Informatica® Managed File Transfer
10.2.3

HTTPS Automated Connection Guide

Informatica Managed File Transfer HTTPS Automated Connection Guide
10.2.3
February 2019

Publication Date: 2020-05-21

Table of Contents

Preface

Use the *Informatica HTTPS Automated Connection Guide* to learn about the operational tasks available with the Managed File Transfer HTTPS automated connection. Learn how to run file transfers with the Managed File Transfer HTTPS Server. You can also learn how to use the Secure Mail feature for sending packages.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit https://network.informatica.com.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

Informatica maintains documentation for many products on the Informatica Knowledge Base in addition to the Documentation Portal. If you cannot find documentation for your product or product version on the Documentation Portal, search the Knowledge Base at https://search.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at http://velocity.informatica.com. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

To find online support resources on the Informatica Network, visit https://network.informatica.com and select the eSupport option.

CHAPTER 1

Introduction

The Managed File Transfer HTTPS Server is a web server application built to function similar to an FTP/FTPS server from the client's viewpoint. Although the HTTPS Server comes with a GUI web client, some users may wish to automate their file transfers to and from this server.

Disclaimer: Informatica reserves the right to revise the Automated Connection API at any time. When upgrading to the latest version of Managed File Transfer, you should thoroughly test your processes to make sure they function as expected.

Connecting

The first step to running transfers against the Managed File Transfer HTTPS Server is to establish a connection to the server. The connection is made using the HTTPS (HTTP over SSL) protocol. Unlike FTP, there is no explicit connection step made to the server. Simply make a request to the Login URL and both the connection to the server and the login attempt are made.

Hostname and Port

The hostname and port number used to connect to the Managed File Transfer HTTPS Server will be provided by the server's administrator. If your installation of Managed File Transfer is using the default port number 443, it is not required to include the port number in the command URI.

For example, if the hostname is '192.168.1.100' and the port number is 443, your connection URL could be:

```
https://192.168.1.100/login
```

If the hostname is 'mycompanyname.com' and the port number is 9443, then your URL could be:

```
https://mycompanyname.com:9443/login
```

Credentials

The credentials to use the Managed File Transfer HTTPS Server will be provided to you by the server's administrator.

Handling Responses

Each request is answered with a response. Managed File Transfer HTTPS Server does not use the standard HTTP response codes to return information about the request back to the client. Standard response codes can be used to determine that the request was received by a Managed File Transfer HTTPS Server. The actual message reply from the server is stored as a custom response header named X-GDX-Reply.

The X-GDX-Reply header will contain success or error codes and messages specific to Managed File Transfer HTTPS Server. The format of this header message is a status code followed by a single white space, followed by the message details.

```
200 Welcome, testuser!
```

These header codes and messages can be utilized to determine the success or failure of the operation.

Status Codes

200 - 299

`Informational or success status codes` - The operation performed against the server were successful.

500 – 509

`Internal server error` - The server experienced a critical error and the server's administrator should be contacted immediately.

510 – 519

`Bad/Invalid request` - The server could not process the request because information supplied by the user is invalid or incomplete. See the X-GDX-Reply header message for more details.

530 – 539

`Login/account related errors`- Indicates that an error occurred with the account or login, such as Invalid login or account disabled. See the X-GDX-Reply header message for more details.

550 – 559

`Permission errors` - The user does not have permission or authority to perform the requested action. See the X-GDX-Reply header message for more details.

560 – 569

`Errors related to files or directories on the system` - An error occurred when accessing a file or directory on the server, such as a file or directory does not exist.

580-589

`File I/O Errors` - An internal server error occurred while trying to access a file or directory.

590

`Unknown error` - An unexpected error occurred while trying to process the command. See the X-GDX-Reply header message for more details.

Request Commands

Each request made to one of the URLs listed below is considered a command that is run against the server. Some commands require parameters, such as 'username' and 'password' to log in to the server. Other commands such as 'pwd' do not require a parameter. Commands are executed by making a request to the command's URLs. For example, the following request is sending a command to the server to list the contents of the current working directory:

```
https://192.168.1.100/list
```

This request sends a logout command to the server:

```
https://192.168.1.100/logout
```

Below is a list of supported commands. Each item details the URL to run the command, which parameters are supported and required, and whether or not the request can be made as a GET, POST, or either. All parameters are required unless noted otherwise.

