covisint
a subsidiary of Compuware Corporation

FTP Connector

August 2004
Product Revision Level 2.0.0

Table of Contents

INTRODUCING THE FTP CONNECTOR

This guide contains information needed to use the File Transfer Protocol (FTP) service on the Covisint Connect messaging hub.

With the FTP Connector described in this guide, you can use FTP commands to exchange information with your trading partners through the Covisint Connect messaging hub.

To use the FTP Connector, you should have prior knowledge of FTP commands and how to use them.

The FTP Connector interface allows users with standard FTP client software and TCP/IP communications over ANX, ENX or VPN to execute the following actions:

- Submit business documents to be processed by the Covisint Connect messaging hub.
- Retrieve business documents from a Covisint Connect messaging hub mailbox.

Implementation – FTP Connector

Covisint has implemented FTP (File Transfer Protocol) access to the messaging hub using a custom FTP server. File transfers use normal FTP commands. In other words, FTP commands such as ***get*** and ***put*** are no different than standard FTP used today.

The application-level protocol used for the FTP server access methods are based on the File Transfer Protocol outlined in the Internet document RFC 959.

Requirements – FTP Connector

Our requirements assume the trading partner (FTP user) has completed registration and has created FTP inbound and outbound connectivity channels with Covisint. This includes network connectivity and the exchange of ***usernames*** and ***passwords***.

Getting Started – FTP Connector

To use the FTP Connector, follow these steps:

1 Obtain a User ID and password from Covisint.

This partnership is established when you register with Covisint. If you have any questions, contact your Covisint representative.

2 Determine FTP server and establish connectivity.

Before connecting to the FTP Connector, the necessary communication requirements must be satisfied. The following table lists the IP addresses for the FTP Connector.

Network	IP Address	FTP Control Port	FTP Data Port
ANX	206.18.241.63	21	20

3 Log in.

Note:

Bold text indicates user input.

Using a standard console-based FTP client, log in with your user ID and password to establish an FTP session with the messaging hub.

ftp> ***206.18.241.63***

Connected to 206.18.241.63.

220 Service ready for new user

Name (206.18.241.63:account): ***user ID***

331 User name okay, need password for {user ID}

Password: ***Password***

230 User logged in, proceed

Remote system type is UNIX.

Note:

The output may appear different than other FTP clients.

4 Send a loopback document

Loopback testing allows FTP Connector users to submit and receive messages. The test can be used to verify connectivity.

There are two methods to send a document for loopback testing. The first method is to prepare an EDI document with your ID as both sender and receiver of the document. Save the document as ***ftptest***. The second method uses the ***routing*** command to specify the document type, to and from. Loopback testing requires the ***to*** and ***from*** to be your ID.

First Method (Modify EDI document).

Note:

Bold text indicates user input.

ftp> **cd /send**

200 Directory changed to /send

ftp> **put ftptest**

200 Command PORT okay

150 File status okay; about to open data connection

226-Transfer Complete

226 Tracking ID: F64D9487-4F78-3C44-67A0-296D29F4D180

local: ftptest remote: ftptest

3502 bytes sent in 0.00099 seconds (3457.96 Kbytes/s)

ftp> **dir**

200 Command PORT okay

150 File status okay; about to open data connection

ftptest F64D9487-4F78-3C44-67A0-296D29F4D180

226 Transfer Complete

56 bytes received in 0.0059 seconds (9.34 Kbytes/s)

Note:
The output may appear different than other FTP clients.

Second Method (using the quote command).

Note:
Bold text indicates user input.

ftp> **cd /send**

200 Directory changed to /send

ftp> **quote routing to={your ID},from={your ID},doctype={document type},format={document format}**

200 Command ROUTING okay

ftp> **put ftptest**

200 Command PORT okay

150 File status okay; about to open data connection

226-Transfer Complete

226 Tracking ID: F64D9487-4F78-3C44-67A0-296D29F4D180

local: ftptest remote: ftptest

3502 bytes sent in 0.00099 seconds (3457.96 Kbytes/s)

ftp> **dir**

200 Command PORT okay

150 File status okay; about to open data connection

ftptest F64D9487-4F78-3C44-67A0-296D29F4D180

226 Transfer Complete

56 bytes received in 0.0059 seconds (9.34 Kbytes/s)

Note:
The output may appear different (stet) other FTP clients.

