server.
- PGP data is also streamed without staging. File data is encrypted/decrypted as it is being transmitted.

As long as MFT has network connectivity to the target servers, transfers running in the DR site should run as effectively as transfers running in the primary site.

- Transfers with Partners: Since the partner IP Address and IP Name does not change, DR-initiated transfers continue to run without changes. Ensure that the partner firewall accepts TCP connections from the DR site IP address. The best practice is to allow connections from both the primary site and DR site when their firewall is first configured to accept connections.
- Internal Transfers: If the DNS name of the target servers does not change, DR-initiated transfers continue to run without changes. If the DNS name changes, update the server definitions to point to the correct DR IP name. Properly configured DR networks should ensure seamless network connections to DR servers. This assumes that DNS Servers, Load Balancers, and firewalls are configured correctly.

Network/IP Address Considerations

As the Disaster Recovery (DR) site has different IP addresses and possibly different IP names than the primary site, ensure that the connectivity is to the expected system. The following network and IP address issues must be taken care of:

- Server Definitions for Partners
    - External (Partner) IP Addresses.
        - Typically do not need to change.
    - Internal IP Addresses (for routine data to internal machines using protocols, such as Platform Server, SFTP, and FTP).
        - Do these sites have different IP Addresses and IP Names? Do they share DNS Name and IP Address?
        - Are firewalls configured to accept requests from the DR site IP Address?
- Server definitions for Cloud Storage
    - Typically do not need to change.
- Server definitions for Internet Transfer Jobs
    - Internet Transfer jobs are typically routed to one or more Internet Server instances through a DNS Name.
- Partner Access to MFT Internet Server
    - Typically done through a DNS Name.

- Access to LDAP servers
    - If the LDAP servers are also in DR mode, the IP Name and IP Address might change.
- User Definitions
    - If Users are restricted by the subnet, user definitions might need to change.
- Transfer Definitions
    - If Transfer definitions are restricted by subnet, transfer definitions might need to change.
- System Configuration Server instances
    - Each installed Internet Server instance can have a configuration entry with the IP Address for that instance.
    - Dynamically created instances, when managed by Kubernetes, typically do not need to be updated.
- Command Center and Internet Server instances
    - If a bind IP Address is defined, you may need to update the service Bind IP Address in the `server.xml` file.
- Connection Manager Nodes
    - Connection Manager Agents
        - Define subnets where Command Center connection requests are accepted.
        - Define subnets where Internet Server connection requests are accepted.
        - Define subnets where Connection Manager Agent connection requests are accepted.
- Connection Manager Servers
    - Define subnets where Command Center connection requests are accepted.
    - Define subnets where Internet Server connection requests are accepted.
    - Define the IP Address of Connection Manager Agents.
- System Configuration Lockout Rules

    - Can contain exclusion IP addresses.
- SMTP Servers
    - Typically defined as an IP Name.
    - May need to be configured for the DR site.
- Antivirus and Data Loss Prevention Servers
    - Must be configured to DR AV/DLP servers.
- Webhooks
    - Webhooks to local servers may need to be changed.
- Platform Server Management
    - Node definitions may need to be configured for Command Center instances.
- pDNI Management
- The pDNI configuration file (`DNIConfig.cfg`) node definitions may need to be configured to accept TCP Connections from Command Center instances.
- `web.xml` IP Address
    - `LoadBalancerIPAddressList`

Keep records of any changes made to IP addresses, as these changes need to be restored when moving from the DR site to the primary site.

Note: It is a good practice to use DNS names whenever possible. While IP addresses can change, IP Names typically do not change. It's much easier for DNS Servers to modify DNS Name resolution than changing IP Names and IP Addresses in application configurations.

Database Considerations

Configuring the database is a critical aspect of Disaster Recovery (DR). All MFT Internet Server and Command Center configuration information is stored in the database. This includes user, server, keys, transfer, alert, and other configurations. It also includes a configuration record for each Internet Server and Command Center instance connected to the same database that has been installed or started.

