

Networking Overview Of Rumpus FTP

What you should know about FTP and the networking required to support it.

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 Network Setup Assistant in the Rumpus control application. The setup assistant will provide specific setup information for your server, though it will not explain the FTP protocol or networking requirements in detail.

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 FTP 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 it's 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. See the "Advanced Options" section of the "FTP Settings" help page in Rumpus for details.

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 not be possible.

For clients on networks with restrictions on inbound-only connections, passive mode connections are required, as the FTP server will not be able to connect through the firewall. Also, browsers (Netscape Navigator and Microsoft Internet Explorer) 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.

Most dedicated FTP client applications allow you to choose between active mode and passive mode connections. If one doesn't work, it's often worthwhile to try the other.

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. All 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 Fetch and Interarchy on the Mac) provide a transcript window which can be used to see exactly what is happening on the control connection.

Here is a sample FTP transcript:

```
Connecting to 192.168.1.27 port 21
220-Welcome to Rumpus!
220 Service ready for new user
USER john
331 Password required
PASS
230 User logged in
SYST
215 MACOS Peter's Server
PWD
257 "/" is current directory.
MACB ENABLE
```

```
200 MacBinary enabled
PWD
257 "/" is current directory.
TYPE A
200 Type set to A (ASCII)
PASV
227 Entering Passive Mode (192,168,1,27,11,187)
LIST
  The directory listing is sent via the data connection
TYPE I
200 Type set to I (Image) [MACBINARY ENABLED]
PASV
227 Entering Passive Mode (192,168,1,27,11,187)
RETR test
150 Opening connection (384 bytes)
  The file "test" is sent on the data connection
226 Transfer complete
Download completed at 1/14/02 11:11:01 AM
QUIT
221-Bye from Rumpus
221 Goodbye - service closing connection
```

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:

```
Connecting to 192.168.1.27 port 21
220-Welcome to Rumpus!
220 Service ready for new user
```

Here, the client attempts to initiate 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.

```
USER john
331 Password required
PASS
230 User logged in
```

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.

```
SYST
215 MACOS Peter's Server
PWD
257 "/" is current directory.
```

Once a user is logged in, the client can issue any command it wishes. Many clients, like this one (Fetch), 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. For example, a number of Mac FTP clients look for "MACOS Peter's Server" to determine whether or not the server supports the MacBinary file transfer protocol. (For more information about MacBinary, see the "File Transfer Basics" help article.)

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.

```
MACB ENABLE
200 MacBinary enabled
PWD
257 "/" is current directory.
```

In this transcript, Fetch has determined that the server is a Mac server capable of transferring files in MacBinary mode. It issues the Mac-specific "MACB ENABLE" command to tell the server that files should be transferred using MacBinary, and Rumpus responds with a "200 OK" message. Fetch then checks the working directory again.

```
TYPE A
200 Type set to A (ASCII)
PASV
227 Entering Passive Mode (192,168,1,27,11,187)
LIST
The directory listing is sent via the data connection
```

Most FTP clients will automatically request a directory listing after logging in. Here, the client sets the transfer mode to "A", or ASCII, so that the upcoming directory listing will be sent as plain text ASCII. Next, 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 (Fetch) now displays the directory listing for the user in the Fetch window. In this sample session, the user drags the file "test" from the Fetch window to a folder in the Finder. This initiates a file download.

```
TYPE I
200 Type set to I (Image) [MACBINARY ENABLED]
PASV
227 Entering Passive Mode (192,168,1,27,11,187)
RETR test
150 Opening connection (384 bytes)
  The file "test" is sent on the data connection
226 Transfer complete
Download completed at 1/14/02 11:11:01 AM
```

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 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
221 Goodbye - service closing connection
```

When the user is done, the Fetch window is closed or Fetch is 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.