



FTP Technical Overview

Contents

FTP Connections	2
A Sample FTP Session	4
For Additional Help...	7

The majority of questions (and problems) that come up about implementing an FTP server with Rumpus are actually about networking and FTP in general. This article is intended to provide a basic understanding of how FTP servers work on a TCP/IP network and what you need to do to make sure users can access your FTP server.

For a very basic description of how to configure a local, private network so that your FTP server can be accessed by external Internet users, be sure to open the Get Connected window in the Rumpus control application, and flip to the "From Outside Your LAN" tab. There, Rumpus displays important networking information and offers links to additional resources.

FTP Connections

The FTP protocol is very different from HTTP (Web) in a number of important ways. First, FTP is stateful, meaning that a single connection between the client and server is created and maintained during an FTP session, and actions taken by the client may build on one another. Under HTTP, each transaction is separate from every other, and multiple unique transactions may be required to accomplish a single task (such as display a Web page).

More importantly, there are 2 distinct types of FTP connections: Control and Data. A single control connection is established when the client initiates an FTP session. The control connection is then maintained and used to send commands and receive response messages. When data needs to be transferred (such as when a file is uploaded or downloaded, or when a directory listing is requested) a data connection is opened, used for the transfer, and then closed.

When the client wishes to transfer a file or retrieve a directory listing, it issues a command to the server requesting a data connection. Data connections can be "passive" or "active". Passive mode data connections are used when the client needs to initiate the connection and have the server "listen" for it. The client can also choose to request an active mode connection, in which case the server initiates the connection back to the client.

Many client networks are configured to allow only outgoing connections, which means that clients on these networks will be able to use only passive mode connections. For this reason, most FTP clients default to passive mode, though most provide an option to change this default. In Fetch, for example, a checkbox called "Use passive mode transfers" is provided on the Preferences window, on the "Firewall" tab in version 4 and on the "General" tab in version 5.

Active mode connections take place, by default, on the standard FTP Data port, port 20. Passive mode connections don't, and it's up to the server to tell the client which port to connect to. So that the server can keep data connections straight among multiple users, multiple ports are used. By default, Rumpus uses ports 3000 and above for its passive mode data connections. The highest port used is 3000 plus the number of simultaneous connections allowed. So, for example, if your Rumpus server is configured to allow up to 24 simultaneous users, Rumpus will use ports 3000 through 3024 for inbound data connections. If networking restrictions require it, the Rumpus passive mode port range can be changed.

The defined FTP control connection port is 21, and most firewalls and routers are configured to permit communication on this port. For this reason, firewalls and routers will generally allow FTP users to log in to an FTP server anywhere on the Internet. However, since data connections are more varied, it's much more common to have trouble raising FTP data connections.

If users are able to Login to your FTP server, but can't transfer files or get directory listings, it is very, very likely that the problem is due to a firewall or router blocking access to your FTP server. (The other likely reason is that the user simply hasn't been granted permission in Rumpus to perform the task. See the "User Accounts" help page in Rumpus for details.)

The inability to see directory listings or transfer files is easily the most common problem when running FTP servers, and is virtually always caused by a network that won't permit a data connection to be established.

If your network is protected by a firewall, we recommend that your FTP server be placed on the outside of the firewall, if possible. This will assure that FTP transactions aren't blocked by your firewall. If this isn't possible, you will need to enable inbound traffic to your FTP server on the FTP connection port (21), as well as the data ports used by Rumpus (3000 and above). If you don't do this, clients that connect using passive mode data connections (where the client initiates the connection to your server) will be blocked.

For clients on networks with restrictions on inbound connections, passive mode connections are required, as the FTP server will not be able to connect through the firewall. Also, most browsers that provide FTP capability use passive mode connections only. It is crucial that access to ports 3000 and up (through 3000 plus the maximum number of simultaneous connections) on your FTP server is available.

To allow active mode connections, both your network and the network of the client must allow connections on port 20 from the server side to the client side.

To allow passive mode connections, both your network and the network of the client must allow connections on ports 3000 through 3008 (the upper bound may vary as described above) from the client side to the server.

A Sample FTP Session

Once a control connection can be made between the FTP client and the server, the first step in diagnosing almost any problem is to look at the session transcript. Actions on the control connection are performed using plain ASCII text, and the transcript is simply a record of the commands and responses sent between the client and server. Virtually all dedicated FTP clients (like FileZilla, Fetch and Transmit) provide a transcript window which can be used to see exactly what is happening on the control connection.