There are three ways to ensure that the DR site includes an updated database:

1. Backup/Restore the Database
2. Mirror the Database
3. Transactional Replication

Each method has advantages and disadvantages.

Backup/Restore the Database

At predefined intervals, the database is backed up and transmits it to the DR site. The interval can be daily or, at most, weekly. If the DR site is running in hot-standby mode, restore the database with the backup copy. If the DR site is running in standby or on-demand mode, prepare the DB server with the database backup copy.

> **Note:** When running in DR mode, create procedures to back up the DR database so that it can be moved back to the primary site when the Disaster Recovery ends.

Advantages:

- Simplest and most cost-effective approach.
- Additional DB mirroring/replication software is not required.

Disadvantages:

- Procedures must be set up to back up the database daily or weekly and transfer the DB backup file to the DR site
- DB updates (including configuration, usage, and audit records) that occur after the last backup is lost.
- You may lose the Internet Server and Command Center instance configurations that define connectivity to all the instances. You can set an environment variable to update the system configuration when an Internet Server and Command Center instance is started.

This approach is suggested when running in **Passive: Started on demand** mode.

Mirror the Database

For MFT DR, database mirroring means that all updates made to the primary database are also made to the backup database. However, MFT can only connect to the primary database. It cannot connect to the backup database unless MFT is running on the DR site.

Advantages:

- The database is always up to date when moving from the primary site to the backup site.
- No procedures are required to back up and restore the database.
- No impact on MFT performance.
- Network latency is less of an issue than with Transaction replication.

Disadvantages:

- Requires a continuous network connection from the primary site to the DR site to mirror the database.
- May require a database software upgrade to a version that supports mirroring.

This approach is suggested when running in **Passive Standby** mode.

Transactional Replication

Transactional Replication means that the primary site and DR site are part of the same database cluster. MFT can be executed simultaneously on both the primary and DR sites. Updates to the primary site are replicated on the DR site, and updates to the DR site are replicated on the primary site.

Advantages

- The database is always up to date when moving from the primary site to the backup site.
- No procedures are required to back up and restore the database.

Disadvantages

- Requires a continuous network connection from the primary site to the DR site to mirror the database.
- Network connection between the primary and DR sites must have low latency,

preferably under 5 ms.

- If latency is too high, performance may suffer, especially when both sites are updating the database.
- May require a database software upgrade to a version that supports mirroring.

This approach is required when running in **Always Active** Disaster Recovery mode.

Internet Server and Command Center Instances at Disaster Recovery Site

For all the three modes, before entering Disaster Recovery (DR) mode, you must install Internet Server and Command Center at the DR site. You have two options when installing MFT Internet Server and Command Center instances at the DR site:

- Mirror the Primary Site: Create DR MFT Command Center and Internet Server instances for each primary site MFT Command Center and Internet Server instance. This ensures that the DR site has sufficient resources to process all transfers. Generally, the MFT Internet Server and Command Center instances at the DR site should have different host names to avoid confusion with the primary MFT Internet Server and Command Center instances.
- Install Minimum Instances: Install just enough MFT Command Center and Internet Server instances to process the files. For example, if you have two Command Centers and four Internet Server instances at the primary site, you may be able to run with one Command Center and two Internet Server instances at the DR site. The best practice is to install at least one Internet Server in the DR DMZ and one in the DR internal network. The exact number of instances depends on the expected transfer volume.

Other Applications at the Disaster Recover Site

Many other applications can interface with MFT Internet Server and Command Center. This includes TIBCO and Cloud Software Group products and internal applications. TIBCO products that typically interface with MFT include:

- TIBCO Platform™ Servers
    - Initiate transfers to other Platform Servers or to the Internet Server.
    - Used by pDNI Event Driven Processing to transfer files with partners or to Internet Servers.
- TIBCO pDNI
    - Event-driven processing.
    - Can communicate with Platform Servers to transfer files with internal or partner servers.
- TIBCO EMS (Enterprise Messaging Service)
    - Used by business process software, including ActiveMatrix BusinessWorks, to initiate file transfers with MFT Internet Server.
    - EMS DR instances can be integrated with the primary EMS server instances.
    - MFT Internet Server and Command Center require a DNS name to connect to TIBCO EMS. You can configure multiple IP names on the MFT URL that can be used to connect to EMS.