5 Receive the Loopback Document (Mailbox only).

Note:
Bold text indicates user input.

ftp> **cd /receive**

200 Directory changed to /receive

ftp> **dir**

200 Command PORT okay

150 File status okay; about to open data connection

F64D9487-4F78-3C44-67A0-296D29F4D180 M100006 4026 EDI850_X12

.

.

.

226 Transfer Complete

996 bytes received in 0.25 seconds (3.88 Kbytes/s)

ftp> **get M100006**

200 Command PORT okay

150 File status okay; about to open data connection

226-Transfer Complete

226 Tracking ID: {tracking id number}

local: **M100006** remote: **M100006**

4026 bytes received in 0.031 seconds (95.57 Kbytes/s)

Note:

The output may appear different than other FTP clients.

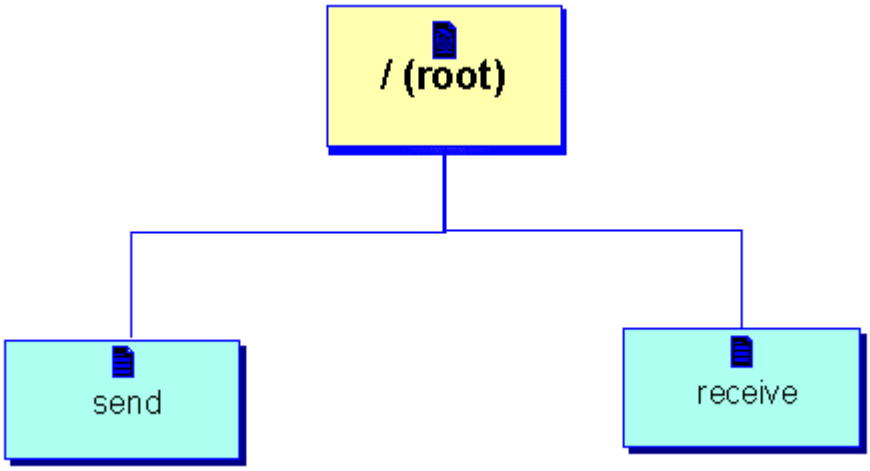
FUNCTIONAL DESCRIPTION-FTP CONNECTOR

Covisint does not provide the FTP client necessary to access the FTP Connector. A list of FTP clients is provided in Appendix A of this document. The FTP Connector does not support the ***Record*** or ***Block*** transfer structures.

Refer to the examples later in this chapter for correct command usage.

Directory Structure

The FTP Connector has the following structure for sending and receiving data.



The "cd" Command

The ***cd*** command is used to move from one directory to another in the directory structure.

cd / Changes the directory to root (/).

cd /send Changes the directory to send for sending data.

cd /receive Changes the directory to receive for receiving data.

The "quote routing" Command

The ***quote routing*** command gives you more flexibility to specify routing information for documents posted to the messaging hub. If a document sent to the messaging hub lacks a header with routing information (to, from, doctype) then the ***quote routing*** command allows you to specify these during their FTP session.

The command is based on name value pairs separated by commas. For example, ***quote routing to=partner1,from=partner2,doctype=850,format=X12*** sets the routing information for all documents during an FTP session. The routing command is not reset until you log out. The values may be changed at any time by reissuing the ***quote routing*** command. The following table lists the ***quote routing*** values accepted by the messaging hub.

Name	Description	Required / Optional
TO	The message destination.	Required
FROM	Who the message is from.	Required
DOCTYPE	Type of document. *Example* 850, DELFOR, etc.	Required
FORMAT	Format of the document. *Example* X12, EDIFACT, etc.	Required
STDVERSION	Version of the document. *Example* v2040, 97a, etc.	Optional
TOQUAL	Sender's qualifier. *Example* ZZ, 09, etc.	Optional
FROMQUAL	Receiver's qualifier. *Example* ZZ, 09, etc.	Optional
TID	Sender's tracking, or message, ID.	Optional

Example of quote command:

```
ftp> quote routing to=supplier123,from=oem123,doctype=850,format=X12
200 Command ROUTING okay
```

The "dir" Command

The ***dir*** command allows you to view a full listing of the current directory's contents. For security purposes, the ***dir*** command can be used only in the ***receive*** and ***send*** directories. Using ***dir*** in the send directory only displays the results for files ***put*** during the current session. For example, if you log out and log back in, the files ***put*** in the previous login are lost.