Authentication

Managed File Transfer HTTPS Server supports password authentication and client certificate authentication. If client certificate authentication is configured, requests to the Login URL are not required and requests to URLs, such as Upload, can be made directly.

If password authentication is required, a request must be made to the Login URL before any other operations can be made.

Login

The Login command is used to authenticate a user. Users can login to Managed File Transfer HTTPS Server by making a request to this URL. Unless client certificate authentication is performed, an initial request should always be made to this URL to start a user session.

URL	Request Type	Parameters
/login	POST	username - the name of the user on the server password - the password required to login

Logout

The Logout command is used to close the user session on the Managed File Transfer HTTPS Server. Make a request to this URL to log out of the server.

URL	Request Type	Parameters
/logout	GET or POST	none

Standard Operations

PWD (Print Working Directory)

The PWD command is used to retrieve the current working directory on the server. Requests made to this URL will return the users current working directory on the server. The absolute path to the current working directory will be returned as part of the X-GDX-Reply header message. The path will be enclosed in double quotes.

URL	Request Type	Parameters
/pwd	GET or POST	none

Delete

The Delete command is used to remove files from the server.

URL	Request Type	Parameters
/delete	GET or POST	file – the relative or absolute path of the file to delete

Rename

The Rename command is used to rename files on the server. If the current working directory contains the file(s) to rename, then the from and to parameters may contain only the file names. However, by being able to provide path information in these parameters, the rename command may be used to move files on the server. For example:

```
http://192.168.1.100/rename?from=file.txt&to=../file.txt
```

This command will move the file 'file.txt' in the current working directory to the parent directory.

URL	Request Type	Parameters
/rename	GET or POST	from - the relative or absolute path of the file or directory to rename to - the relative or absolute path of the new name

List

The List command is used to list the contents of a directory on the server. The target directory can be supplied as a parameter to this command. If the directory is not supplied, this command will list the contents of the users current working directory. The contents of the directory are returned as the response body with content type 'text/plain'. The format of the directory listing is as follows:

```
    2009-12-03 14:02:19 D 0 backup
```

The parts are delimited by a tab (\t) character:

Part 1) The last modified date of the file or directory. The timestamp is in ISO format yyyy-MM-dd HH:mm:ss. Note that the hour(hh) is displayed as a 24-hour clock.

Part 2) Indicates file or directory. The 'D' character denotes that it is a directory and 'F' stands for file. 'U' for unknown/other.

Part 3) The size of the file in bytes.

Part 4) The name of the file or directory.

URL	Request Type	Parameters
/list	GET or POST	dir (optional) – the relative or absolute path of the directory to list.

Checksum

The Checksum command is used to calculate the hash of a remote file. The reply is returned on the first line of the response body and can be used to compare with the hash value of the downloaded local file to verify data integrity. The supported hash algorithms are SHA1, MD5, and CRC32.

URL	Request Type	Parameters
/hash	GET or POST	file (required) – the path relative to the current working directory, or an absolute path to the file. algorithm – the hash algorithm to use when calculating a checksum. Valid values are SHA1 (default), MD5, or CRC32. length - The starting position within the file. This value is used for calculating partial file checksums. By default the value is 0, which will perform the checksum on the entire file.

CD (Change Directory)

The CD command is used to change the current working directory. The absolute path to the new working directory will be returned as part of the X-GDX-Reply header message. The path will be enclosed in double quotes.

URL	Request Type	Parameters
/cd	GET or POST	dir – the relative or absolute path of the target directory.

CDUP (Change Directory Up)

The CDUP is a convenience command that is used to change the current working directory to the parent directory. The absolute path to the new working directory will be returned as part of the X-GDX-Reply header message. The path will be enclosed in double quotes.

URL	Request Type	Parameters
/cdup	GET or POST	none

MKDIR (Make Directory)

The MKDIR command is used to create a new directory on the server. The absolute path to the newly created directory will be returned as part of the X-GDX-Reply header message. The path will be enclosed in double quotes.

URL	Request Type	Parameters
/mkdir	GET or POST	dir – the relative or absolute path of the directory to create.