      Note: Be careful when adding a DR EMS DNS name to the MFT URL. If connections cannot be made to the primary EMS, requests may accidentally be sent to the DR EMS.
- TIBCO ActiveMatrix BusinessWorks
    - Typically used to create complex processes to perform business functions, including MFT file transfers.
    - Uses EMS to communicate with MFT Command Center to initiate transfers.
    - Can issue REST calls directly to MFT Command Center to perform administrative functions, such as creating users, servers, and transfers.
    - Can issue REST calls directly to MFT Command Center to perform the following file transfers:
        - Internet Server file transfers
        - Platform Server file transfers
- TIBCO Rendezvous®

    - Used by MFT to communicate with TIBCO Hawk®
- TIBCO Hawk
    - MFT (version 8.6.0 and above) uses TIBCO Hawk to monitor MFT Internet Server and Command Center.
    - MFT uses a TIBCO Rendezvous connection to communicate with the Hawk Agent and the Hawk server.

Moving to the DR Site

Assume that the primary site is down and cannot be recovered for an extended period. After you decide that you wabt to move to the Disaster Recovery (DR) site, you must perform the following steps:

Database Considerations

The action needed depends on whether the databases were mirrored or replicated, or backed up and restored:

- Mirrored or replicated databases
    - No action is needed as the DR database should be up to date.
- Backed Up and Restored Databases
    - Restore the most current database backup to the DR database server.

Changing Any Internal IP Names and IP Addresses

You must make the necessary changes to the following files:

- Server definitions
- JMS configurations
- Server configurations

Changing DNS Servers, Routers, and Firewalls

- DNS Changes

    - Route external connections to the DR server.
    - Route internal connections to DR internal servers.
- Firewalls and Routers
    - Route external connections to the DR server.
    - Route internal connections to DR internal servers.

Starting Internal Applications on DR Servers

- TIBCO Applications
    - TIBCO MFT Internet Server and TIBCO MFT Command Center instances.
    - TIBCO MFT Platform Servers and event-driven Processing (pDNI) instances.
    - TIBCO ActiveMatrix BusinessWorks™ instances.
    - TIBCO Hawk® instances.
- Internal Applications

Note: Outgoing requests (requests initiated by the DR site) to partner servers typically run without changes, assuming the partner firewall ports are open. Typically, when partners open firewall ports for incoming requests, the partner should open ports for both the primary site and the DR site. This ensures error-free transfers when moving to the DR site.

Returning to the Primary Site

Once the disaster has been resolved and you return to the primary site, ensure that the primary site is set up to seamlessly accept transfers. Perform the following steps to return to the primary site:

Database Considerations

The action needed depends on whether the databases were mirrored or replicated, or backed up and restored:

- Mirrored or replicated databases
    - Ensure that the MFT database has had sufficient time for all changes made on the DR site to be replicated to the primary site database.
- Backed Up and Restored Databases
    - Back up the DR database and transfer the back-up file to the primary site.
    - Disable access to the DR site to prevent further updates to the database.
    - Stop the MFT Internet Server and Command Center instances.
    - Change firewall rules to deny access to the MFT Internet Server and Command Center instances.
    - Restore the database to the primary database server.

Restoring Any Modified IP Addresses

Restore the changes made to the following files:

- Server definitions
- JMS configurations
- Server configurations

Restoring Changes Made to DNS Servers, Routers, and Firewalls

- DNS Changes
    - Route external connections back to the primary server.
    - Route internal connections back to internal servers.
- Firewalls and Routers
    - Route external connections back to the primary server.
    - Route internal connections back to internal servers.

