

Review Article

Open Access

Scheduled File Transfer Job Failures: Reasons and Remediation Measures from an MFT Perspective

Prashanth Kodurupati

Information Technology, Managed File Transfer Engineer, PragmaEdge LLC, Alpharetta, United States of America

ABSTRACT

Scheduled pull jobs/file transfers are a critical and widely used automation element of Managed File Transfer (MFT) systems. These pull jobs may fail for a number of reasons, including wrong file path, missing the file availability window, misconfigured pull request, authentication issues, and network connectivity issues (temporary). These issues can be remediated by establishing proper communication channels between clients and hosts and updating the clients with necessary information like file paths, dependencies, etc.

*Corresponding author

Prashanth Kodurupati, Information Technology, Managed File Transfer Engineer, PragmaEdge LLC, Alpharetta, United States of America.

Received: August 03, 2022; Accepted: August 08, 2022; Published: August 14, 2022

Keyword: Scheduled File Transfer

Introduction

Automation is key to making routine operational processes like file transfers and transfer permissions seamless and effortless, allowing relevant professionals like Managed File Transfer (MFT) engineers to focus on other important tasks. However, automating certain tasks like pull and push jobs from the MFT system has its own set of challenges, mostly stemming from an improper setup. One of these challenges is customers failing to successfully “pull” a file from the host server.

Literature Review

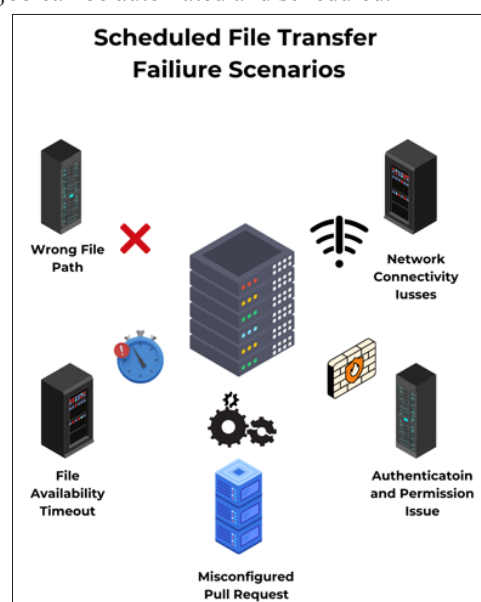
The literature review on Managed File Transfer and its automation element is adequately extensive. It covers several facets of MFT systems, their strengths, and benefits, including the fact that automation can help reduce human errors [1]. The process of scheduling file transfer is also considered an MFT automation strength that not only reduces human errors but also makes business integrations and operations more seamless and efficient [2]. However, automation alone pertains to file transfer in both ways, i.e., from client to server and from server to client. We can limit it to “pull jobs” where clients schedule/request a file transfer from the host (business servers) [3]. The literature (especially MFT guides) also discusses the importance of file transfer paths on both the client and server-side [4]. The available academic literature covers several different aspects of MFT automation. However, the scenarios specific to failures of scheduled transfer jobs are more actively discussed in forums and blogs than in academic literature.

Problem Statement: Scheduled File Transfer Job failures

MFTs are used to schedule, regulate, and facilitate a wide range of file transfers. One category these transfers can be divided into is push and pull transfers. From a server’s/business’s perspective, a pull job is when a client-server is requesting a file from the server, i.e., it’s being pulled away from the server. This is typically

regulated through permissions and tracked to ensure that the right individuals, client servers, etc., are capable of initiating or following through with a pull job. A real-time log of each transfer/pull job in the MFT ensures that the MFT engineers responsible for the server and all file transfers have visibility over pull requests, completions, and failure instances. In contrast, a push request is a server pushing a file toward a client server.

Both of these transfer jobs can be initiated and completed manually or scheduled. For example, a bank may request updated data on the credit reports of their customers at the end of the day or the beginning of the day, and since the same set of files is requested, the pull job can be automated and scheduled.