File Information

The File Information command is used to retrieve information about a specific file or directory. The information is returned as the response body with content type 'text/plain'. The format of the file information is identical to the listing returned from the List command. If no information is returned in the response body, then the file or directory does not exist.

URL	Request Type	Parameters
/fileInfo	GET or POST	file – the relative or absolute path of the file or directory to retrieve information about.

File Transfer

Upload

The upload command is used to transfer a file to the server. The request must be a multipart POST request and only one file may be uploaded per request. A file is a required part of the multipart request, but any name parameter name given to the file part will be ignored.

URL	Request Type	Parameters
/upload2	POST/ Multipart	to – the relative or absolute path of the destination file. append (optional) – If the file exists on the target directory, set this parameter to true to append the new file to the existing one. transferMode – use B for binary transfers (default) or A for ascii transfers. file – the file being uploaded as part of the multipart request.

Upload Raw Data

The Upload Raw Data command is used to upload data directly to the server where the data is the content of the request body. This request must be a POST request. The name of the file will be automatically derived

and will be returned as part of the X-GDX-Reply header message. This is a special command where the request body must contain the file data being uploaded.

URL	Request Type	Parameters
/uploadRawData	POST	none

Download

The Download command is used to download a file from the server. The file will be returned as the response body. The content type will always be application/force-download, along with the content disposition field containing the name of the file. The content-length header is also included in the response indicating the size of the file.

URL	Request Type	Parameters
/download	GET or POST	file (required) – the file to download. This can be a path relative to the current working directory, or an absolute path to the file. offset – for downloading partial files. Enter the starting position of the file to begin downloading from. transferMode – use B for binary transfers (default) or A for ascii transfers.

CHAPTER 2

Secure Mail

This section outlines how an external client application, such as a plugin for Microsoft Outlook, Lotus Notes, etc. can interface with Managed File Transfer in order to utilize the Secure Mail feature for sending packages.

The external client application will need to connect and log in to Managed File Transfer as a Web User and make a request to one of the HTTPS servlets for processing. These servlets allow you to create a package, attach files to a package, send a package, and retrieve current Secure Mail settings.

Create Package

This servlet is used to create the package. Validation is done on the server side to ensure the submitted configuration is acceptable based on the package requirements defined by the server administrators. Once validated, the package will be created in DRAFT mode. The package ID is determined at the time the package is created and will be returned as the response body of the request made to this servlet.

URL	Request Type	Parameters
/createPackageWithOptions	POST	The only requirement of this request is the request body containing the XML definition of the package.

XML Example

The following is a sample XML definition that would be sent to the servlet. The content type should be text/xml.

```
<createPackageXML class="com.linoma.commons.packages.CreatePackageXML">
<toAddress>
<![CDATA[sales@linoma.com, support@linoma.com
</toAddress>
<subject>
<![CDATA[This is a test
</subject>
<message>
<![CDATA[This is the message for my test
</message>
<protectionLevel>password</protectionLevel>
<passwordGeneration>manual</passwordGeneration>
<password>mypass</password>
<expiresAfter>5</expiresAfter>
<maxDownloads>3</maxDownloads>
<replyAllowed>true</replyAllowed>
<readReceipt>true</readReceipt>
<includePassword>true</includePassword>
</createPackageXML>
```

XML Field Definitions

toAddress

The address of where the package should be sent

- This element is required
- Contents can be enclosed within a CDATA tag

subject

The subject to be used for the package

- This element is required
- Contents can be enclosed within a CDATA tag

message

The message contents describing the package

- This element is required if the package does not include attachments
- Contents can be enclosed within a CDATA tag

protectionLevel

The protection level for the package

- This element is not required and defaults to the Secure Mail settings
- It supports the options ‘url’, ‘password’, and 'certified'

passwordGeneration

The password generation configuration

- This element is not required and defaults to the Secure Mail settings
- It supports the options ‘manual’ and ‘automatic’

password

The password to be used

- This element is required if passwordGeneration is set to manual

includePassword

The option of whether or not to include the password in the email

- This element is not required and defaults to the Secure Mail settings
- This may not be allowed depending on the admin settings

expiresAfter

The number of days that the package will expire in

- This element may be required depending on the Secure Mail settings
- If omitted, then default specified in the Secure Mail settings will be used
- If the default is not specified, the package will be set to never expire

maxDownloads

The maximum number of downloads allowed for each file in the package

- This element may be required depending on the Secure Mail settings

- If omitted, it will default to no limit on the number of downloads depending on the Secure Mail settings

readReceipt

The option of sending a read receipt to the sender

- This element is optional
- It supports the values of ‘true’ and ‘false’
- If omitted, it will default to false

replyAllowed

The option of allowing replies from non-registered recipients of this package

- This element is not required and defaults to the Secure Mail settings
- This may not be allowed depending on the Secure Mail settings
- It supports the values of 'true' and 'false'