Restarting Internal Applications

- TIBCO Applications

    - TIBCO MFT Platform Servers and event-driven Processing (pDNI)
    - TIBCO ActiveMatrix BusinessWorks™
    - TIBCO Hawk®
- Internal Applications

Interface to Other TIBCO Products

TIBCO MFT interfaces with TIBCO® EMS and TIBCO ActiveMatrix BusinessWorks™. Below is an explanation of the interfaces supported.

ActiveMatrix BusinessWorks™

TIBCO MFT has plug-ins for ActiveMatrix BusinessWorks™, that is for both TIBCO Business Studio™ (BW6) and TIBCO Designer™ (BW5). The plug-ins support the following capabilities:

1. Initiate a Platform Server transfer.
2. Wait for a Platform Server transfer to complete.
3. Initiate an Internet Server transfer.
4. Wait for an Internet Server transfer to complete.
5. Inquire on Internet Server or Platform Server audit records.
6. Wait for alerts.

The plug-ins are not shipped with ActiveMatrix BusinessWorks™ or with any TIBCO MFT product. You must download these plug-ins from the TIBCO download site and then install and configure them in ActiveMatrix BusinessWorks™ before they can be used.

The plug-in interface uses EMS (or other JMS Servers) as a pipe for receiving data from and sending data to the ActiveMatrix BusinessWorks™ client. Therefore EMS (or other JMS Servers) are required for ActiveMatrix BusinessWorks™ clients to work.

To use EMS or JMS, the MFT Command Center is required. Only MFT Command Center can configure EMS/JMS connections.

TIBCO EMS (or JMS)

You can use TIBCO Managed File Transfer to interface with EMS or other supported JMS products such as ActiveMQ or IBM MQ. The following capabilities are supported through EMS/JMS:

1. The following features are available from ActiveMatrix BusinessWorks™:

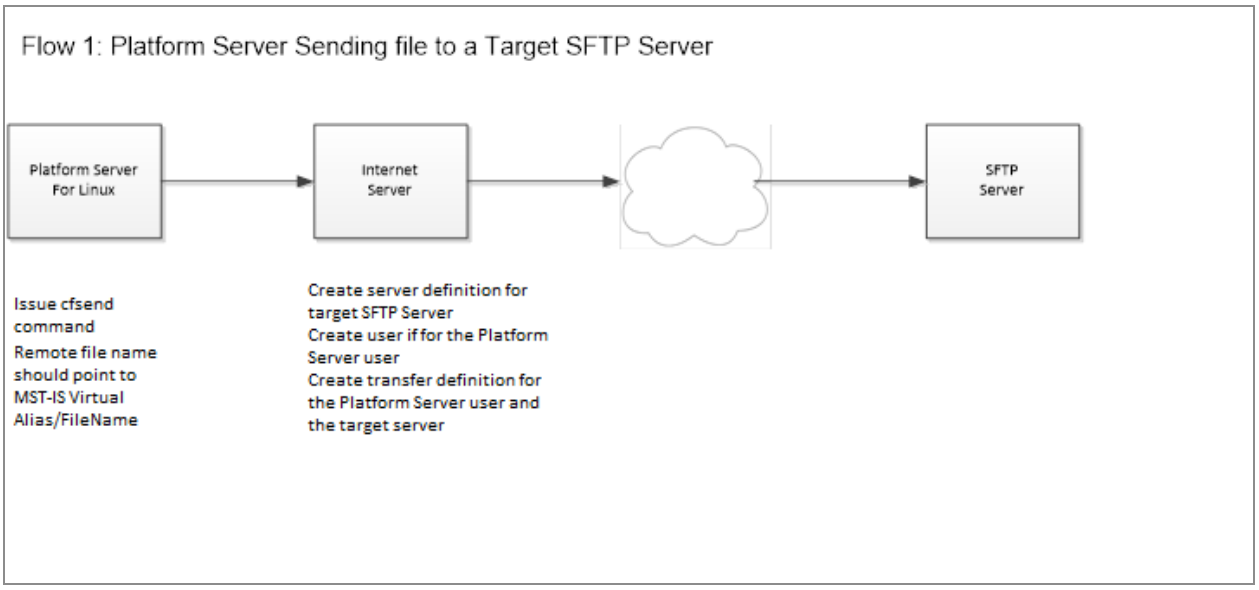
   a. Initiate a Platform Server transfer.
   b. Wait for a Platform Server transfer to complete.
   c. Initiate an Internet Server transfer.
   d. Wait for an Internet Server transfer to complete.
   e. Inquire on Internet Serve or Platform Server audit records.
   f. Wait for alerts.
2. Write file transfer data to an EMS/JMS queue, instead of to a file.
3. Read file transfer data from an EMS/JMS queue, instead of from a file.
4. Write transfer start (Internet Server) notifications to a topic.
5. Write transfer completion (Internet Server and Platform Server) notifications to a topic.