Format for the "dir" command:

Tracking ID	Filename	Bytes	Document Type
F64D9487-4F78-3C44-67A0-296D29F4D180	M1000006	4096	EDI850_X12

The "put" and "mput" Commands

To put files, the current directory must be the send directory (***cd /send***). Use the ***put*** and ***mput*** commands to send files to the messaging hub. If the ***quote routing*** command was issued, this applies to all ***puts*** during an FTP session. To change values during a session, issue the ***quote routing*** command again.

The messaging hub supports both ASCII and binary modes. If you send messages that contain binary data, be sure to issue the ***type*** command to set the mode to binary.

If a put command fails and a Tracking ID was not returned, then resend the message.

Example of sending a message.

Note:
Bold text indicates user input.

ftp> **cd /send**
200 Directory changed to /send
ftp> **lcd files**
Local directory now /export/home/company/files
ftp> **put DELFOR.DAT**
200 Command PORT okay
150 File status okay; about to open data connection
226-Transfer Complete
226 Tracking ID: F64D9487-4F78-3C44-67A0-296D29F4D180
local: DELFOR.DAT remote: DELFOR.DAT
4965 bytes sent in 0.0012 seconds (3885.12 Kbytes/s)
ftp> **mput DEL***
mput DELFOR.DAT? **y**
200 Command PORT okay
150 File status okay; about to open data connection
226-Transfer Complete
226 Tracking ID: E30C18B0-5A60-0C13-E5DD-B3FDFA485499

local: DELFOR.DAT remote: DELFOR.DAT

4965 bytes sent in 0.00093 seconds (5202.40 Kbytes/s)

mput DELJIT.TXT? **y**

200 Command PORT okay

150 File status okay; about to open data connection

226-Transfer Complete

226 Tracking ID: 89CB2133-C765-3592-629E-980C99B1D1D4

local: DELJIT.TXT remote: DELJIT.TXT

2855 bytes sent in 0.009 seconds (309.99 Kbytes/s)

ftp> **dir**

200 Command PORT okay

150 File status okay; about to open data connection

F64D9487-4F78-3C44-67A0-296D29F4D180 DELFOR.DAT 4965

E30C18B0-5A60-0C13-E5DD-B3FDFA485499 DELFOR.DAT 4965

89CB2133-C765-3592-629E-980C99B1D1D4 DELJIT.TXT 2855

226 Transfer Complete

165 bytes received in 0.021 seconds (7.51 Kbytes/s)

The "get" and "mget" Commands

To get or retrieve files from the messaging hub, you must be in the receive directory (***cd /receive***). By using the ***get*** and ***mget*** commands, you can retrieve files from your messaging hub mailbox. Once a message is retrieved, the message is not visible with the ***dir*** command.

> Note:
>
> Wildcards are supported using the wildcard character (%). For example, ***mget M0%*** retrieves all files from the mailbox with "***M0***" as the starting characters.

To retrieve a previously retrieved file, within storage time limits, the ***get*** command can be used to retrieve the file; ***mget*** does not work.

The messaging hub supports both ASCII and binary modes. If you retrieve messages that contain binary data, be sure to issue the ***type*** command to set the mode to binary.

Example of retrieving messages from a mailbox.

Note:

Bold text indicates user input.

```
ftp> cd /receive
200 Directory changed to /receive
ftp> dir
200 Command PORT okay
150 File status okay; about to open data connection
84a2818d-985a-11d7-8432-0050dae9ce1a    M0000028      14343   830
8609b2b2-985a-11d7-8432-0050dae9ce1a    M0000030      14343   830
86b953a5-985a-11d7-8432-0050dae9ce1a    M0000033      14343   830
86593243-985a-11d7-8432-0050dae9ce1a    M0000035      14343   830
8770bcce-985a-11d7-8432-0050dae9ce1a    M0000042      14343   830
878185b0-985a-11d7-8432-0050dae9ce1a    M0000048      14343   830
226 Transfer Complete
17279 bytes received in 0.3 seconds (56.28 Kbytes/s)
ftp> get M0000028
200 Command PORT okay
150 File status okay; about to open data connection
226-Transfer Complete
226 Tracking ID: 84a2818d-985a-11d7-8432-0050dae9ce1a
local: M0000028 remote: M0000028
14343 bytes received in 0.039 seconds (362.48 Kbytes/s)
ftp> mget M000003%
mget M0000030? y
200 Command PORT okay
150 File status okay; about to open data connection
226-Transfer Complete
226 Tracking ID: 8609b2b2-985a-11d7-8432-0050dae9ce1a
local: M0000030 remote: M0000030
```