Attach File

This servlet is used to attach files to a package. The package ID returned from the Create Package with Data request must be sent along with each file to attach. The files must be attached one at a time and therefore cannot contain more than one file in a single request. If more than one file is attached with a single request, an error will be thrown.

URL	Request Type	Parameters
/attachLocalFileToPackage	POST - Multipart	packageId – The ID of the package to attach the file to. file – The file being attached to the package.

Send Package

This servlet is the final step for sending secure mail packages. After all files have been attached, a request is made to this servlet which does the final round of validation on the package and submits it to be sent by Managed File Transfer.

URL	Request Type	Parameters
/sendPackage	POST	packageId – The ID of the package to be sent.

Settings

This servlet is available for authenticated Web Users with secure mail capabilities. With this servlet, users can query the current settings and configuration of secure mail so they know how packages are required to be configured.

URL	Request Type	Parameters
/secureMailSettings	POST	none

XML Example

The response body will contain XML with all the current settings. If the settings are not set by the server, the entry will be omitted from the xml.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Secure Mail Settings</comment>
<entry key="defaultSubject">Subject of the Day</entry>
<entry key="enabledProtectionLevels">U,P</entry>
<entry key="defaultProtectionLevel">P</entry>
<entry key="enabledPasswordGenerations">S,M</entry>
<entry key="defaultPasswordGeneration">M</entry>
<entry key="allowIncludePasswordInEmail">true</entry>
<entry key="defaultIncludePasswordInEmail">true</entry>
<entry key="enforceExpirationRange">true</entry>
<entry key="packageExpirationMin">1</entry>
<entry key="packageExpirationMax">7</entry>
<entry key="defaultPackageExpiration">3</entry>
<entry key="enforceDownloadRange">true</entry>
<entry key="maxDownloadsMin">1</entry>
<entry key="maxDownloadsMax">3</entry>
<entry key="defaultMaxDownloads">2</entry>
<entry key="replyAllowed">true</entry>
<entry key="defaultReplyAllowed">true</entry>
</properties>
```

XML Field Definitions

defaultSubject

The default subject to be used for the package

enabledProtectionLevels

Which protection levels are supported

- ‘U’ = URL and/or ‘P’ = Password and/or 'C' = Certified Delivery

defaultProtectionLeve

IThe default protection level is used

- ‘U’ = URL, ‘P’ = Password, or 'C' = Certified Delivery

enabledPasswordGenerations

Which password generation methods are supported

- ‘S’ Generated Automatically and/or ‘M’ Manually Specified

defaultPasswordGeneration

The default password generation method

- ‘S’ Generated Automatically or ‘M’ Manually Specified

allowIncludePasswordInEmail

Whether or not the password may be included in the email

'true' or 'false'

defaultIncludePasswordInEmai

lThe default setting for including the password in the email

'true' or 'false'

enforceExpirationRange

Enforce specific day(s) range for package expiration

'true' or 'false'

packageExpirationMin

Minimum number of day(s) allowed for package expiration

packageExpirationMax

Maximum number of day(s) allowed for package expiration

defaultPackageExpiration

The default number of days until the package expires

enforceDownloadRange

Enforce a specific range for number of downloads allowed per file

'true' or 'false'

maxDownloadsMin

Minimum number of downloads allowed per file

maxDownloadsMax

Maximum number of downloads allowed per file

defaultMaxDownloads

The default number of downloads allowed per file

enabled

The setting if secure mail is enabled

'true' or 'false'

replyAllowed

Allows the non-registered recipients to reply to a package

- 'true' or 'false'

defaultReplyAllowed

The default value for allowing non-registered recipients to reply to a package