Sometimes, these transfers fail, and there is a variety of scheduled pull job fail scenarios:

Improper File Path Location

A scheduled pull job may request the file from host servers following the wrong mailbox path/file path, resulting in a failure. This may stem from a number of underlying problems like a miscommunication between the host server (business) and the client-server (clients), resulting in no or wrong file path being communicated, causing a pull job to revert to default settings. It may also happen when a file location is changed within the host server, or files are bundled in a different way than usual. Changes in naming conventions can also render a file path redundant.

File Availability Timeout

Most servers and their MFT systems make certain files available for client “pulls” for a limited time period. There are several reasons for that, including traffic management and bandwidth control. If the scheduled pull job doesn’t take advantage of the window available, the transfer permissions may expire, resulting in the failure of the scheduled pull job.

Misconfigured Pull Job

It’s possible to misconfigure the pull request in a number of ways, including adding the wrong path or not initiating the request while keeping the MFT requirements of the host server in mind. Misconfiguration may also stem from not including the necessary dependencies for file transfer.

Authentication and Permission Issues

If a scheduled file transfer (pull job) doesn’t have the requisite permissions for a specific file or is not authenticated to request a file transfer without the right information or conditions, the transfer may fail [5]. This is something that may change from time to time, so scheduled jobs might result in successful file transfers one day but not the next (for example, if permissions are not updated after a system update or overhaul). Using the wrong authentication protocols may also result in a transfer failure, like setting up certificates when the pull request requires an ID and password.

Network Connectivity Issues

Even if the pull job is properly configured and the individual/team requesting the transfer has the right authentication and permissions to request the file, the scheduled file transfer job may still fail if there are temporary network connectivity issues. They can prevent the client from pulling the requisite file even if the issue is resolved if it persists long enough for the temporary transfer permission to expire.

Proposed and Implemented Solutions

The proposed solutions vary based on the problems. Most of the solutions can also be evaluated from both client and host server perspectives.

Communicating the Right File Path to Clients

Before scheduling a pull job, the client-server should ensure that they have access to the right file path. If they are aware of any development that may have resulted in the change of the file path they have access to, like a system update, changes in naming convention, host server switching to new FTP solutions, etc., they may consider proactively requesting the right file path or ensuring that the one they have access to is still valid.

The host server should communicate the appropriate file path to all client servers that may request a file through manual or automatically scheduled pull jobs ahead of time, especially if they have changed for any reason. This can prevent them from

being overloaded with client requests later on or modifying file availability expiration timelines that may interfere with the subsequent files that are made available for transfer.

In many instances, this updated file path can be communicated to the client servers automatically, but with highly restrictive files, sensitive files, or transfer instances that are not routine, the right path may be communicated directly.

Allocating Adequate Time for a Pull

Both clients setting up a pull job and host servers deciding upon the time a file should be made available for transfer should ensure that the other entity has the necessary information available. This includes both standard and custom availability time. For some files, there may be established protocols when it comes to making the files available for download, and they may be influenced by several factors, including different time zones of the client and host servers, file size, number of clients accessing it, bandwidth constraints, etc.

The goal should be to allocate enough time for all clients to access and download the file, even if almost all requests are made around the same time, which may lead to network congestion.

Proper Pull Job Configuration and Scheduling

For a pull job configuration, the host’s responsibility is to communicate all the necessary parameters and conditions to the client. This includes FTP protocols, download availability windows, authentication requirements, dependencies, etc.

Then, it’s the client’s responsibility to ensure that each pull request is configured as per the conditions set by the host servers. This includes setting up the right dependencies, providing relevant authentication information, scheduling the pull request during the availability window, and even setting up retries in case of network connectivity issues. This is especially important for the automated pull requests scheduled when there may not be someone to tackle and troubleshoot the systems or reconfigure the request in real-time when a transfer fails. The client should also ensure that there are enough “retries” configured to make the file transfer possible, even if network connectivity issues cause the transfer to pause or fail for a few instances.