14343 bytes received in 0.015 seconds (950.65 Kbytes/s)

mget M0000033? **y**

200 Command PORT okay

150 File status okay; about to open data connection

226-Transfer Complete

226 Tracking ID: 86b953a5-985a-11d7-8432-0050dae9ce1a

local: M0000033 remote: M0000033

14343 bytes received in 0.033 seconds (428.70 Kbytes/s)

mget M0000035? **y**

200 Command PORT okay

150 File status okay; about to open data connection

226-Transfer Complete

226 Tracking ID: 86593243-985a-11d7-8432-0050dae9ce1a

local: M0000035 remote: M0000035

14343 bytes received in 0.027 seconds (523.60 Kbytes/s)

ftp> **dir**

200 Command PORT okay

150 File status okay; about to open data connection

8770bcce-985a-11d7-8432-0050dae9ce1a M0000042 14343 830

878185b0-985a-11d7-8432-0050dae9ce1a M0000048 14343 830

226 Transfer Complete

16652 bytes received in 0.29 seconds (55.30 Kbytes/s)

APPENDIX – FTP CONNECTOR

FTP Clients

Most command-line FTP clients should be compatible with the FTP Connector. Below is a list of tested FTP clients.

Platform	OS Version
Sun	Sun Solaris 2.6 - Sun Solaris 5.9
Microsoft Windows	98/2000/XP

Best Practices

Push/Pull

The ***push/pull*** approach is common when a customer does not operate an FTP server. This approach allows a customer to use standard FTP client software to ***push*** data to an FTP server. The FTP command that is commonly used is ***put***. The ***put*** command allows the customer to take a document (usually from the file system) and send it to the FTP server. To retrieve documents, the customer would use the FTP command ***get***. This command allows the customer to retrieve, or ***pull***, files (usually by file name) from the FTP server. For an illustration of the ***push/pull*** approach, see Figure 1.

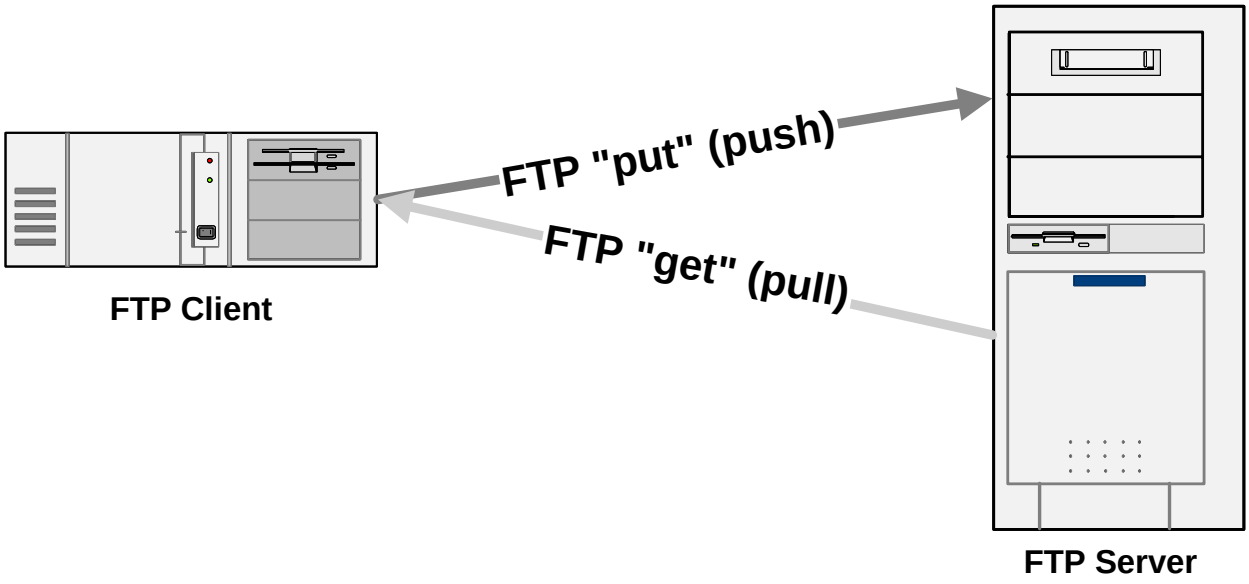


Figure 1. Push/Pull Approach

Push/Push

The ***push/push*** approach is common when trading partners operate their own FTP servers. This approach models a publish/subscribe paradigm, meaning each trading partner publishes (FTP puts) documents and listens, or waits, for documents to be pushed to them (subscribe). This approach minimizes document latency that can be caused by polling in the ***pull*** approach. For an illustration of ***push/push***, see Figure 2.