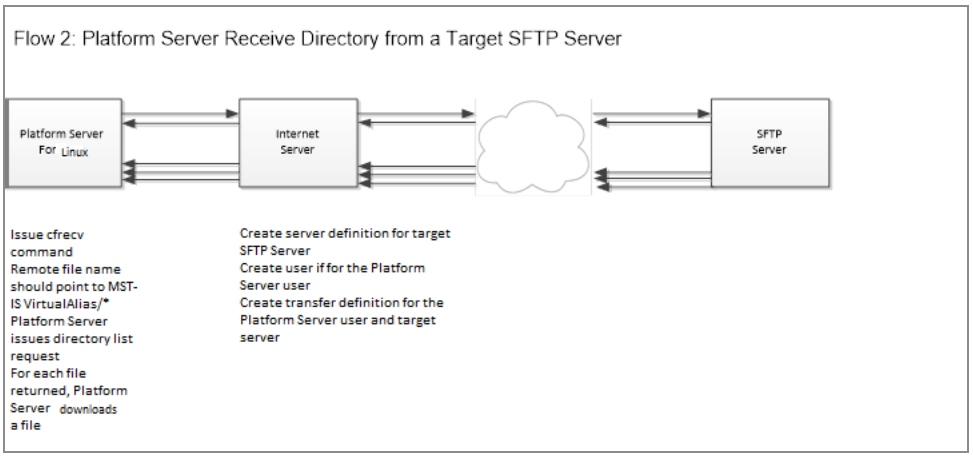
TIBCO Spotfire® and TIBCO JasperReports®

TIBCO MFT does not have a direct interface to Spotfire® or TIBCO JasperSoft®, but these products can be used to create sophisticated reports above and beyond the reports created by MFT Command Center. All MFT configuration and audit information is stored in the MFT database. So it is a relatively simple task for users familiar with Spotfire® or JasperReports® to create reports using the MFT database input.

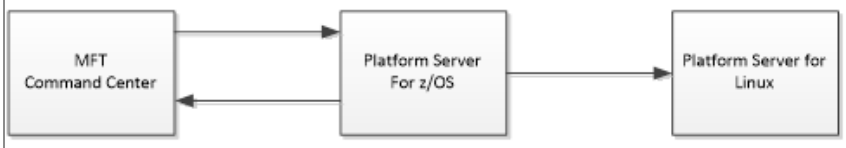
Sample Transfer Flows

Here are some sample transfer flows.