Ensuring Adequate Permissions/Authentication

Host servers should focus on defining permissions and authentications quite clearly [6]. This will ensure that only the right people and clients have access to the files and can pull them on their servers, and it would also make it easy for client servers to configure the pull requests. For scheduled pull requests, it’s important to make sure they are configured with the necessary access rights and permissions.

Resolving Internet Connection Issues

This pertains mostly to the host side of the file transfer. Host servers should ensure that they have adequate bandwidth to accommodate all pull requests within the allocated time window. In some cases, the servers may have to ensure that adequate infrastructure is there to accommodate the pull requests and traffic movement without congestion blocking pull requests (resulting in failures) or causing other network hindrances.

Use Case

Many of these use cases are generated from the perspective of a credit bureau and its clients, but they can be translated to a

comprehensive range of businesses and industries.

Problem	Solutions	Use Cases (MFT on Client & Server)
Improper Path	* Client: Verify path with credit bureau before scheduling pull job.	* Credit bureau updates file structure and notifies all clients of the new path.
	* Server (Credit Bureau): Communicate path changes proactively to clients.	* New credit bureau merges with another, requiring clients to update paths.
	* Review logs for failed transfers and identify path-related errors.	* Client misspells directory name in pull job configuration.
File Timeout	* Client: Allocate sufficient time for file transfer based on credit bureau guidelines.	* Credit bureau sets a short availability window due to high traffic volume.
	* Server (Credit Bureau): Consider extending the availability window for critical files.	* Credit bureau experiences network congestion during peak transfer times.
	* Implement retry logic on the client side for failed transfers.	
Misconfiguration	* Client: Review the pull job configuration for errors in authentication, protocols, etc.	* Client configures the pull job using an incorrect authentication method (username vs cert).
	* Server (Credit Bureau): Clearly document pull job requirements for clients.	* Client forgets to update security protocol (FTPS vs FTP) after credit bureau upgrade.
	* Utilize MFT features for validation and error checking during configuration.	
Permissions/Auth Issues	* Client: Ensure proper access rights are configured for the scheduled pull job.	* Credit bureau updates access control policies for specific data sets.
	* Server (Credit Bureau): Define clear authentication and authorization rules for clients.	* Client loses access credentials due to personnel change at the credit bureau.
	* Review logs for access-related errors during failed transfers.	
Network Issues	* Server (Credit Bureau): Ensure adequate network infrastructure for concurrent transfers.	* Credit bureau experiences unexpected network outage during scheduled transfer.
	* Client: Consider retry logic and timeouts for network connectivity issues.	* The client's internet service provider has a temporary service disruption.

Conclusion

The scheduled file transfers can fail for any number of reasons, from wrong file paths and missing download windows to network issues. However, the underlying issue at the root of the problems and proposed and implemented solutions is clarity and communication between client and host servers. MFT engineers

on both ends of the transfer should develop a comprehensive understanding of parameters and requirements around pull requests to minimize the instances of transfer failures.

References

1. Sardjono W, Pujadi T (2016) Performance evaluation of systems Managed File Transfer in banking industry using IT Balanced Scorecard. 2016 International Conference on Information Management and Technology (ICIMTech) <https://ieeexplore.ieee.org/document/7930294>.
2. Dunford D (2013) Managed file transfer: the next stage for data in motion? Network Security 12-15.
3. Pick B (2017) What is MFT and How Does It Work? Fortra <https://www.goanywhere.com/blog/how-does-mft-work>.
4. (2016) TIBCO MFT Internet Server Quick Start Guide. TIBCO https://docs.tibco.com/pub/managed-file-transfer-internet-server-rocketstream/7.2.6/doc/pdf/TIB_mftis-rs_7.2.6_quick_start_guide.pdf.
5. Tseng YF, Fan CI, Cho YF (2018) An authenticated re-encryption scheme for secure file transfer in named data networks. Communication Systems 31.
6. Alhothaily A, Hu C, Alrawais A, Song T, Cheng X, et al. (2017) A Secure and Practical Authentication Scheme Using Personal Devices. IEEE Access 5.

Copyright: ©2022 Prashanth Kodurupati. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.