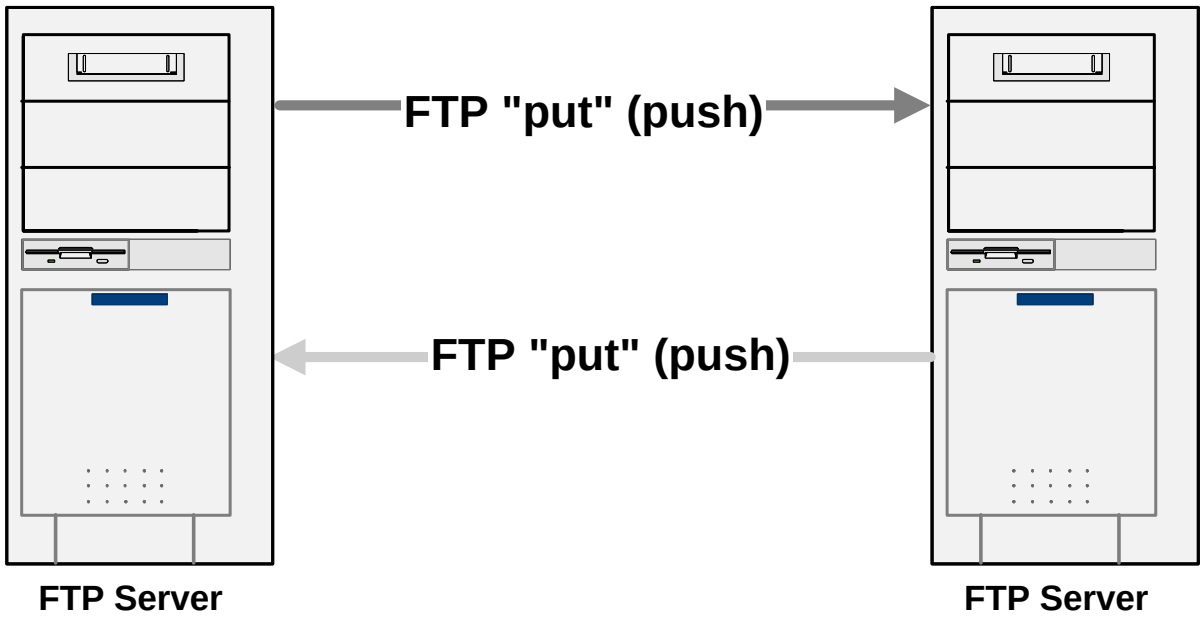


Figure 2. Push/Push Approach

Common Errors – FTP Connector

Error	Recommended Actions
550 Requested action not taken or 550 File {filename} unavailable	Make sure your directory reflects the command issued. For ***put***, users must be in the /send directory. For ***get***, users must be in the /receive directory.
425 Can't open data connection	Possible network issue. Retry sending or receiving document. If the problem continues, contact Covisint Support.
426 Connection closed; transfer aborted	Possible network issue. Retry sending or receiving document. If the problem continues, contact Covisint Support.
530 Access denied	Retry login. If unsuccessful, contact Covisint Support.

Glossary

The following terms are used in the guide:

Channel - A channel defines the connectivity options for a trading partner. Channels include both inbound and outbound connection options. An example of an inbound channel would be FTP inbound ("put"). An example of an outbound channel would be a Mailbox ("get") or FTP push.

Files - From the perspective of the client using FTP Connector, everything is in file terms. EDI data is locally stored as files that are uploaded to the server host on a per-file basis. Documents on the server host are perceived as individual files and can be downloaded to the local host one at a time.

FTP - File Transfer Protocol, an application-level protocol standard used on IP networks, such as the Internet, to transfer files between (homogeneous or non-homogeneous) computers.

FTP Client - Applications on client computers that provide an interface to users and communicate with FTP servers to transfer files. This software is typically provided as a component of the operating system. Covisint will not provide an FTP client. See Appendix A, "FTP clients," for a list of FTP Client software.

FTP Server - A Messaging Hub component that provides access to Messaging Hub Services using the FTP protocol.

FTP user command - The commands provided by an FTP client's user interface.

TCP - A transportation-level protocol that is widely used on IP networks, including the Internet.