Flow 3: Command Center Initiates a Platform Server Transfer



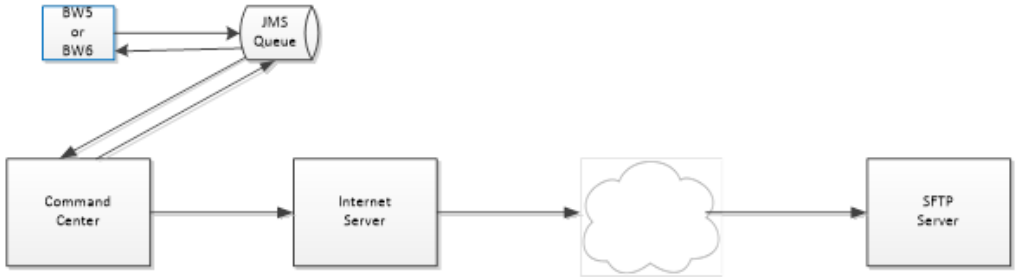
Command Center creates server definition for Platform Server for z/OS
Command Center initiates connection to Platform Server for z/OS
Command Center notifies Platform Server for z/OS to send a file to Platform Server for Linux
Command Center waits for a completion response from Platform Server for z/OS

Platform Server for z/OS receives request from Command Center
If Command Center is not waiting for transfer completion, Platform Server for z/OS notifies Command Center that the that request is accepted
Platform Server for z/OS initiates transfer to Platform Server for Linux
If Command Center is not waiting for transfer completion, when transfer completes, Platform Server for z/OS notifies Command Center that the transfer is completed

Platform Server for Linux processes the request like any other transfer request.

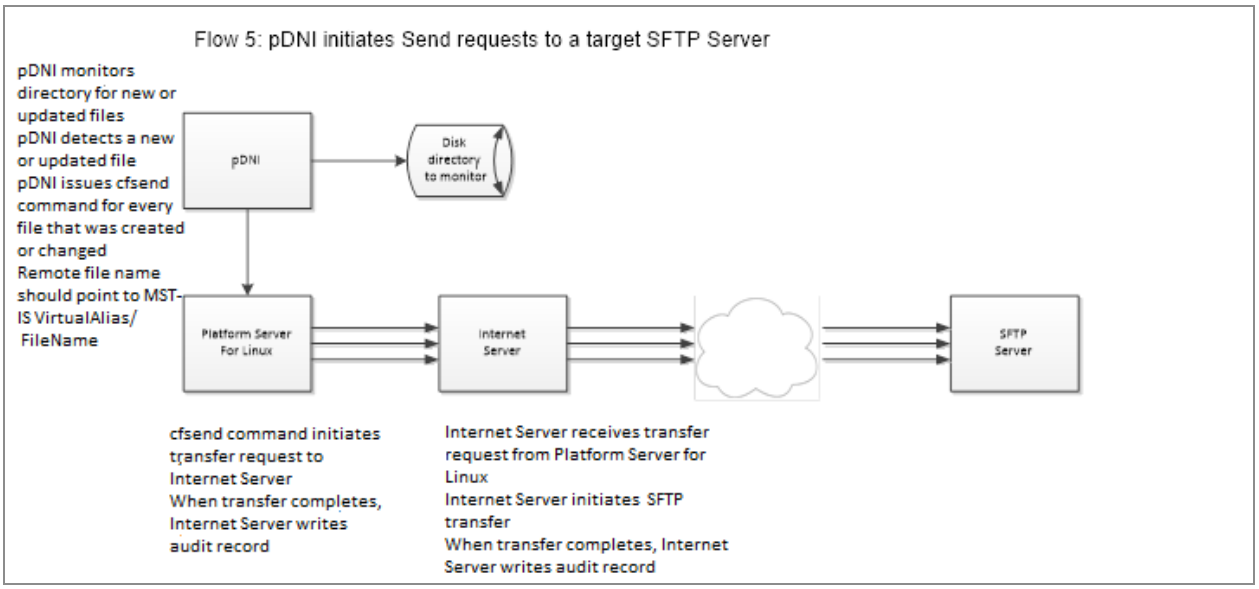
Flow 4: JMS initiates an Internet Server Transfer and waits for a response