Here is a sample FTP transcript:

```
220: Welcome To Rumpus!
      Service ready for new user
Connected to 192.168.1.175.
Cmd: USER john
331: Password required
Cmd: PASS xxxxxxxx
230: User logged in
Logged in to 192.168.1.175 as john.
Cmd: SYST
215: MACOS Peter's Server
Cmd: PWD
257: "/" is current directory.
Cmd: PASV
227: Entering Passive Mode (192,168,1,175,11,187)
Cmd: LIST
150: Opening connection
      The directory listing is sent via the data connection
226: Transfer complete
Cmd: TYPE I
200: Type set to I (Image)
Cmd: PASV
227: Entering Passive Mode (192,168,1,175,11,185)
Cmd: RETR test
150: Opening connection (43182 bytes)
226: Transfer complete
QUIT
221-Bye from Rumpus
```

Taken as a whole, an FTP session transcript may be intimidating, but reading through it a few lines at a time makes things relatively simple:

```
220: Welcome To Rumpus!  
      Service ready for new user  
      Connected to 192.168.1.175.
```

Here, the client initiates a connection with the server. As soon as the connection is established, the server sends a "ready!" message, letting the client know that it is prepared to accept commands.

```
Cmd: USER john  
331: Password required  
Cmd: PASS xxxxxxxx  
230: User logged in  
      Logged in to 192.168.1.175 as john.
```

The first task the client must perform is logging in. It issues a "USER" command, followed by the account name, and when prompted, supplies the authentication password using the "PASS" command. The server responds with the "User logged in" message.

```
Cmd: SYST  
215: UNIX Type: L8  
Cmd: PWD  
257: "/" is current directory.
```

Once a user is logged in, the client can issue any command it wishes. Many clients, like this one (Transmit), next ask the server what type of server it has connected to. The response to the "SYST" command is important, as many FTP clients will alter their behavior for different FTP servers on different operating systems. In the example above, Rumpus identifies itself as a standard Unix-style FTP service.

Note for Mac servers: Rumpus for Mac will, by default, identify itself as "MACOS Peter's Server", which is the standard Mac FTP service response, allowing clients to recognize that the service supports Mac-specific features like MacBinary mode transfers. ("Peter" refers to Peter Lewis, the author of the very first FTP Server for Mac, back in the 1990s. He and Rumpus aren't related in any way, but Rumpus reports itself that way so that Mac-specific clients will behave properly.)

The "PWD" command simply asks the server to identify the current working directory for the client. Rumpus responds with "/", telling the client it has been logged in at the top level directory.

```

Cmd: PASV
227: Entering Passive Mode (192,168,1,175,11,187)
Cmd: LIST
150: Opening connection

```

The directory listing is sent via the data connection

```
226: Transfer complete
```

Most FTP clients will automatically request a directory listing after logging in. Here, a "PASV" command lets the server know that a data connection is needed, and that it will use passive mode to create it. The server then begins "listening" for the connection, and tells the client what address and port to connect to. Once the client has raised the data connection, a "LIST" command tells the server to send the directory listing on the data connection. When the server has completed the directory listing, it closes the data connection.

The client now displays the directory listing for the user in the Fetch window. In this sample session, the user then downloads the file "test".

```

Cmd: TYPE I
200: Type set to I (Image)
Cmd: PASV
227: Entering Passive Mode (192,168,1,175,11,187)
Cmd: RETR test
150: Opening connection (43182 bytes)
226: Transfer complete

```

Files are generally transferred in binary mode (since they may contain non-ASCII characters), so the client tells the server to send binary data (as in an Image). As we saw in the directory listing above, a data connection is needed to transfer the file data, so the client requests another passive mode connection. As before, the server starts listening for the connection and tells the client where to connect to. Once the connection is open, the "RETR" command ("RETRieve") specifies the file to be sent.

```
QUIT
```

```
221-Bye from Rumpus
```

When the user is done, the client window is closed or quit, so the client logs out with the "QUIT" command.

For Additional Help...

As always, contact Maxum Technical Support at "support@maxum.com" if you need additional help. Even if the information above doesn't answer your question or solve your problem, having a basic understanding of how FTP works will make it easier for our technical support staff to help you. When you contact us, be sure to include details about the network that exists between the client and the server (especially information about firewalls) and a session transcript, if available.