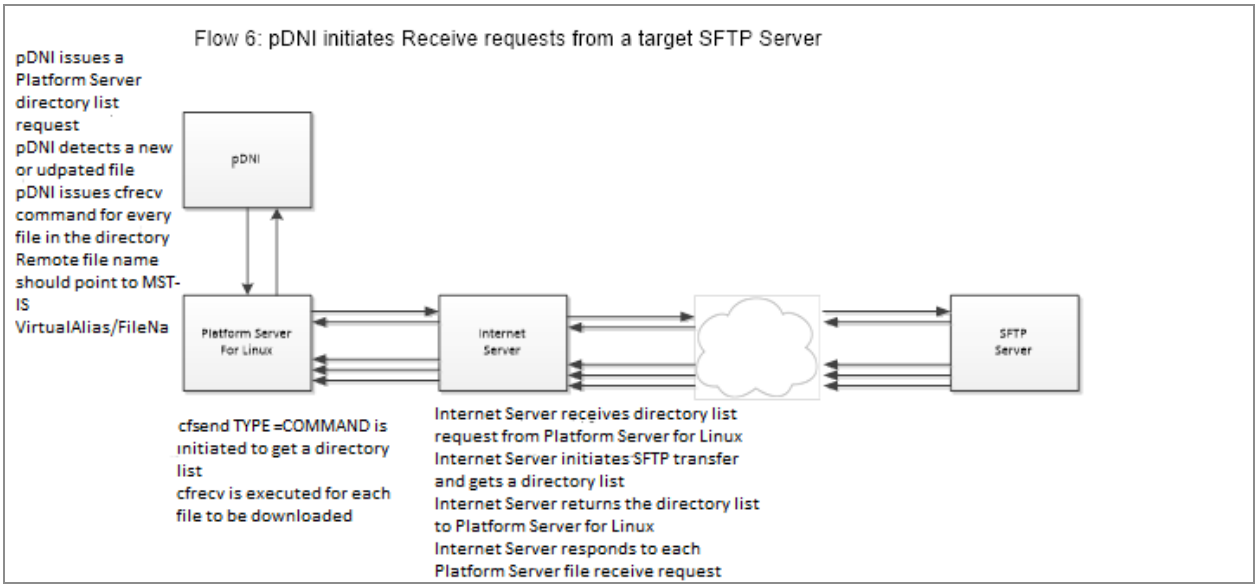
BW writes XML to JMS transfer queue
BW waits for "Transfer Submitted"message
BW waits for "Transfer Complete"message



Detects JMS message on the JMS transfer queue
Initiates transfer request to Internet Server
Writes "Transfer Submitted" message to JMS transfer queue
Polls the Audit file until the transfer completes
Writes "Transfer Complete" message to JMS transfer queue

Internet Server receives Command Center transfer request
Internet Server initiates transfer request to the target SFTP Server
Internet Server writes an audit record when the transfer completes


Flow 5: pDNI initiates Send requests to a target SFTP Server
pDNI monitors directory for new or updated files
pDNI detects a new or updated file
pDNI issues cfsend command for every file that was created or changed
Remote file name should point to MST-IS VirtualAlias/ FileName
pDNI
Disk directory to monitor
Platform Server For Linux
Internet Server
SFTP Server
cfsend command initiates transfer request to Internet Server
When transfer completes, Internet Server writes audit record
Internet Server receives transfer request from Platform Server for Linux
Internet Server initiates SFTP transfer
When transfer completes, Internet Server writes audit record


Flow 6: pDNI initiates Receive requests from a target SFTP Server
pDNI issues a Platform Server directory list request
pDNI detects a new or udpated file
pDNI issues cfrecv command for every file in the directory
Remote file name should point to MST-IS VirtualAlias/FileNa
pDNI
Platform Server For Linux
Internet Server
SFTP Server
cfsend TYPE =COMMAND is initiated to get a directory list
cfrecv is executed for each file to be downloaded
Internet Server receives directory list request from Platform Server for Linux
Internet Server initiates SFTP transfer and gets a directory list
Internet Server returns the directory list to Platform Server for Linux
Internet Server responds to each Platform Server file receive request

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.